

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ
АДМИНИСТРАЦИЯ ПАРКА ВЫСОКИХ ТЕХНОЛОГИЙ

**Открытые семантические технологии
проектирования интеллектуальных систем**

OSTIS-2013

Open Semantic Technologies for Intelligent Systems

**МАТЕРИАЛЫ
III МЕЖДУНАРОДНОЙ
НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ**

(Минск, 21 – 23 февраля 2013 года)

Минск
БГУИР
2013

УДК 004.822-027.31

ББК 32.0813-02

О-83

Редакционная коллегия:

В. В. Голенков (отв. ред.), Н. М. Боргест, Т. А. Гаврилова, Л. С. Глоба, В. А. Головки, А. Н. Гордей, В. В. Грибова, Н. А. Гулякина, А. П. Еремеев, П. П. Ермолов, И. В. Ефименко, А. В. Заболеева-Зотова, Ю. А. Загоруйко, А. А. Иванюк, В. С. Ижуткин, А. С. Клещев, Б. А. Кобринский, О. А. Козлов, Л. Г. Комарцова, О. П. Кузнецов, В. М. Курейчик, Б. М. Лобанов, К. А. Найденова, О. А. Невзорова, Г. С. Осипов, Б. В. Палюх, А. А. Петровский, Г. С. Плесневич, В. Г. Родченко, Л. В. Рычкова, С. В. Смирнов, В. С. Смородин, С. Ю. Соловьёв, П. И. Соснин, В. Л. Стефанюк, Д. Ш. Сулейманов, В. Б. Тарасов, Ю. Ф. Тельнов, А. В. Тузиков, А. А. Харламов, И. Э. Хейдоров, В. Ф. Хорошевский, А. Ф. Чернявский, А. А. Шарипбаев, С. С. Щербак

Организаторы конференции

Белорусский государственный университет информатики и радиоэлектроники
Российская ассоциация искусственного интеллекта
Государственное учреждение «Администрация Парка высоких технологий» (Республика Беларусь)
Объединённый институт проблем информатики Национальной академии наук Беларуси
Тверской государственный технический университет
Научно-исследовательский институт «Прикладная семиотика» АН РТ
Институт информатизации образования Российской академии образования
Международная ИТ-компания «Itransition»
Компания «Melesta»

Международная сервисная ИТ-компания «Exigen® Services»
Компания «Qulix Systems»
Digital-агентство «ARTOX media»
Компания «Речевые Технологии»
Компания «ВирусБлокАда»
Компания «Сакрамент»
Компания «htc»
Компания «ИксБи Софтваре»
Частное унитарное предприятие «Лаборатория стартапов»
Компания «Адмирал Софт»
Компания «Antalika»
Компания «Эксдел»
Компания «СофтКлуб»
Компания «СофтТеко»

Техническая и информационная поддержка

Международный журнал «Программные продукты и системы»
Научный журнал «Информатика»
Научно-практический журнал для специалистов «Электроника ИНФО»

Научно-практический журнал «Речевые технологии»
Научный журнал «Онтология проектирования»
Открытый проект «MyGui»
Открытый проект «Ogre»

*Издание осуществлено по заказу государственного учреждения
«Администрация Парка высоких технологий»*

**Открытые семантические технологии проектирования
О-83 интеллектуальных систем = Open Semantic Technologies for Intelligent
Systems (OSTIS-2013): материалы III Междунар. научн.-техн. конф.
(Минск, 21-23 февраля 2013г.) / редкол. : В. В. Голенков (отв. ред.) [и др.]. –
Минск : БГУИР, 2013. – 592 с.
ISBN 978-985-488-956-6.**

Сборник включает прошедшие рецензирование статьи III Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем».

Сборник предназначен для преподавателей высших учебных заведений, научных сотрудников, студентов, аспирантов, магистрантов, а также для специалистов предприятий в сфере проектирования интеллектуальных систем.

Материалы сборника одобрены Программным комитетом OSTIS-2013 и печатаются в виде, представленном авторами.

**УДК 004.822-027.31
ББК 32.0813-02**

ISBN 978-985-488-956-6

© УО «Белорусский государственный университет информатики и радиоэлектроники», 2013

ПРОГРАММНЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

- **Кузнецов Олег Петрович** - д.т.н., проф., зав.лаб. «Институт проблем управления РАН», г. Москва, РФ - **председатель**
- **Боргест Николай Михайлович** - к.т.н., доц., профессор в «Самарском государственном аэрокосмическом университете имени Академика С.П.Королёва», г.Самара, РФ
- **Гаврилова Татьяна Альбертовна** - д.т.н., проф., зав.каф. информационных технологий в менеджменте «Высшая Школа менеджмента СПбГУ», г.Санкт-Петербург, РФ
- **Глоба Лариса Сергеевна** - д.т.н., проф., заф.каф. в «Национальном техническом университете Украины», «Киевский политехнический институт», г. Киев, Украина
- **Голенков Владимир Васильевич** - д.т.н., проф., заф.каф. в Белорусском государственной университет информатики и радиоэлектроники, г. Минск, РБ
- **Головко Владимир Адамович** - д.т.н., проф., заф.каф. интеллектуальных информационных технологий в Брестском государственном техническом университете г. Брест, РБ
- **Гордей Александр Николаевич** - д.фил.н., проф., директор Республиканского института китаеведения имени Конфуция БГУ, г. Минск, РБ
- **Грибова Валерия Викторовна** - д.т.н., с.н.с., зав.лаб. интеллектуальных систем в Институте автоматизации и процессов управления ДВО РАН, г.Владивосток, РФ
- **Гулякина Наталья Анатольевна** - к.ф.-м.н., доц., зам.заф.каф. в Белорусском государственном университете информатики и радиоэлектроники, г. Минск, РБ
- **Еремеев Александр Павлович** - д.т.н., проф., зав. каф. прикладной математики в МЭИ (ТУ), г. Москва, РФ
- **Ермолов Павел Петрович** - к.т.н., доц.каф. радиотехники и телекоммуникаций в Севастопольском национальном техническом университете, г. Севастополь, Украина
- **Ефименко Ирина Владимировна** - к.фил.н., зам.декана факультета филологии в Высшей Школе менеджмента СПбГУ, г. Москва, РФ
- **Заболеева-Зотова Алла Викторовна** - д.т.н., проф., ВолгГТУ, г. Волгоград, РФ
- **Загорулько Юрий Алексеевич** - к.т.н., с.н.с., заведующий лаб. Института систем информатики им. А.П. Ершова СО РАН, г. Новосибирск, РФ
- **Иванюк Александр Александрович** - д.т.н., доц., зав.каф. в Белорусский государственный университет информатики и радиоэлектроники, г. Минск, РБ
- **Ижуткин Виктор Сергеевич** - д.ф.-м.н., проф. ФГБОУ ВПО «Национальный исследовательский университет (Московский энергетический институт)», г. Москва, РФ
- **Клещев Александр Сергеевич** - д.ф.-м.н., проф., главный научный сотрудник Института автоматизации и процессов управления Дальневосточного отделения РАН, г. Владивосток, РФ
- **Кобринский Борис Аркадьевич** - д.мед.н., проф., руководитель «Медицинский центр новых информационных технологий», г. Москва, РФ
- **Козлов Олег Александрович** - д.п.н., проф., заместитель директора по учебно-методической работе «Институт информатизации образования» Российской академии образования, г. Москва, РФ
- **Комарцова Людмила Георгиевна** - д.т.н., проф., Калужский филиала МГТУ им. Баумана, г. Калуга, РФ
- **Курейчик Виктор Михайлович** - д.т.н., проф., заместитель руководителя ТТИ ЮФУ (Таганрогский технологический институт Южного федерального университета) по инновационной и научной деятельности, г. Таганрог, РФ
- **Лобанов Борис Мефодьевич** - д.т.н., главный научный сотрудник ОИПИ НАН Б, г. Минск, РБ
- **Найденова Ксения Александровна** - к.т.н., с.н.с, старший научный сотрудник в Военно-медицинской академии, г. Санкт-Петербург, РФ

- **Невзорова Ольга Авенировна** - к.т.н., доц., зам. дир. НИИ «Прикладная семиотика» АН РТ, г. Казань, РФ
- **Осипов Геннадий Семенович** - д.ф.-м.н., проф., заместитель директора по научной работе «Институт системного анализа Российской академии наук», г. Москва, РФ
- **Палюх Борис Васильевич** - д.т.н., проф., ректор «Тверской государственной технической университет», г. Тверь, РФ
- **Петровский Александр Александрович** - д.т.н., профессор, заф. каф. электронных вычислительных средств «Белорусский государственный университет информатики и радиоэлектроники», г. Миснк, РБ
- **Плесневич Геральд Станиславович** - к.ф.-м.н., профессор «Московский энергетический университет», г. Москва, РФ
- **Родченко Вадим Григорьевич** - к.т.н., доц., «Гродненский государственный университет имени Янки Купалы», г. Гродно, РБ
- **Рычкова Людмила Васильевна** - к.фил.н., заведующий кафедрой «Гродненский государственный университет имени Янки Купалы», г. Гродно, РБ
- **Смирнов Сергей Викторович** - д.т.н., проф., директор «Институт проблем управления сложными системами Российской академии наук» (ИПУСС РАН), г. Самара, РФ
- **Сморodin Виктор Сергеевич** - д.т.н., заведующий кафедрой МПУ «Гродненский государственный университет имени Янки Купалы», г. Гродно, РБ
- **Соловьёв Сергей Юрьевич** - д.ф.-м.н., профессор «Московский государственный университет имени М.В.Ломоносова», г. Москва, РФ
- **Соснин Петр Иванович** - д.т.н., проф., зав. кафедрой вычислительной техники «Ульяновский государственный технический университет», г. Ульяновск, РФ
- **Стефанюк Вадим Львович** - д.т.н., проф., ведущий научный сотрудник «Институт проблем передачи информации Российской академии наук», г. Москва, РФ
- **Сулейманов Джавдет Шевкетович** - д.т.н., проф., академик АН Татарстана «Казанский госуниверситет», г. Казань, РФ
- **Тарасов Валерий Борисович** - к.т.н., доц., «Московский государственный технический университет им. Н.Э.Баумана», г. Москва, РФ
- **Тельнов Юрий Филиппович** - д.э.н., проф., заведующий кафедрой Прикладной информатики в экономике «Московский государственный университет экономики, статистики и информатики», г. Москва, РФ
- **Тузиков Александр Васильевич** - д.ф.-м.н., проф., генеральный директор «Объединенный институт проблем информатики НАН Беларуси», г. Минск, РБ
- **Харламов Александр Александрович** - д.т.н., с.н.с. «Институт Высшей Нервной Деятельности и Нейрофизиологии РАН», г. Москва, РФ
- **Хейдоров Игорь Эдуардович** - к.ф.-м.н., доц., доцента кафедры радиофизики и цифровых медиатехнологий БГУ, г. Минск, РБ
- **Хорошевский Владимир Федорович** - д.т.н., проф., главный научный сотрудник НИУ ВШЭ и зав. сектором «Вычислительный центра РАН», г. Москва, РФ
- **Чернявский Александр Фёдорович** - академик НАН Беларуси, доктор техн. наук, профессор БГУ, г. Минск, РБ
- **Шарипбаев Алтынбек Амирович** - д.т.н., проф., профессор «Евразийский национальный университет имени Л.Н. Гумилева», г. Астана, Казахстан
- **Щербак Сергей Сергеевич** - к.т.н., доц., доцент кафедры Информационно-управляющих систем «Кременчугский национальный университет имени Михаила Остроградского», г. Кременчуг, Украина

ОРГАНИЗАЦИОННЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

- **Батура Михаил Павлович,**
ректор Белорусского государственного университета информатики и радиоэлектроники
- **Кузнецов Александр Петрович,**
проректор Белорусского государственного университета информатики и радиоэлектроники по научной работе
- **Никульшин Борис Викторович,**
проректор Белорусского государственного университета информатики и радиоэлектроники по учебной работе и информатизации
- **Живицкая Елена Николаевна,**
проректор Белорусского государственного университета информатики и радиоэлектроники по учебной работе и менеджменту качества
- **Шилин Леонид Юрьевич,**
декан факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники
- **Голенков Владимир Васильевич,**
заведующий кафедрой интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники
- **Сушко Оксана Ренгольдовна,**
начальник патентно-информационного отдела Белорусского государственного университета информатики и радиоэлектроники
- **Титович Анна Францевна,**
начальник Управления международного сотрудничества Белорусского государственного университета информатики и радиоэлектроники
- **Лихачевский Дмитрий Викторович,**
начальник Управления подготовки научных кадров высшей квалификации Белорусского государственного университета информатики и радиоэлектроники
- **Боярко Алла Викторовна,**
руководитель пресс-службы Белорусского государственного университета информатики и радиоэлектроники

ORGANIZATIONAL COMMITTEE

- **Mikhail Batura,**
Rector, Belarusian State University of Informatics and Radioelectronics
- **Alexander Kuznetsov,**
Vice-Rector for Research and Development, Belarusian State University of Informatics and Radioelectronics
- **Boris Nikulshin,**
Vice-Rector for Education and Informatization, Belarusian State University of Informatics and Radioelectronics
- **Elena Zhivitskaya,**
Vice-Rector for Quality Management, Belarusian State University of Informatics and Radioelectronics
- **Leanid Shylin,**
Dean of the Faculty of Information Technologies and Control, Belarusian State University of Informatics and Radioelectronics
- **Vladimir Golenkov,**
Head of Informational Intelligent Technologies Chair, Belarusian State University of Informatics and Radioelectronics
- **Oksana Sushko,**
Belarusian State University of Informatics and Radioelectronics
- **Anna Titovich,**
Belarusian State University of Informatics and Radioelectronics
- **Dmitry Likhachevsky,**
Belarusian State University of Informatics and Radioelectronics
- **Alla Boyarko,**
Belarusian State University of Informatics and Radioelectronics

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ	21
--------------------	----

СЕКЦИЯ 1. ПРИНЦИПЫ ПОСТРОЕНИЯ И СТРУКТУРА ТЕХНОЛОГИИ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	23
--	----

Грибова В.В., Клещев А.С. ОБЛАЧНЫЕ СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СЕРВИСОВ	25
--	----

Загоруйко Ю.А. ТЕХНОЛОГИИ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ, ОСНОВАННЫЕ НА ИНТЕГРИРОВАННОЙ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ	31
---	----

Ануреев И.С., Атучин М.М. ТЕХНОЛОГИЯ ФОРМАЛЬНО-ЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ И ПРОТОТИПИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	43
--	----

Глоба Л.С., Терновой М. Ю., Новогрудская Р.Л. МЕТОД ОРГАНИЗАЦИИ СЛАБОСВЯЗАННЫХ ИНФОРМАЦИОННЫХ РЕСУРСОВ НА ПОРТАЛАХ ЗНАНИЙ	49
---	----

Голенков В.В., Гулякина Н.А. ОТКРЫТЫЙ ПРОЕКТ, НАПРАВЛЕННЫЙ НА СОЗДАНИЕ ТЕХНОЛОГИИ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	55
--	----

СЕКЦИЯ 2. БАЗОВЫЕ СЕМАНТИЧЕСКИЕ МОДЕЛИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ И ИХ РЕАЛИЗАЦИЯ	79
---	----

Тимченко В.А. МЕТОД ПРЕОБРАЗОВАНИЯ КЛАССОВ СЕМАНТИЧЕСКИХ СЕТЕЙ	81
---	----

Жилякова Л.Ю. ПОСТРОЕНИЕ АССОЦИАТИВНОЙ МОДЕЛИ ПАМЯТИ. КОГНИТИВНЫЙ ПОДХОД	87
---	----

Невзорова О.А., Кириллович А.В. ТЕХНОЛОГИИ СВЯЗЫВАНИЯ ДАННЫХ В ПРОСТРАНСТВЕ ОТКРЫТЫХ ДАННЫХ НА ПРИМЕРЕ МАТЕМАТИЧЕСКОЙ КОЛЛЕКЦИИ	91
---	----

Ланин В.В., Лядова Л.Н. ИССЛЕДОВАТЕЛЬСКИЙ ПОРТАЛ ПО МОДЕЛИРОВАНИЮ ИНФОРМАЦИОННЫХ СИСТЕМ	97
---	----

Галушка И.Н., Завгородний В.В., Солошич С.Н., Щербак С.С. ПРАКТИЧЕСКИЕ АСПЕКТЫ РЕАЛИЗАЦИИ ПОИСКОВЫХ ИНТЕРФЕЙСОВ СЕМАНТИЧЕСКИХ ПРИЛОЖЕНИЙ LINKED DATA	103
--	-----

Вишняков В.А., Бородаенко Ю.В. СЕМАНТИЧЕСКИЕ ВЭБ-СЕРВИСЫ ДЛЯ ИНТЕГРАЦИОННЫХ РЕШЕНИЙ	107
--	-----

Яцкевич Н.И., Апанасович В.В. КРУПНОМАСШТАБНАЯ ОЦЕНКА СЕМАНТИЧЕСКИХ ТЕХНОЛОГИЙ	111
---	-----

Колб Д.Г., Фурман О.Д. РАЗРАБОТКА WEB-САЙТОВ СО СЛОЖНОСТРУКТУРИРОВАННЫМ КОНТЕНТОМ НА ОСНОВЕ СЕМАНТИЧЕСКИХ СЕТЕЙ	115
Корончик Д. Н. РЕАЛИЗАЦИЯ ХРАНИЛИЩА УНИФИЦИРОВАННЫХ СЕМАНТИЧЕСКИХ СЕТЕЙ	125
Вереник Н.Л., Татур М.М. ПОСТРОЕНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ВЕКТОРНОГО ПРОБЛЕМНО-ОРИЕНТИРОВАННОГО ПРОЦЕССОРА	129
СЕКЦИЯ 3. СЕМАНТИЧЕСКИЕ МОДЕЛИ, СРЕДСТВА И МЕТОДЫ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ И ПАКЕТОВ ПРОГРАММ, ОРИЕНТИРОВАННЫХ НА ОБРАБОТКУ ЗНАНИЙ	133
Кулинич А.А. КОНЦЕПТУАЛЬНЫЕ «КАРКАСЫ» ПЛОХО ОПРЕДЕЛЕННЫХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ	135
Крюков К.В., Кузнецов О.П., Суховеров В.С. О ПОНЯТИИ ФОРМАЛЬНОЙ КОМПЕТЕНТНОСТИ НАУЧНЫХ СОТРУДНИКОВ	143
Мальковский М.Г., Соловьев С.Ю. ИССЛЕДОВАНИЕ РОДОВИДОВЫХ ОТНОШЕНИЙ В ТЕРМИНОЛОГИЧЕСКИХ СЕТЯХ	147
Алексеев А.А., Добров Б.В., Лукашевич Н.В. ЛИНГВИСТИЧЕСКАЯ ОНТОЛОГИЯ – ТЕЗАУРУС РУТЕЗ	153
Загорулько М.Ю., Сидорова Е.А. ПРОГРАММНЫЙ ИНСТРУМЕНТАРИЙ РАЗРАБОТКИ ЛИНГВИСТИЧЕСКИХ РЕСУРСОВ	159
Зуенко А.А., Кулик Б.А., Фридман А.Я. ИНТЕЛЛЕКТУАЛЬНЫЕ ОБУЧАЮЩИЕ СИСТЕМЫ НА ОСНОВЕ АЛГЕБРАИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ВОПРОСНО-ОТВЕТНЫХ ТЕКСТОВ	165
Ивашенко В.П. УНИФИЦИРОВАННОЕ ПРЕДСТАВЛЕНИЕ И ИНТЕГРАЦИЯ ЗНАНИЙ	171
Давыденко И.Т. ТЕХНОЛОГИЯ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ НА ОСНОВЕ УНИФИЦИРОВАННЫХ СЕМАНТИЧЕСКИХ СЕТЕЙ	185
Смирнов С.В. ПАКЕТЫ ПРОГРАММ КАК ФОРМАЛЬНЫЕ ОНТОЛОГИИ: ПОСТРОЕНИЕ И ИСПОЛЬЗОВАНИЕ	191
Коварцев А.Н., Жидченко В.В., Попова-Коварцева Д.А., Аболмасов П.В. ПРИНЦИПЫ ПОСТРОЕНИЯ ТЕХНОЛОГИИ ГРАФОСИМВОЛИЧЕСКОГО ПРОГРАММИРОВАНИЯ	195
Пивоварчик О.В. СЕМАНТИЧЕСКИЕ ЯЗЫКИ ДЛЯ ОПИСАНИЯ ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММ, ОРИЕНТИРОВАННЫХ НА ОБРАБОТКУ ЗНАНИЙ	205

СЕКЦИЯ 4. СЕМАНТИЧЕСКИЕ МОДЕЛИ, СРЕДСТВА И МЕТОДЫ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ РЕШАТЕЛЕЙ ЗАДАЧ	217
Наместников А.М., Субхангулов Р.А., Филиппов А.А. ОНТОЛОГИЧЕСКИ-ОРИЕНТИРОВАННАЯ СИСТЕМА КЛАСТЕРИЗАЦИИ И ПОЛНОТЕКСТОВОГО ПОИСКА ПРОЕКТНЫХ ДОКУМЕНТОВ	219
Панкова Л.А., Пронина В.А. НЕЧЁТКИЕ СЕМАНТИЧЕСКИЕ МОДЕЛИ ТЕКСТОВОГО ПОИСКА	225
Липницкий С.Ф., Мамчич А.А., Степура Л.В. ПОИСК И РЕФЕРИРОВАНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ В МНОГОЯЗЫЧНОЙ СРЕДЕ	229
Хорошевский В.Ф. АВТОМАТИЗАЦИЯ ПРОЦЕССОВ ВЫЯВЛЕНИЯ ТЕХНОЛОГИЧЕСКИХ ТРЕНДОВ В СИСТЕМЕ АРМ «ТРЕНД»	233
Ефименко И.В. ГИБРИДНЫЙ ПОДХОД К ВЫЯВЛЕНИЮ КОМПЛЕКСНЫХ ОБЪЕКТОВ В ОБЛАСТИ НАУЧНО-ТЕХНИЧЕСКОГО ПРОГНОЗИРОВАНИЯ: ПРИНЦИП «ЧЕРНОГО ЯЩИКА»	241
Массель Л.В., Массель А.Г. СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ НА ОСНОВЕ ИНТЕГРАЦИИ ОНТОЛОГИЧЕСКОГО, КОГНИТИВНОГО И СОБЫТИЙНОГО МОДЕЛИРОВАНИЯ	247
Прокопчук Ю.А. МОДЕЛИ СПОНТАННОЙ КОГНИТИВНОЙ ДИНАМИКИ	251
Шалфеева Е.А. СЕМАНТИЧЕСКИЕ МОДЕЛИ ПРЕДСТАВЛЕНИЯ РЕШАТЕЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ	258
Загорулько Г.Б., Загорулько Ю.А. ПОДХОД К ИНТЕГРАЦИИ РАЗНОРОДНЫХ МЕТОДОВ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ СЛОЖНЫХ ЗАДАЧ	265
Шункевич Д.В. МОДЕЛИ И СРЕДСТВА КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ МАШИН ОБРАБОТКИ ЗНАНИЙ НА ОСНОВЕ СЕМАНТИЧЕСКИХ СЕТЕЙ	269
Моросанова Н.А. АВТОМАТИЗАЦИЯ ПОИСКА ТРАНСФОРМАЦИЙ СХЕМ ОЦЕНКИ УВЕРЕННОСТИ	281
Комарцова Л.Г. ИНТЕГРИРОВАННЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ В ГИБРИДНЫХ СИСТЕМАХ	285
Найдёнова К.А., Пархоменко В.А. НЕЙРОПОДОБНАЯ КОМБИНАТОРНАЯ СТРУКТУРА ДАННЫХ ДЛЯ АЛГОРИТМОВ СИМВОЛЬНОГО МАШИННОГО ОБУЧЕНИЯ	289

Анищенко И.С., Жукевич А.И., Родченко В.Г. ПРИМЕНЕНИЕ КЛАСТЕРНЫХ СТРУКТУР ДЛЯ ПРЕДСТАВЛЕНИЯ КЛАССОВ ПРИ ПОСТРОЕНИИ СИСТЕМ РАСПОЗНАВАНИЯ	295
Виноградов Г.П., Кузнецов В.Н. ИТЕРАЦИОННЫЕ МЕТОДЫ СОГЛАСОВАННОЙ ОПТИМИЗАЦИИ С ИДЕНТИФИКАЦИЕЙ ЭЛЕМЕНТОВ МОДЕЛИ ВЫБОРА	299
Мутовкина Н.Ю., Ключин А.Ю., Кузнецов В.Н. СЕМАНТИЧЕСКОЕ ОПРЕДЕЛЕНИЕ ТИПА АГЕНТА В МНОГОАГЕНТНОЙ СИСТЕМЕ. ПРОБЛЕМА МЕЖАГЕНТНОГО ВЗАИМОДЕЙСТВИЯ	309
СЕКЦИЯ 5. СЕМАНТИЧЕСКИЕ МОДЕЛИ, СРЕДСТВА И МЕТОДЫ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	317
Гецэвіч Ю.С., Скопінава А.М. КАМΠΑНАНТЫ ІДЭНТЫФІКАЦЫІ КОЛЬКАСНЫХ ВЫРАЗАЎ З АДЗІНКАМІ ВЫМЯРЭННЯ Ў ТЭКСТАХ НА БЕЛАРУСКАЙ І РУСКАЙ МОВАХ	319
Сулейманов Д.Ш., Гатиатуллин А.Р., Вагапов Д.Р. СЕМАНТИКО-СИНТАКСИЧЕСКАЯ МОДЕЛЬ ТАТАРСКОГО ПРЕДЛОЖЕНИЯ В КОНТЕКСТЕ РЕЛЯЦИОННО-СИТУАЦИОННОЙ СИСТЕМЫ	329
Килеев В.В., Сидоркина И.Г. МЕТОДЫ ВЕРИФИКАЦИИ ГРАММАТИКИ ЕСТЕСТВЕННЫХ ЯЗЫКОВ ФИННО-УГОРСКОЙ ГРУППЫ НА УРОВНЕ СЕМАНТИЧЕСКОГО ПРЕДСТАВЛЕНИЯ	333
Шарипбаев А.А., Бекманова Г.Т., Муканова А.С., Ергеш Б.Ж. ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ МОРФОЛОГИЧЕСКИХ ПРАВИЛ КАЗАХСКОГО ЯЗЫКА В ВИДЕ СЕМАНТИЧЕСКИХ ГИПЕРГРАФОВ	337
Заболеева-Зотова А.В., Орлова Ю.А., Петровский А.Б. ПОДХОД К АВТОМАТИЗАЦИИ АНАЛИЗА И СИНТЕЗА ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ	341
Дмитриев А.С., Заболеева-Зотова А.В. ИДЕНТИФИКАЦИЯ ПРОСТРАНСТВЕННО-ВРЕМЕННЫХ СВЯЗЕЙ МЕЖДУ ВЫСКАЗЫВАНИЯМИ В ЗАДАЧАХ СЕМАНТИЧЕСКОГО АНАЛИЗА ТЕКСТА	345
Солошенко А.Н., Орлова Ю.А., Дмитриев А.С. АВТОМАТИЗАЦИЯ РЕФЕРИРОВАНИЯ НОВОСТНЫХ ИНТЕРНЕТ-ТЕКСТОВ	351
Константинов В.М., Розалиев В.Л., Дианов И.А. АВТОМАТИЗИРОВАННАЯ СИСТЕМА СОЗДАНИЯ РУКОПИСНЫХ ДОКУМЕНТОВ	357
Хусаинов А.Ф., Сулейманов Д.Ш. ПРОТОТИП ПЛАТФОРМЫ АНАЛИЗА РЕЧИ НА ТАТАРСКОМ ЯЗЫКЕ	361

Харламов А.А., Ермоленко Т.В. СЕМАНТИЧЕСКАЯ СЕТЬ ПРЕДМЕТНОЙ ОБЛАСТИ КАК ОСНОВА ДЛЯ ФОРМИРОВАНИЯ СЕТИ ПЕРЕХОДОВ ПРИ АВТОМАТИЧЕСКОМ РАСПОЗНАВАНИИ СЛИТНОЙ РЕЧИ	369
Гецэвіч Ю.С., Пакладок Д.А., Брэк Д.В. КАМПАНЕНТЫ ДЛЯ РОЗНЫХ ПЛАТФОРМАЎ СІНТЭЗАТАРА МАЎЛЕННЯ ПА ТЭКСЦЕ ДЛЯ ІНТЭЛЕКТУАЛЬНЫХ СІСТЭМ	375
Худойбердиев Х.А. О СИНТЕЗАТОРЕ ТАДЖИКСКОЙ РЕЧИ ПО ТЕКСТУ	383
Маджиднежад В.М., Хейдоров И.Э. ДИАГНОСТИКА ПАТОЛОГИИ ГОЛОСОВОГО ТРАКТА НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ	387
Пекарь Д.В. , Садов В.С. ОБОБЩЕННАЯ МОДЕЛЬ ЭМОЦИИ И ПРОЦЕССА АНАЛИЗА РЕЧЕВОГО СИГНАЛА В СИСТЕМЕ БЕЗОПАСНОСТИ	391
Алиев Р.М. ОПТИМИЗАЦИЯ ОКОННОГО ФУРЬЕ И НЕПРЕРЫВНОГО ВЕЙВЛЕТ ПРЕОБРАЗОВАНИЙ ДЛЯ СПЕКТРАЛЬНОГО АНАЛИЗА МУЗЫКАЛЬНЫХ СИГНАЛОВ	395
Дорофеев Н.С., Розалиев В.Л., Заболеева-Зотова А.В. СИСТЕМА РАСПОЗНАВАНИЯ ЖЕСТОВ РУССКОГО ЯЗЫКА ГЛУХИХ	399
Корончик Д. Н. УНИФИЦИРОВАННЫЕ СЕМАНТИЧЕСКИЕ МОДЕЛИ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ И ТЕХНОЛОГИЯ ИХ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ	403
Ветров Ю.А. ПРОБЛЕМЫ ПРОМЫШЛЕННОГО ПРОЕКТИРОВАНИЯ ИНТЕРФЕЙСОВ	407
Беляков С.Л., Белякова М.Л., Савельева М.Н. ИНТЕЛЛЕКТУАЛЬНАЯ ВИЗУАЛИЗАЦИЯ ПРИ АВТОМАТИЧЕСКОМ КАРТОГРАФИРОВАНИИ	413
Алексеев А. В., Шпирко А.А. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ СЦЕНЫ ДЛЯ ОКРАШИВАНИЯ ЧЕРНО-БЕЛЫХ ИЗОБРАЖЕНИЙ	417
СЕКЦИЯ 6. КОМПЛЕКСНЫЕ МЕТОДЫ, ОНТОЛОГИЯ И МЕНЕДЖМЕНТ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	421
Соснин П.И., Лапшов Ю.А., Маклаев В.А. ПРОГРАММНОЕ УПРАВЛЕНИЕ ПОТОКАМИ РАБОТ В КОНЦЕПТУАЛЬНОМ ПРОЕКТИРОВАНИИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ	423
Филатова Н.Н., Требухин А.Г. ТЕХНОЛОГИЯ ОТКРЫТОГО ПРОЕКТИРОВАНИЯ СХЕМ АВТОМАТИЗАЦИИ	433

Иванченко Н.А., Иванченко Г.Ф. СЕМАНТИКО-ОНТОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ ТЕХНИКО- ТЕХНОЛОГИЧЕСКОГО ПОТЕНЦИАЛА ПРЕДПРИЯТИЯ	437
Боргест Н.М., Коровин М.Д. ТРЕБОВАНИЯ К КОНСТРУКТОРУ ОНТОЛОГИЙ ДЛЯ МАШИНОСТРОЕНИЯ	441
Боргест Н.М., Коровин М.Д. ПОДХОДЫ К ИНТЕГРАЦИИ ПРОГРАММНЫХ СИСТЕМ В ИНТЕЛЛЕКТУАЛЬНОМ ПОМОЩНИКЕ ПРОЕКТАНТА	445
Боргест Н.М., Шустова Д.В., Одинцова С.А., Князихина Ю.Е. ПРОБЛЕМЫ СИНОНИМОВ В ТЕЗАУРУСЕ ИНТЕЛЛЕКТУАЛЬНОГО ПОМОЩНИКА ПРОЕКТАНТА	449
Жуков И.И., Гракова Н.В. СИСТЕМА УПРАВЛЕНИЯ ПРОЕКТИРОВАНИЕМ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	453
Додонов А.Г., Ландэ Д.В. ЖИВУЧЕСТЬ ИНФОРМАЦИОННЫХ ОБЪЕКТОВ В СЕТИ ИНТЕРНЕТ	457
Касторнов А.Ф. , Касторнова В.А. ТЕХНОЛОГИЯ КОМПОНЕНТНОГО (МОДУЛЬНОГО) ПРОЕКТИРОВАНИЯ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ СИСТЕМАТИЗАЦИИ И КОНТРОЛЯ ЗНАНИЙ	461
Бурдо Г.Б., Палюх Б.В., Мельникова В.В. АВТОМАТИЗИРОВАННАЯ СИСТЕМА ТЕСТИРОВАНИЯ ПРОГРАММНЫХ СРЕДСТВ НА ОСНОВЕ МОДИФИЦИРОВАННЫХ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ	467
СЕКЦИЯ 7. ЛОГИКО-СЕМАНТИЧЕСКИЕ МОДЕЛИ ПРИКЛАДНЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	471
Бибило П.Н., Романов В.И. ИСПОЛЬЗОВАНИЕ ПРОДУКЦИОННО-ФРЕЙМОВОЙ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В АВТОМАТИЗИРОВАННОМ ЛОГИЧЕСКОМ ПРОЕКТИРОВАНИИ ЗАКАЗНЫХ ЦИФРОВЫХ СБИС	473
Дёмин В.В., Кабыш А.С., Дунец И.П., Дунец А.П., Головкин В.А. ПРИМЕНЕНИЕ RAM-BASED СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ СТАЙНЫХ РОБОТОВ	477
Прокопович Г.А. НЕЙРОСЕТЕВАЯ МОДЕЛЬ ДЛЯ РЕАЛИЗАЦИИ ПОИСКОВЫХ ДВИЖЕНИЙ МОБИЛЬНОГО РОБОТА	483
Сычёв В.А., Гецевич Ю.С., Прокопович Г.А. КОМПОНЕНТЫ НАТУРАЛЬНО-РЕЧЕВОГО КОМПЛЕКСА ОПРЕДЕЛЕНИЯ ФОРМЫ ПРЕПЯТСТВИЯ ДЛЯ МОБИЛЬНЫХ РОБОТОВ	489
Заболеева-Зотова А.В., Бобков А.С., Петровский А.Б. МОДЕЛИРОВАНИЕ ДВИЖЕНИЙ ЧЕЛОВЕКА НА ОСНОВЕ НЕЧЕТКИХ ТЕМПОРАЛЬНЫХ ВЫКАЗЫВАНИЙ	497
Шпирко А.А., Дорофеев Н.С., Бобков А.С. РАСПОЗНАВАНИЕ ПОЗЫ ТЕЛА ЧЕЛОВЕКА С ПОМОЩЬЮ 3D КАМЕРЫ С СЕНСОРом ГЛУБИНЫ	501

Житко В.А., Гецевич Ю.С., Лобанов Б.М. СПРАВОЧНАЯ СИСТЕМА С РЕЧЕВЫМ ИНТЕРФЕЙСОМ	505
Елисеева О.Е., Русецкий К.В. КОМПОНЕНТНОЕ ПРОЕКТИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ ДЛЯ ПОДГОТОВКИ ШКОЛЬНИКОВ К ЦЕНТРАЛИЗОВАННОМУ ТЕСТИРОВАНИЮ ПО ИНОСТРАННЫМ ЯЗЫКАМ	511
Киселёв В.В., Елисеева О.Е. ИНТЕЛЛЕКТУАЛЬНАЯ ОБУЧАЮЩАЯ СИСТЕМА AI-TUTOR ДЛЯ ЭКСПРЕСС-ИЗУЧЕНИЯ ИНОСТРАННЫХ ЯЗЫКОВ В СЕТИ ИНТЕРНЕТ	517
Головня А.И. ИСПОЛЬЗОВАНИЕ АФФИКСАЦИИ В ЭКСПЕРТНЫХ ОБУЧАЮЩИХ СИСТЕМАХ ПРИ ИЗУЧЕНИИ СЛОВООБРАЗОВАНИЯ	521
Козлов О.А., Михайлов Ю.Ф. МЕТОД ИНТЕЛЛЕКТУАЛИЗАЦИИ ТЕСТИРОВАНИЯ ИНДИВИДУАЛЬНОЙ ТРАЕКТОРИИ ИЗУЧЕНИЯ СТУДЕНТОМ НЕКОТОРОЙ ПРЕДМЕТНОЙ ОБЛАСТИ ЗНАНИЙ НА ОСНОВЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ	529
Тельнов Ю.Ф., Казаков В.А., Трембач В.М. СОЗДАНИЕ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНОГО ПРОСТРАНСТВА ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ НА ОСНОВЕ СЕРВИСНО-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ И МНОГОАГЕНТНОЙ ТЕХНОЛОГИИ	533
Горбачёв Н.Н. РЕГУЛЯРНАЯ АКТУАЛИЗАЦИЯ УЧЕБНО-МЕТОДИЧЕСКОГО КОНТЕНТА С ИСПОЛЬЗОВАНИЕМ МЕТАДАННЫХ НА ОСНОВЕ МОДЕЛИРОВАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ	541
Шарипбаев А.А., Омарбекова А.С., Барлыбаев А.Б. СЕМАНТИЧЕСКАЯ МОДЕЛЬ ЭЛЕКТРОННОГО УНИВЕРСИТЕТА	547
Шарипбаев А.А., Омарбекова А.С., Кинтонова А.Ж., Ниязова Р.С., Барлыбаев А. ГЕНЕРАЦИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ЭЛЕКТРОННЫХ УЧЕБНЫХ ИЗДАНИЙ НА ОСНОВЕ ОНТОЛОГИЧЕСКОЙ МОДЕЛИ	551
Соловьев В.И. СЕТЕЦЕНТРИЧЕСКАЯ СМАРТ-КОБОРГ СИСТЕМА ДЛЯ ОПЕРАТИВНОГО ДИАГНОСТИРОВАНИЯ И УПРАВЛЕНИЯ	555
Лейченко А.Н. ИНТЕЛЛЕКТУАЛЬНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ПО СОВРЕМЕННОМУ ИСКУССТВУ	561
Бранцевич П.Ю., Ли И. ОРГАНИЗАЦИЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ОЦЕНКИ ТЕХНИЧЕСКОГО СОСТОЯНИЯ СЛОЖНЫХ МЕХАНИЗМОВ	569
Поттосин Ю.В., Поттосина С.А. ЗАДАЧИ НА ГРАФАХ В ТЕОРИИ ПРОЕКТИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ	573
Тагави С.А., Поттосин Ю.В. РАЗЛОЖИМОСТЬ И ПОИСК ВСЕХ РЕШЕНИЙ СИСТЕМЫ БУЛЕВЫХ ФУНКЦИЙ	579
АВТОРСКИЙ УКАЗАТЕЛЬ	585

TABLE OF CONTENTS

FOREWORD	19
-----------------	----

SECTION 1. CONSTRUCTION PRINCIPLES AND STRUCTURE OF COMPONENT DESIGN TECHNOLOGY OF INTELLIGENT SYSTEMS	23
---	----

Gribova V, Kleschev A. CLOUD AND SEMANTIC TECHNOLOGIES FOR INTELLIGENT SERVICES DEVELOPMENT	25
--	----

Zagorulko Yu.A. TECHNOLOGIES OF DEVELOPMENT OF INTELLIGENT SYSTEMS BASED ON THE INTEGRATED KNOWLEDGE REPRESENTATION MODEL	31
--	----

Anureev I.S., Atuchin M.M. OPEN TECHNOLOGY OF FORMAL LOGICAL DESIGN AND PROTOTYPING OF INTELLIGENT SYSTEMS	43
---	----

Globa L., Ternovoy M., Novogradskaya R. THE METHOD OF KNOWLEDGE PORTAL INFORMATION ELEMENTS ORGANIZATION	49
---	----

Golenkov V.V., Guliakina N.A. OPEN PROJECT, AIMED AT CREATION OF TECHNOLOGY OF COMPONENT DESIGN OF INTELLECTUAL SYSTEMS	55
--	----

SECTION 2. BASIC SEMANTIC MODELS OF INTELLIGENT SYSTEMS AND THEIR IMPLEMENTATIONS	79
--	----

Timchenko V.A. THE METHOD OF SEMANTIC NETWORK CLASSES TRANSFORMATION	81
---	----

Zhilyakova L.Yu. MODELLING OF THE ASSOCIATIVE MEMORY. COGNITIVE APPROACH	87
---	----

Nevzorova O.A., Kirillovich A.V. DATA BINDING TECHNOLOGY IN THE SPACE OF OPEN DATA ON THE EXAMPLE OF MATHEMATICAL COLLECTIONS	91
--	----

Lanin V.V., Lyadova L.N. RESEARCH PORTAL OF MODELING INFORMATION SYSTEMS	97
---	----

Galushka I.N., Zavgorodniy V. V., Soloshish S.N., Shcherbak S.S. PRACTICAL ASPECTS OF SEARCH INTERFACE IMPLEMENTATION FOR SEMANTIC APPLICATIONS BASED ON LINKED DATA	103
---	-----

Vishniakou U.A., Borodaenko J.V. SEMANTIC WEB-SERVICES FOR INTEGRATION DECISIONS	107
---	-----

Yatskevich M.I., Apanasovich V.V. SEMANTIC EVALUATION AT LARGE SCALE	111
---	-----

Kolb D.G., Furman O.D. DEVELOPMENT OF WEB-SITES WITH A COMPLEX CONTENT BASED ON SEMANTIC NETWORKS	115
Koronchik D. N. IMPLEMENTATION OF STORAGE FOR UNIFIED SEMANTIC NETWORKS	125
Verenik N.L., Tatur M.M. BUILDING INTELLIGENT SYSTEM ON VECTOR ASIP	129
SECTION 3. SEMANTIC MODELS, TOOLS AND METHODS OF COMPONENT DESIGN KNOWLEDGE BASES AND PROGRAM PACKAGES FOR THE PROCESSING OF KNOWLEDGE	133
Kulinich A. A. CONCEPTUAL "TEMPLATES" OF THE UNCERTAINTY SUBJECT DOMAINS	135
Kryukov K.V., Kuznetsov O.P., Suhoverov V.S. A CONCEPT OF FORMAL COMPETENCE FOR RESEARCH WORKERS	143
Malkovsky M.G., Soloviev S.Y. HIERARCHIAL RELATIONS IN TERMINOLOGICAL NETWORK	147
Alekseev A.A., Dobrov B.V., Loukachevitch N.V. LINGUISTIC ONTOLOGY – RUTHES THESAURUS	153
Zagorulko M.Yu., Sidorova E.A. TOOLS FOR LANGUAGE RESOURCES SOFTWARE DEVELOPMENT	159
Zuenko A.A., Kulik B.A., Fridman A.Ya. INTELLIGENT TEACHING SYSTEMS BASED ON ALGEBRAIC REPRESENTATION OF QUESTION-AND-ANSWER TEXTS	165
Ivashenko V.P. UNIFIED KNOWLEDGE REPRESENTATION AND KNOWLEDGE INTEGRATION	171
Davydenko I.T. COMPONENT TECHNOLOGY OF DESIGN OF KNOWLEDGE BASES BASED ON UNIFIED SEMANTIC NETWORKS	185
Smirnov S.V. PROGRAM PACKAGES AS A FORMAL ONTOLOGIES:CONSTRUCTION AND APPLICATION	191
Kovartsev A.N., Zhidchenko V.V., Popova-Kovartseva D.A., Abolmasov P.V. THE BASICS OF GRAPH-SYMBOLIC PROGRAMMING TECHNOLOGY	195
Pivovarchyk O.V. SEMANTIC LANGUAGES USED FOR DESCRIPTION OF LANGUAGES PROGRAMMING, ORIENTED TO KNOWLEDGE TREATMENT	205

SECTION 4. SEMANTIC MODELS, TOOLS AND METHODS OF COMPONENT DESIGN PROBLEMS SOLVERS	217
Namestnikov A.M., Subkhangulov R.A., Filippov A.A. ONTOLOGICAL SYSTEM FOR CLUSTERING AND FULL-TEXT SEARCHING OF THE CAD DOCUMENTS	219
Pankova L.A., Pronina V.A. SEMANTIC TEXT RETRIEVAL BASED ON FUZZY SET THEORY	225
Lipnitsky S.F., Mamchich A.A., Stepura L.V. RETRIEVAL AND SUMMARIZATION OF TEXT INFORMATION IN A MULTILINGUAL ENVIRONMENT	229
Khoroshevsky V. F. TECHNOLOGY TRENDS WATCHING IN AUTOMATED WORKSTATION “TREND”	233
Efimenko I. V. HYBRID APPROACH FOR IDENTIFYING COMPEX CONCEPTS IN TECHNOLOGICAL FORESIGHT: BLACKBOX MODEL	241
Massel L.V., Massel A.G. SEMANTIC TECHNOLOGIES BASED ON THE INTEGRATION OF THE ONTOLOGY, COGNITIVE AND EVENT MODELING	247
Prokopchuk I.A. MODEL OF SPONTANEOUS COGNITIVE DYNAMICS	251
Shalfeeva E. THE SEMANTIC MODELS FOR REPRESENTATION OF INTELLIGENT SOFTWARE’ PROBLEM SOLVER	257
Zagorulko G.B., Zagorulko Yu.A. APPROACH TO INTEGRATION OF HETEROGENEOUS METHODS OF DECISION SUPPORT FOR COMPLICATED PROBLEMS	265
Shunkevich D.V. MODELS AND MEANS OF KNOWLEDGE PROCESSING MACHINES COMPONENT DESIGN ON BASIS OF SEMANTIC NETWORKS	269
Morosanova N.A. AUTOMATED SEARCH FOR UNCERTAIN REASONING MODELS TRANSFORMATIONS	281
Komartsova L.G. INTEGRATED INTELLIGENT TECHNOLOGY IN HYBRID SYSTEMS	285
Naidenova X., Parkhomenko V. A NEURAL NETWORK-LIKE COMBINATORIAL DATA STRUCTURE FOR SYMBOLIC MACHINE LEARNING ALGORITHMS	289

Anishchenko I.S., Zhukevich A.I., Rodchenko V.G. THE USE OF CLUSTER STRUCTURES TO REPRESENT OF CLASSES FOR CONSTRUCTION OF RECOGNITION SYSTEMS	295
Vinogradov G.P., Kuznetsov V.N. ITERATIVE METHODS OF OPTIMIZATION WITH COORDINATED IDENTIFICATION ELEMENOTOV CHOICE MODELS	299
Mutovkina N.Yu, Klyushin A.Yu., Kuznetsov V.N. SEMANTIC DEFINITION OF THE TYPE OF AGENT IN MULTI-AGENT SYSTEM.THE PROBLEM OF INTERACTION BETWEEN AGENTS	309
SECTION 5. SEMANTIC MODELS, TOOLS AND METHODS OF COMPONENT DESIGN USER INTERFACES OF INTELLIGENT SYSTEMS	317
Hetsevich Y.S., Skopinava A.M. COMPONENTS FOR IDENTIFICATION OF QUANTITIVE EXPRESSIONS WITH MEASUREMENT UNITS IN BELARUSIAN AND RUSSIAN TEXTS	319
Suleymanov D. Sh., Gatiatullin A. R., Vagapov D.R. SEMANTICO-SYNTACTIC MODEL OF THE TATAR SENTENCE IN THE CONTEXT OF RELATIONAL-SITUATIONAL SYSTEM	329
Kileev V.V., Sidorkina I.G. NATURAL FINNO-UGRIC LANGUAGE GRAMMAR VERIFICATION METHODS AT THE LEVEL OF SEMANTIC REPRESENTATION	333
Sharipbayev A.A., Bekmanova G.T., Mukanova A.S., Yergesh B.Zh. ONTOLOGICAL MODELS OF MORPHOLOGICAL RULES OF KAZAKH LANGUAGE IN THE FORM OF SEMANTIC HYPERGRAPHS	337
Zaboleeva-Zotova A.V., Orlova Y.A., Petrovsky A.B. APPROCH TO AUTOMATION OF ANALYSIS AND SYNTHESIS OF TECHNICAL DOCUMENTATION	341
Dmitriev A.S., Zaboleeva-Zotova A.V. IDENTIFICATION OF SPACE-TEMPORAL CONNECTIONS BETWEEN STATEMENTS IN PROBLEMS SEMANTIC ANALYSIS TEXT	345
Soloshenko A.N., Orlova Y.A., Dmitriev A.S. AUTOMATIZATION OF INTERNET NEWS TEXT ABSTRACTING	351
Konstantinov V.M., Rozaliev V.L., Dianov I.A. RECOGNITION OF SIGN RUSSIAN LANGUAGE GESTURES	357
Khusainov A.F., Suleymanov D.S. SPEECH ANALYSIS PLATFORM PROTOTYPE FOR TATAR LANGUAGE	361

Kharlamov A.A., Yermolenko T.V. SUBJECT AREA SEMANTIC NETWORK AS A BASIS FOR COMPILING TRANSITION NETWORK IN AUTOMATIC SPEECH RECOGNITION	369
Hetsevich Y.S., Pakladok D.A., Brek D.V. COMPONENTS FOR DIFFERENT PLATFORMS OF TEXT-TO-SPEECH SYNTHESIZERS FOR INTELLECTUAL SYSTEMS	375
Khudoyberdiev Kh. A. ON A TAJIK SPEECH SYNTHESIZER UNDER A TEXT	383
Majidnezhad V., Kheidorov I. AN ANN-BASED METHOD FOR VOCAL FOLD PATHOLOGY DIAGNOSIS	387
Pekar D. V. , Sadov V. S. GENERALIZED MODEL OF HUMAN EMOTION AND PROCESS FLOW OF SPEECH SIGNAL ANALYSIS IN SECURITY SYSTEM	391
Aliyev R.M. OPTIMIZATION OF SHORT-TIME FOURIER AND CONTINUOUS WAVELET TRANSFORMS FOR SPECTRAL ANALYSIS OF MUSICAL SIGNALS	395
Dorofeev N.S., Rozaliev V.L., Zaboлева-Zotova A.V. SYSTEM FOR RUSSIAN SIGN LANGUAGE RECOGNITION	399
Koronchik D. N. UNIFIED SEMANTIC MODELS OF USER INTERFACE FOR INTELLIGENT SYSTEMS AND TECHNOLOGY FOR THEIR DEVELOP	403
Vetrov Y.A. CURRENT PROBLEMS OF INDUSTRIAL USER EXPERIENCE DESIGN DISCIPLINE	407
Belyakov S.L., Belyakova M.L., Savelyeva M.N. INTELLECTUAL IMAGING IN AUTOMATIC MAPPING	413
Alekseev A.V., Shpirko A.A. INTELLIGENT SCENE ANALYSIS FOR COLORING BLACK AND WHITE IMAGES	417
SECTION 6. COMPREHENSIVE METHODS, ONTOLOGY AND MANAGEMENT OF COMPONENT DESIGN OF INTELLIGENT SYSTEMS	421
Sosnin P.I., Lapshov Y.A., Maklaev V.A. PROGRAM MANAGEMENT OF WORKFLOWS IN CONCEPTUAL DESIGNING OF AUTOMATED SYSTEMS	423
Filatova N.N., Trebukhin A.G. TECHNOLOGY OPEN DESIGNING OF CIRCUITS AUTOMATION	433

Ivanchenko N., Ivanchenko G. SEMANTIC - ONTOLOGICAL DESIGN OF TECHNICAL - TECHNOLOGICAL POTENTIAL OF ENTERPRISE	437
Borgest N.M., Korovin M.D. REQUIREMENTS FOR ONTOLOGY CONSTRUCTORS IN THE ENGINEERING PLANT SUBJECT AREA	441
Borgest N.M., Korovin M.D. APPROACHES TO THE INTEGRATION OF SOFTWARE SYSTEMS IN THE INTELLECTUAL DESIGNER ASSISTANT	445
Borgest N.M., Shustova D.V., Odintsova S.A., Knyazihina Y.E. PROBLEMS OF SYNONYMS IN THE THESAURUS OF INTELLIGENT ASSISTANT OF DESIGNER	449
Zhukau I.I., Grakova N.V. SYSTEM OF PROJECT MANAGEMENT FOR INTELIGENT SYSTEMS	453
Dodonov A.G., Lande D.V. SURVIVABILITY OF INFORMATION OBJECTS IN THE INTERNET	457
Kastornov A.F., Kastornova V.A. TECHNOLOGY OF COMPONENT (MODULAR) DESIGN OF EXPERT SYSTEM FOR SYSTEMATIZATION AND CONTROL OF KNOWLEDGE	461
Burdo G.B., Palyukh B.V., Melnikova V.V. AUTOMATED SYSTEM OF TESTING OF SOFTWARE ON THE BASIS OF THE MODIFIED GENETIC ALGORITHMS	457
SECTION 7. LOGICO-SEMANTIC MODEL OF APPLIED INTELLIGENT SYSTEMS	471
Bibilo P.N., Romanov V.I. APPLICATION OF THE PRODUCTION RULE MODEL FOR KNOWLEDGE REPRESENTATION IN AUTOMATION LOGICAL DESIGN OF CUSTOM DIGITAL	473
Demin V.V., Kabush A.S., Dunets I.P., Dunets A.P., Golovko V.A. APPLICATION RAM-BASED NETWORK FOR SWARM ROBOTICS DETECTION	477
Prakapovich R.A. NEURAL NETWORK MODEL FOR MOBILE ROBOT SEARCH MOVEMENTS	483
Sychou U.A., Hetsevich Yu.S., Prakapovich R.A. COMPONENTS OF THE NATURAL SPEECH COMPLEX FOR IDENTIFICATION OF DIFFERENT TYPES OF OBSTACLES FOR MOBILE ROBOTS	489
Zaboleeva-Zotova A.V., Bobkov A.S., Petrovsky A.B. MODELING HUMAN MOVEMENTS BASED ON THE FUZZY TEMPORAL STATEMENTS	497
Shpirko A.A., Dorofeev N.S., Bobkov A.S. RECOGNITION OF BODY POSTURES USING A 3D CAMERA WITH THE DEPTH SENSOR	501

Zhitko V.A., Hetsevich Y.S., Lobanov B.M. HELP SYSTEM WITH SPEECH USER INTERFACE	505
Yeliseyeva O.E., Rusetski K.V. COMPONENT DESIGN OF INTELLIGENT TUTORING SYSTEM TO PREPARE STUDENTS FOR CENTRALIZED TESTING IN A FOREIGN LANGUAGE	511
Kiselev V.V., Yeliseyeva O.E. INTELLIGENT TUTORING SYSTEM AI-TUTOR FOR RAPID LEARNING OF FOREIGN LANGUAGES IN THE INTERNET	517
Golovnja A.I. USING OF AFFIXATION IN EXPERT TRAINING SYSTEMS DURING LEARNING OF WORD-FORMATION	521
Kozlov O.A., Mikhailov J.F. METHOD OF INTELLECTUALIZATION OF TESTING INDIVIDUAL TRAJECTORY OF STUDY OF THE STUDENT OF A CERTAIN DOMAIN OF KNOWLEDGE ON THE BASIS OF ARTIFICIAL НИРОННЫХ NETWORKS	529
Telnov Yu.F., Kazakov V.A., Trembach V.M. CREATING OF INFORMATION-EDUCATIONAL SPACE BASED ON SERVICE- ORIENTED ARCHITECTURE AND MULTI-AGENT TECHNOLOGY	533
Gorbachev N.N. PERMANENT UPDATING OF EDUCATIONAL CONTENT USING KNOWLEDGE DOMAIN BASED METADATA	541
Sharipbayev A.A., Omarbekova A.S., Barlybayev A.B. SEMANTIC MODEL OF AN INTELLECTUAL E-UNIVERSITY	547
Sharipbaev A.A., Omarbekova A.S., Kintonova A.J., Niyazova R.S., Barlybayev A. GENERATION OF INTELLECTUAL ELECTRONIC EDUCATIONAL EDITIONS ON THE BASIS OF ONTOLOGIC MODEL	551
Soloviev V.I. NETWORK-CENTRIC COBORG-SYSTEM FOR OPERATIONAL DIAGNOSIS AND CONTROL	555
Leichanka N. INTELLIGENT INFORMATION SYSTEM ON MODERN ART	561
Brancevich P.J., Li.Y. ORGANIZATION OF INTELLECTUAL SYSTEM OF THE ASSESSMENT TECHNICAL CONDITIONS COMPOSITE MECHANISMS	569
Pottosin Yu.V., Pottosina S.A. THE PROBLEMS ON GRAPH IN THE THEORY OF DESIGN OF DIGITAL DEVICES	573
Taghavi S.A., Pottosin Y.V. DECOMPOSIBILITY AND SEARCH FOR ALL SOLUTIONS OF A SYSTEM OF BOOLEAN FUNCTIONS	579
AUTHOR INDEX	587

ПРЕДИСЛОВИЕ

Основным практическим результатом исследований в области искусственного интеллекта является создание не только интеллектуальных систем, но и технологий, обеспечивающих быстрое и качественное построение таких систем. Разработка указанных технологий требует решения следующих задач:

- чёткого выделения логико-семантического уровня интеллектуальных систем, который абстрагируется от всевозможных вариантов технической реализации этих систем (в том числе и от использования принципиально новых компьютеров, ориентированных на аппаратную поддержку интеллектуальных систем);
- разработки онтологии проектирования интеллектуальных систем и унификации описания логико-семантических моделей интеллектуальных систем;
- обеспечения платформенно независимого характера логического проектирования интеллектуальных систем, результатом которого является унифицированное описание логико-семантических моделей проектируемых интеллектуальных систем;
- использования методики компонентного проектирования интеллектуальных систем, в основе которой лежит постоянно пополняемая библиотека многократно используемых компонентов интеллектуальных систем (многократно используемых подсистем, знаний, машин и операций обработки знаний, компонентов пользовательских интерфейсов);
- обеспечения семантической совместимости многократно используемых компонентов интеллектуальных систем, входящих в состав указанной библиотеки.

Основной целью ежегодных конференций OSTIS (Open Semantic Technology for Intelligent Systems) является создание условий для расширения сотрудничества различных научных школ, вузов и коммерческих организаций, направленного на разработку и применения комплексной массовой и постоянно совершенствуемой технологии компонентного проектирования интеллектуальных систем.

Конференцию OSTIS 2013-го года организаторы посвящают 80-летию Дмитрия Александровича Поспелова – первого Президента Российской ассоциации искусственного интеллекта.

Председатель Программного комитета конференции OSTIS-2013.

Председатель Совета Российской ассоциации искусственного интеллекта

Кузнецов Олег Петрович

FOREWORD

Creation not only intelligent systems but also technologies that ensure fast and efficient construction of intelligent systems are the main practical result of research in artificial intelligence. Development of these technologies requires the following tasks:

- precise separation of logical and semantic level of intelligent systems, which is abstracted from various versions of the technical implementation of these systems (including the use of innovative computer-based hardware support for Intelligent Systems);
- developing an ontology design of intelligent systems and unifying description of logical and semantic models of intelligent systems;
- provide a platform independent nature of the logical design of intelligent systems, which result is a unified description of logical and semantic models of an intelligent systems;
- use of a component design methodology of intelligent systems, which is based on permanently increasing library of reusable components of intelligent systems (reusable subsystems, knowledge, machine and operations of knowledge processing, user interface components)
- ensuring semantic compatibility of reusable components of intelligent systems that make up the library

Creating the conditions for the expansion of cooperation between different scientific schools, universities and business organizations, aimed on development and using of comprehensive mass and continuously improved component design technology of intelligent systems is the main purpose of annual conferences OSTIS (Open Semantic Technology for Intelligent Systems).

Conference OSTIS-2013 is dedicated to the 80th anniversary of Dmitry Alexandrovich Pospelov – the first President of the Russian Association for Artificial Intelligence

Programme Committee Chair.

Chairman of the Council of the Russian Association for Artificial Intelligence

Kuznetsov Oleg Petrovich

СЕКЦИЯ 1.

**ПРИНЦИПЫ ПОСТРОЕНИЯ И СТРУКТУРА ТЕХНОЛОГИИ
КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ
СИСТЕМ**

SECTION 1.

**CONSTRUCTION PRINCIPLES AND STRUCTURE OF COMPONENT
DESIGN TECHNOLOGY OF INTELLIGENT SYSTEMS**



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОБЛАЧНЫЕ СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СЕРВИСОВ

Грибова В.В., Клещев А.С.

Федеральное государственное бюджетное учреждение науки Институт автоматизации и процессов управления Дальневосточного отделения Российской академии наук, г. Владивосток, Россия

`gribova@iacp.dvo.ru,`

`kleshev@iacp.dvo.ru`

В работе описана технология разработки и сопровождения интеллектуальных систем как интеллектуальных сервисов. В основе предлагаемой технологии разработки интеллектуальных систем лежит принцип четкого разделения между декларативными знаниями (знаниями предметной области) и процедурными (о методе решения задачи). Описаны принципы формирования каждого компонента интеллектуальной системы - базы знаний и данных, решателя задач и пользовательского интерфейса.

Ключевые слова: интеллектуальные системы; облачные технологии; семантические технологии; онтологии; базы знаний.

ВВЕДЕНИЕ

Разработка интеллектуальных систем и систем обработки сложно-структурированной информации, является трудоемким процессом, но еще более сложным и трудоемким является сопровождение таких систем. Поэтому исследование методов, средств, технологии разработки и сопровождения остается одной из актуальных задач.

В настоящее время можно выделить две ключевые технологии разработки таких систем: первая технология сводится к тому, что разработчику предлагается готовый решатель, и, в соответствии с представлением информации, на которое он рассчитан, он должен сформировать базу знаний, при этом часть знаний о решении задачи встроить в нее. Эту технологию поддерживают оболочки интеллектуальных систем. В качестве способа представления знаний используются базы правил. Первый опыт их применения показал, что базы знаний, основанные на правилах, исключительно сложно не только разрабатывать, но, прежде всего, сопровождать [Bachant et al., 1984], поскольку эксперты не понимают такое представление. Разработанные подходы, стратегии, методы и процедуры работы с экспертами по извлечению знаний позволили упростить их создание, однако модификация, расширение знаний в процессе эксплуатации системы, осталась трудоемким процессом, для чего такие системы не предназначены, что было отмечено еще в [Compton,

1988], поскольку метод решения задачи частично содержится в базе знаний.

Вторая технология исходит от обратного принципа – при построении интеллектуальных систем наиболее сложным для создания и сопровождения компонентом является база знаний и любая другая сложно-структурированная информация. Поэтому она должна разрабатываться так, чтобы быть повторно-используемой в различных интеллектуальных системах. Этот подход предполагает сначала формирование метаструктуры или онтологии, по которой может быть создан класс различных баз знаний либо другой сложно-структурированной информации, ей соответствующий. Решатель задачи основан на онтологии обрабатываемой информации и, таким образом, может быть повторно использован для различных интеллектуальных систем, имеющих ту же онтологию обрабатываемой информации. Основные преимущества такой технологии – процедурные знания отделены от декларативных, решатель задач является повторно-используемым для класса задач с общей онтологией обрабатываемой информации. Данная технология применена в [Gennari et al., 2003, Клещев и др., 2006a]. В основе технологии Протеже лежит объектно-ориентированный подход к формированию онтологий. В онтологии определяется иерархия классов для представления основных ее объектов, множество слотов для описания их свойств и отношений между ними, а также множество экземпляров классов. Решатель задач состоит из трех основных компонентов:

описания алгоритма, включающее все шаги решения задачи, описания так называемой онтологии метода, которое включает спецификации форматов входных и выходных данных, а также непосредственно программного кода, реализующего метод. Пользовательский интерфейс входит в метод решения задачи. В основе метода решения лежит утверждение, выдвинутое еще Чандрасекараном о том, что в интеллектуальных системах можно выделить классы задач, которые они решают, и для каждого класса задач разработать метод ее решения, который будет повторно использоваться в различных системах (методы могут быть проблемно-зависимыми).

Однако в данном подходе отсутствует четкое разделение между двумя системами понятий. Такой подход, с одной стороны, «позволяет быстро и относительно просто сконструировать небольшую предметную онтологию», однако отсутствует четкое разделение между базой знаний и онтологией [Пивоварова и др., 2007]. Возможность разработки общих для разных онтологий методов решения интеллектуальных задач также не представляется универсальным механизмом, несмотря на отдельные удачные примеры, представленные в работах.

В работах [Клещев и др., 2006б] также используется идеология полного разделения процедурных и декларативных знаний, принцип формирования базы знаний по онтологии. Существенным различием является четкое разделение между уровнями и, таким образом, разделение труда между инженерами по знаниям (формируют онтологию через интерфейс, ориентированный на метаязык) и экспертами (формируют базу знаний через сгенерированный по онтологии интерфейс).

Целью данной работы является описание дальнейшего развития технологии – методов формирования каждого компонента интеллектуальной системы - базы знаний и данных, решателя задач и пользовательского интерфейса.

1. Информационные ресурсы

Интеллектуальные системы имеют специфический архитектурный компонент – базу знаний, поэтому основной частью технологического процесса создания и сопровождения интеллектуальных систем является разработка и модификация базы знаний. При этом технология должна быть такой, чтобы эксперты могли формировать и сопровождать базы знаний любой степени сложности самостоятельно, без посредников. Задача разработки такой технологии может быть разбита на две подзадачи:

- отделение терминологии инженера по знаниям от терминологии эксперта, создание методов формирования и сопровождения баз знаний, ориентированных на экспертов, в понятной им терминологии;

- автоматизация разработки редакторов баз знаний, поскольку редакторы баз знаний являются сложными программными средствами; очевидно, что практически невозможно создавать редактор под формирование каждой базы знаний, поэтому необходимы методы и средства автоматизации разработки редакторов баз знаний, ориентированных на экспертов.

Рассматривается двухуровневая модель информационных ресурсов (под информационными ресурсами в данной работе понимаются базы знаний, базы данных и вообще любая сложным образом организованная информация), предложенная в работе [Клещев и др., 2006б], в соответствии с которой информация порождается или модифицируется под управлением метаинформации. Таким образом, в соответствии с данной моделью любая информация связана с метаинформацией, по которой она порождается или модифицируется. Метаинформация информационного ресурса является представлением графовой грамматики (грамматики, порождающей размеченные графы) языка представления информации.

Метаинформация (МИ) представляет собой пару $MI=(GM, \sigma M)$, где GM – граф понятий для описания информации (баз знаний и данных), возможно, содержащий циклы и петли, а σM – разметка этого графа. Граф GM есть тройка $\langle Vertexes, Arcs, RootVertex \rangle$, где $Vertexes$ – множество вершин графа, $Arcs$ – множество дуг графа, $RootVertex$ – начальная вершина графа. Множество вершин графа $Vertexes = \{Vertex_i\}_{i=1}^{vertexescount}$ состоит из двух непересекающихся подмножеств – терминальных вершин $Terminal_Vertexes$ и нетерминальных вершин $Neterminal_Vertexes$. Начальной вершиной графа $RootVertex$ является нетерминальная вершина, $RootVertex \in Neterminal_Vertexes$.

Множество дуг графа $Arcs$ есть множество пар $\{Arc_i\}_{i=0}^{arcscount}$, $Arc_i = \langle VertexFrom_i, VertexTo_i \rangle$, где $VertexFrom_i$, $VertexTo_i$ – вершины из которых выходит и в которые входит дуга Arc_i , соответственно, $VertexFrom_i \in Neterminal_Vertexes$, $VertexTo_i \in Vertexes$.

Средства разметки σM включают в себя разметку вершин σV и разметку дуг σA . Эта разметка позволяет при описании графа метаинформации задать ограничения на структуру и содержание графа информации.

Разметка вершин σV задается следующим образом: начальная вершина отображается в имя графа метаинформации, т.е. $\sigma V(RootVertex) = \text{Имя Графа Метаинформации}$; нетерминальные вершины графа отображаются в пару: Имена понятий предметной области либо класса предметных областей, имеющих одну и ту же метаинформацию; Тип структурированного набора,

т.е. $\sigma V(\text{Neterminal_Vertexes}) \in (\text{Имена понятий предметной области} \times \text{Тип структурированного набора})$. Назовем множество дуг графа, выходящих из одной вершины, структурированным набором дуг. Граф метаинформации имеет два возможных типа структурированных наборов - непустое множество и непустое множество альтернатив, т.е. Тип структурированного набора $\in \{\text{Непустое множество, Непустое множество альтернатив}\}$.

Терминальные вершины могут иметь разметку двух типов - терминал, описывающий сорт, и терминал, описывающий значение. Терминал, описывающий сорт, есть пара Терминал, описывающий Сорт_i=<Имя_i, Значение_i>, где Имя_i - строка символов конечной длины, Значение_i принадлежит множеству имен базовых сортов. Терминал, описывающий значение, есть тройка Терминал, описывающий Значение_i=<Имя_i, Сорт_i, Значение_i>, где Имя_i - строка символов конечной длины, Сорт_i \in Имена Базовых сортов, Значение_i \in Сорт_i. Разметка дуг σA задается следующим образом: $\sigma A(\text{Arc}_i) = \text{Спецификатор дуги}$. Спецификатор дуги есть пара <Тип спецификатора, Факультативность>, т.е. Спецификатор дуги=<Тип спецификатора, Факультативность>, Тип спецификатора $\in \{\text{копия, единственность, непустое множество}\}$. Все спецификаторы могут использоваться в сочетании со спецификатором «факультативность», однако спецификатор «факультативность» могут иметь лишь дуги, входящие в структурированный набор с типом «непустое множество».

Информация, как и метаинформация, представляет собой пару $I=(GI, \sigma I)$, где GI - граф (сеть) понятий, который в отличие от графа метаинформации, не содержит циклов и петель, σI - разметка графа. Граф информации GI есть тройка <IVertexes, IArcs, IRootVertex>, где множество вершин графа (IVertexes), множество дуг графа (IArcs), корневая вершина (IRootVertex) описываются также как в графе метаинформации, средства разметки σI , в отличие от графа метаинформации, включают в себя только разметку вершин, т.е. $\sigma I \equiv \sigma V$. Все вершины графа информации отображаются в пару $\sigma V=(SM, RM)$, где SM - служебная метка; RM - метка соответствия между информацией и метаинформацией. Служебные метки могут быть двух типов - порожденные вершины (вершины, из которых выполнен шаг порождения) и непорожденные (вершины, из которых еще не выполнен шаг порождения). Метка соответствия RM - это Имя понятия предметной области, которому соответствует либо Имя понятия предметной области, заданное в метаинформации, либо некоторое имя, входящее в класс предметных областей, также заданный в метаинформации.

Соответствие между информацией и метаинформацией задается следующим образом. Каждой вершине в графе информации соответствует

некоторая единственная вершина в графе метаинформации, называемая ее вершиной-прототипом. Количество вершин в графе информации определяется типом структурированного набора, заданного в разметке графа метаинформации. Так, если в разметке некоторой вершины-прототипа графа метаинформации структурированный набор задается типом «непустое множество», то в графе информации из вершины, соответствующей этой вершине-прототипу выходит i дуг, где $0 \leq i \leq N$, N - количество дуг, выходящих из вершины - прототипа в графе метаинформации; если в разметке вершины-прототипа структурированный набор задается типом «непустое множество альтернатив», то в графе информации из вершины, соответствующей этой вершине-прототипу, выходит единственная дуга.

Структура и разметка графа информации также зависит от типа спецификатора дуги метаинформации. Если в графе метаинформации дуга имеет Тип спецификатора= копия, то разметка вершины, в которую входит дуга со спецификатором данного типа, сохраняется в графе информации; если в графе метаинформации дуга имеет Тип спецификатора=единственность, то этой дуге в графе информации соответствует только одна дуга, входящая в вершину, разметка которой соответствует Имени предметной области (у нетерминалов) или значению (у терминалов); если в графе метаинформации дуга имеет Тип спецификатора=непустое множество, то этой дуге в графе информации соответствует множество дуг, входящих в вершины, разметка которых соответствует некоторому Имени предметной области (у нетерминалов) или значению (у терминалов). Факультативность, указанная в разметке дуги графа метаинформации означает, что в графе информации может отсутствовать понятие (термин) из метаинформации дуга, и, соответственно, вершина, в которую эта дуга входит.

Граф метаинформации можно рассматривать как порождающую графовую грамматику (грамматику, порождающую графы информации), определяющую язык, в терминах которого может быть порождено множество графов информации на этом языке.

Для компьютерного представления метаинформации и информации разработан метаязык (язык ИРУО), реализующий описанную выше модель, и универсальный редактор (редактор ИРУО) [Клещев и др., 2006b], имеющий два режима. Для формирования метаинформации на языке ИРУО используется режим инженера знаний. С помощью структурного редактора инженер знаний формирует на языке ИРУО размеченный граф метаинформации. По этому графу метаинформации редактор ИРУО в режиме эксперта генерирует интерфейс для эксперта (специалиста) предметной области. Информация, сформированная экспертом с помощью редактора ИРУО в режиме эксперта, также сохраняется на языке ИРУО. Таким

образом, редактор по сути является интерпретатором метаинформации, представленной на языке ИРУО, который формирует информацию также на языке ИРУО. Такой подход обеспечил реализацию единственного редактора как метаинформации, так и информации для любой метаинформации, представленной на языке ИРУО.

2. Решатель задач

Основной задачей при разработке решателей задач является создание универсальных методов их разработки из повторно-используемых компонентов в условиях полного отделения процедурных знаний (знаний о решении задач) от декларативных знаний (баз знаний) для снижения трудоемкости проектирования и сопровождения.

В предлагаемой концепции все информационные ресурсы имеют единое унифицированное представление – семантические сети; в этом случае решатель задач можно рассматривать как программу обработки информационных ресурсов, представленных таким образом. Результатом решения задачи является либо формирование нового информационного ресурса, либо модификация входного информационного ресурса

Вычислительный процесс, выполняемый решателем задач, состоит в построении графа выходной информации по графу выходной метаинформации и входным данным, если граф выходной информации неполностью порожден, либо в модификации этого графа, если он, как входные данные, полностью порожден (возможен и смешанный вариант).

Решатель задач представляет собой совокупность агентов. Агент - повторно используемый программный компонент, взаимодействующий с другими агентами посредством приема и передачи сообщений. Агент состоит из двух частей – декларативной и процедурной. Декларативная часть включает описание множества блоков. Каждый блок есть пара – метаинформация сообщения (шаблон сообщения) и множество продуктов. Метаинформация сообщения определяет язык, на котором представлена информация в сообщении, посылаемая этому блоку этого агента.

Продукция состоит из условия (антецедента) и действия (консеквента). Антецеденты продукции блока используются для анализа сообщения, выбора применимых методов и передачи им информации из сообщения; консеквент по сути является вызовом такого метода, который выполняет обработку информации, выбранной антецедентом из сообщения для решения связанной с агентом подзадачи. Для каждого блока продукции определен свой шаблон сообщения. Результатом работы метода может быть формирование и посылка сообщения конкретному агенту, либо подмножеству агентов, ограниченных некоторым условием, а также шаги формирования или модификации некоторых

информационных ресурсов.

Агенты могут быть двух основных типов: повторно-используемыми и специализированные. Повторно-используемыми агентами являются:

- проблемно-независимые вычислительные агенты, решающие стандартные часто-используемые вычислительные задачи;

- агенты, обрабатывающие информационные ресурсы; обрабатываемыми будем называть агенты, осуществляющие шаги порождения и модификации информационных ресурсов, а также запрос информации из них; повторная используемость агентов этого типа достигается за счет того, что все информационные ресурсы имеют единое унифицированное представление, а при порождении и модификации информационных ресурсов поддерживается соответствие между метаинформацией и информацией;

- интерфейсные агенты, осуществляющие взаимодействие с пользователями.

Специализированные агенты – это вычислительные агенты, решающие узкоспециализированные подзадачи, либо управляющие агенты (через которые осуществляется координация вычислений, выполняемых различными агентами).

В этом случае решатель задач – это совокупность повторно-используемых и, возможно, специализированных агентов. Единственное ограничение на состав агентов внутри решателя – они «должны знать» шаблоны сообщений агентов, которым они посылают и от которых они принимают сообщения. Поскольку решатель имеет формальные параметры (входные и выходные), описываемые через метаинформацию информационных ресурсов, каждый решатель применим к классу задач, связанных с этой метаинформацией.

Проектирование решателя задач состоит в декомпозиции основной задачи формирования или модификации информационных ресурсов на совокупность подзадач, решаемых агентами, с учетом ранее разработанных повторно-используемых агентов, и определении взаимосвязей между этими подзадачами.

Объединение агентов (подзадач) в решатель может осуществляться двумя способами:

- через посылку сообщений; сообщение может быть послано либо по обратному адресу, либо в сообщении указывается конкретный адресат;

- созданием специального информационного ресурса – управляющего графа, через который агенты связываются между собой.

Использование единого формата представления метаинформации и информации (язык ИРУО) позволило разработать конечное множество программных интерфейсов (функций), которые

обеспечивают универсальный доступ к информационным ресурсам для их обработки компьютерными программами (агентами). При этом они устроены таким образом, что поддерживается соответствие между информацией и метаинформацией. Использование стандартного набора программных интерфейсов освобождает программиста от дополнительных усилий (написание программного кода) для обеспечения доступа к представлению информационных ресурсов.

3. Пользовательский интерфейс

При разработке пользовательских интерфейсов для интеллектуальных систем необходимо разработать технологию их создания, удовлетворяющую следующим требованиям:

- реализация пользовательского интерфейса, как web-интерфейса, т.е. поддерживающего взаимодействие пользователя с интеллектуальной системой через браузер;
- поддержка WIMP-интерфейса;
- адаптация пользовательского интерфейса к типам пользователей и характеристикам входных/выходных данных.

Первое и второе требования обусловлены необходимостью создания интеллектуальных систем как интернет-сервисов; при этом важно сохранение традиционного и привычного для большинства пользователей WIMP-интерфейса. Третье требование обусловлено стандартами юзабилити, которые прописывают набор правил организации удобного в использовании и дружелюбного интерфейса для различных категорий пользователей в зависимости от характеристик входных/выходных данных.

Пользовательский интерфейс реализуется через повторно-используемых агентов. Для упрощения разработки пользовательского интерфейса в состав агентной платформы входит специальный агент «Вид». Агент «Вид» содержит реализации всех интерфейсных элементов, поддерживаемых текущей версией (текстовое поле, кнопка управления, поле ввода, переключатели – флажки и радио-кнопки, пролистываемый и раскрывающийся списки, комбобоксы, ссылки). Он осуществляет прорисовку элементов и реализует их поведение. Расширение множества поддерживаемых интерфейсных элементов возможно через модификацию агента «Вид». Интерфейсные элементы могут порождать различные события, вызванные действиями пользователя или системы (ввод символов с клавиатуры, выбор элемента, срабатывание системного таймера и др.). Поступившие события обрабатываются агентом «Вид», который формирует сообщения агенту решателя, далее это сообщение обрабатывается этим агентом и посылается новое сообщение либо другому агенту решателя, либо агенту «Вид»,

который формирует очередное состояние пользовательского интерфейса и отображает его через клиентское программное обеспечение (браузер).

Помимо агента «Вид» выделен набор интерфейсных агентов, осуществляющих адаптацию интерфейсной задачи к данным, с которыми она связана, т.е. интерфейсный агент выбирает адекватный интерфейсный элемент либо их группу, соответствующие данным, из множества реализованных в агенте «Вид».

В общем случае каждый интерфейсный агент во входном сообщении получает имя агента решателя, который посылает ему сообщение, и множество входных данных для выбора интерфейсного элемента; выходное сообщение интерфейсного агента, которое он посылает агенту «Вид», включает имя агента решателя, входные данные, интерфейсный элемент (элементы). Имя агента решателя необходимо для того, чтобы агент «Вид» посылал выходную информацию напрямую агенту решателя, минуя интерфейсный агент.

4. Заключение

В работе описана технология разработки и сопровождения интеллектуальных систем как интеллектуальных сервисов. В основе предлагаемой технологии разработки интеллектуальных систем лежит принцип четкого разделения между декларативными знаниями (знаниями предметной области) и процедурными знаниями (о методе решения задачи). Дальнейшее развитие получил двухуровневый подход к созданию баз знаний и сложно-структурированных баз данных, который предполагает, что вся информация представлена в виде неразрывной пары метаинформация-информация; информация формируется экспертом по метаинформации, созданной инженером по знаниям. В отличие от технологии Протеже, которая также предлагает формирование знаний по метаинформации, в предлагаемой технологии жестко разделены терминология инженера по знаниям и эксперта, что делает возможным формирование и сопровождение баз знаний и данных непосредственно носителями этой информации, т.е. экспертами и специалистами предметной области без посредников в лице инженеров знаний.

Поддержка процесса формирования и сопровождения информационных ресурсов обеспечивается универсальным редактором, который по метаинформации, созданной инженером по знаниям, генерирует пользовательский интерфейс для эксперта. В системе Протеже также имеется универсальный редактор знаний, управляемый метаинформацией, однако, по сути там предлагается общий редактор для инженеров по знаниям и экспертов, а отсутствие четкого разделения между их терминологиями делает практически невозможной самостоятельную работу

экспертов предметной области без участия в ней инженеров по знаниям. Дополнительным различием между редакторами Протеже и ИРУО является предоставление последнего как облачного сервиса.

Все информационные ресурсы имеют единый унифицированный формат представления – семантическую сеть. Это обеспечивает возможность доступа ко всем типам ресурсов (как данным, так и знаниям) через оболочку, предоставляющую набор программных интерфейсов, скрывающий формат их внутреннего представления, что значительно упрощает к ним доступ, поскольку разработчику не надо знать детали внутренней организации информации и программировать доступ к ней.

Решатель задач представляется в виде совокупности агентов, где каждому агенту соответствует подзадача решаемой задачи. Агент является повторно-используемым компонентом, представленным семантической сетью и связанным с ней множеством блоков продукции. В отличие от технологии Протеже, которая фиксирует классы задач, решаемых интеллектуальной системой, в данной технологии нет такого ограничения, каждый агент связан с уникальной подзадачей, которая может быть как проблемно-независимой (что расширяет круг ее использования), так и проблемно-ориентированной на класс задач, ограниченный некоторой метаинформацией (проблемно-ориентированный агент в общем случае связан с фрагментом метаинформации и, соответственно, может быть применен к обработке любого фрагмента информации, которая создается или модифицируется по тому фрагменту метаинформации, с которой связан агент).

Пользовательский интерфейс в отличие от Протеже, не является частью метода решения задачи. Он реализуется через системный агент «Вид», который является компонентом платформы и реализует поведение пользовательского интерфейса на основе входящих в него реализаций интерфейсных элементов, а также набора дополнительных интерфейсных агентов, которые обеспечивают адаптацию интерфейса к пользователю и наборам входных данных. Пользовательский интерфейс реализуется через браузер, что обеспечивает также его кроссплатформенность и возможность использования на всех компьютерах, включая мобильные устройства.

Платформой для разработки интеллектуальных интернет-сервисов является Интернет-комплекс IACPaaS (Intelligent Application, Control and Platform as a Service) [Грибова и др., 2011], предоставляющий контролируемый доступ и единую систему администрирования для создания интеллектуальных сервисов и их компонентов, представленных семантическими сетями и агентами (в том числе интерфейсными), поддержку функционирования сервисов и агентов (через передачу и обработку сообщений, запуск методов, а

также распараллеливания вычислений в сервисах).

В основе платформы лежит программно-информационный комплекс, основанный на технологии облачных вычислений и обеспечивающий удаленный доступ конечных пользователей к интеллектуальным системам, а разработчикам и управляющим – к средствам создания интеллектуальных систем и их компонентов, а также управления ими.

Работа выполнена при финансовой поддержке ДВО РАН (инициативный научный проект 12-I-ОНИТ-04) и при финансовой поддержке РФФИ (проект 12-07-00179-а).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Bachant et al., 1984] Bachant, J. McDermott, J. "R1 revisited: four years in the trenches", The AI Magazine (Fall 1984; Fall), pp.21-32
- [Compton, 1988] Compton, P. Horn, K. Quinlan, R. Lazarus, L. "Maintaining an expert system". Proceedings of the fourth Australian Conference on Applications of Expert Systems, (1988), pp.110-129.
- [Gennari et al., 2003] Gennari, J.H.; Musen, M.A.; Ferguson, R.W., etc. The evolution of Protégé: An environment for knowledge-based systems development/International Journal of Human-Computer Studies. 2003. 58(1):89-123
- [Грибова и др., 2011] Грибова В.В., Клещев А.С., Крылов и др. Проект IACPaaS – развиваемый комплекс для разработки, управления и использования интеллектуальных систем // Искусственный интеллект и принятие решений. – 2011. – №1.
- [Клещев и др., 2006а] Клещев А.С., Орлов В.А. Компьютерные банки знаний. Многоцелевой банк знаний// Информационные технологии, № 2, 2006. С. 2-8.
- [Клещев и др., 2006б] Клещев А.С., Орлов В.А. Компьютерные банки знаний. Универсальный подход к решению проблемы редактирования информации. – Информационные технологии. – 2006. – №5.
- [Пивоварова и др., 2007] Л. М. Пивоварова, В. Ш. Рубашкин Компоненты онтологических систем и их реализация в современных проектах//Интернет и современное общество: Труды X Всероссийской объединенной конференции. СПб.: Факультет филологии и искусств СПбГУ, 2007. с.277-279.

CLOUD AND SEMANTIC TECHNOLOGIES FOR INTELLIGENT SERVICES DEVELOPMENT

Gribova V, Kleschev A.

*Institute of Automation and Control Processes,
Far Eastern Branch of Russian Academy of
Sciences, Vladivostok
gribova@iacp.dvo.ru,
kleschev@iacp.dvo.ru*

A technology of development and maintenance of intelligent systems as intelligent services is presented. The proposed technology of intelligent system development based on the principle of clear separation between declarative knowledge (domain knowledge) and procedural (the method of solving the problem). The principles of the formation of each component of an intelligent system (knowledge base, task solver, and user interface) are described.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ТЕХНОЛОГИИ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ, ОСНОВАННЫЕ НА ИНТЕГРИРОВАННОЙ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Загорулько Ю.А.

*Институт систем информатики им. А.П. Ершова Сибирского отделения
Российской академии наук, г. Новосибирск, Россия*

zagor@iis.nsk.su

В докладе рассматриваются технологии разработки интеллектуальных систем на основе интегрированной модели представления знаний, которая объединяет взаимодополняющие друг друга методы и средства представления и обработки знаний следующих моделей: онтологической, сетевой, продукционной, недоопределенной и императивной. Благодаря тому, что интеграция указанных методов и средств выполнена на основе объектно-ориентированного подхода, интегрированная модель позволяет представлять одновременно декларативные и процедурные знания, а также знания о вычислениях. Кроме того, такой способ интеграции позволяет добиться эффективной реализации интеллектуальных систем.

Ключевые слова: технология разработки интеллектуальных систем; интегрированная модель представления знаний; онтология; специализированная оболочка интеллектуальной системы.

ВВЕДЕНИЕ

Как показывает практика, при создании интеллектуальной системы (ИС) самым ответственным и трудоемким процессом является построение ее базы знаний (БЗ) [Гаврилова и др., 2001], которая, в конечном счете, определяет полезность и качество всей системы. В связи с этим возникает насущная потребность в моделях представления знаний и построенных на их основе средствах создания баз знаний, обладающих не только высокой выразительностью и гибкостью, но и являющихся достаточно простыми и понятными в использовании не только инженерам знаний – специалистам по представлению знаний, но и экспертам – специалистам в конкретных предметных областях (ПО).

В рамках работ по искусственному интеллекту накоплен широкий спектр методов и средств представления и обработки знаний. Так, семантические сети [Лозовский, 1984; Лозовский, 1990] используются как универсальная память для хранения любой информации, которую можно представить в терминах объектов и отношений между ними. Продукционные системы [Поспелов, 1990] рассматриваются как удобное средство для выражения знаний экспертов в виде множества правил вида «ЕСЛИ-ТО» и организации логического вывода. Фреймы [Минский, 1978],

представляющие объектно-ориентированный подход в искусственном интеллекте, служат как для повышения уровня представления знаний, так и для обеспечения возможности совместного использования декларативных и процедурных знаний. В рамках парадигмы программирования в ограничениях [Ушаков и др., 2000; Apt, 2000] знания можно задавать в виде системы ограничений на значения параметров моделируемых объектов. Аппарат недоопределенных типов данных [Нариньяни, 1986] и разработанный на его основе метод недоопределенных вычислительных моделей [Нариньяни и др., 1998] дает возможность оперировать неточно заданными значениями и частично описанными объектами. Большим потенциалом обладают языки логического программирования [Братко, 2004], позволяющие описывать сложные процессы обработки информации в декларативном стиле. В последнее время при разработке ИС успешно применяются дескриптивные логики [Baader et al., 2003] и построенные на их основе языки и системы представления знаний [OWL, 2004]. Широкую популярность у разработчиков баз знаний получили онтологии [Guarino, 1998; Гаврилова и др., 2006], с помощью которых можно достаточно просто представить согласованную систему понятий моделируемой области знаний.

Как правило, существующие инструментальные системы и оболочки ИС снабжены каким-то одним

из упомянутых выше средств представления знаний, в то время как при создании прикладных ИС часто возникает необходимость их комплексного использования, поскольку ни одно из них, взятое в отдельности, не может обеспечить в полном объеме потребностей разработки реальной прикладной системы. В связи с этим инструментальные средства разработки ИС должны включать различные взаимодополняющие друг друга методы и средства представления и обработки знаний. При этом возможны разные способы интеграции таких средств – начиная с интеграции на уровне подсистем и компонентов ИС и кончая интеграцией на уровне единого формализма (модели).

Хотя первый способ интеграции проще в реализации, более предпочтительным является второй способ интеграции, при котором все необходимые методы и средства объединяются в одном формализме (модели и/или языке представления знаний). Такая интеграция не только создает предпосылки для построения мощных баз знаний (за счет совместного использования различных взаимодополняющих друг друга методов и средств), но и обеспечивает концептуальное единство создаваемых баз знаний (все компоненты БЗ становятся связанными общими понятиями). На основе такого формализма могут быть созданы инструментальные средства, с помощью которых эксперты смогут сами строить базу знаний ИС в полном объеме, не прибегая к помощи посредников – инженеров знаний и программистов. Благодаря этому значительно упрощается разработка базы знаний ИС и повышается ее качество. Кроме того, интеграция всех требуемых средств в рамках одной модели (языка) создает предпосылки для создания на ее основе эффективных прикладных ИС, так как устраняются неизбежные расходы ресурсов на организацию взаимодействия программных систем, реализующих отдельные методы и средства представления знаний.

Таким образом, использование в качестве базиса ИС модели, объединяющей различные взаимодополняющие методы и средства представления знаний, позволяет удовлетворить большинство требований, выдвигаемых к технологиям создания ИС в [Голенков и др., 2011], в частности, обеспечивает семантическое представление знаний, полностью абстрагированное от особенностей технической реализации интеллектуальной системы, минимизирует участие программистов в разработке ИС, позволяет экспертам в ПО принимать непосредственное участие в разработке ИС, тем самым расширяя круг разработчиков ИС и делая технологии их создания массовыми.

В данной работе будут описаны интегрированная модель представления знаний, построенные на ее основе инструментальные средства, а также основанные на этом инструментарию технологии разработки ИС, ориентированные на экспертов.

1. Интегрированная модель представления знаний

1.1. Требования к модели представления знаний

Для того чтобы модель представления знаний была пригодна для разработки широкого класса интеллектуальных систем, она должна обеспечивать представление всех видов знаний, необходимых для разработки и функционирования ИС.

Прежде всего, необходимо представлять знания о предметной области, в рамках которой будет функционировать система. Сюда входят знания об основных сущностях ПО (понятиях и объектах), а также знания о том, каким образом эти сущности связаны между собой. К последним относятся знания об отношениях, непосредственно связывающих понятия, а также более сложные виды знаний, представляющие различного рода зависимости между понятиями ПО, как логические, так и функциональные. В частности, при разработке большого класса ИС возникает необходимость в представлении знаний о ПО в виде правил «ЕСЛИ-ТО». Также довольно часто возникает потребность в представлении знаний в виде ограничений на значения параметров, описывающих некоторый объект или систему объектов.

Для лучшего структурирования ПО и обеспечения более лаконичного ее описания важно иметь возможность выстраивания понятий предметной области в иерархию «общее-частное» и поддержки наследования свойств по этой иерархии.

Базы знаний большинства ИС также содержат конкретные знания из предметной области (предметные знания), которые представляются в виде экземпляров понятий (конкретных объектов) и связей между ними – экземпляров отношений или ограничений, заданных на значениях атрибутов экземпляров понятий.

Другим важным видом знаний, которые нужно представлять в ИС, являются знания о решаемых в рамках моделируемой ПО задачах и способах их решения (методах и алгоритмах). Эти знания характеризуют проблемную область (ПрО) интеллектуальной системы. Такого рода знания могут быть как декларативными, так и процедурными. К первому типу относятся знания, описывающие пространство решаемых ИС задач, включая разбиение задач на подзадачи и связь подзадач с методами их решения, ко второму – знания, представляющие как методы решения задач, так и конкретные алгоритмы.

Кроме знаний, которые постоянно присутствуют в БЗ, необходимо представлять знания, описывающие фрагмент действительности (ситуацию), который задает контекст и входные данные для задач, решаемых ИС. Такие знания, как и предметные знания, обычно представляются в виде экземпляров понятий и отношений и/или

ограничений, заданных на значениях атрибутов экземпляров понятий.

При описании предметных знаний, контекста и входных данных задачи довольно часто возникает необходимость представлять знания об объектах, имеющих неточные значения атрибутов, путем задания оценок таких значений, например, в виде множеств возможных значений и/или ограничений на значения. Для представления таких знаний необходимо обеспечить возможность использования в экземплярах понятий атрибутов, имеющих недоопределенные значения, и задания экземпляров ограничений, связывающих такие атрибуты.

Рассмотренные выше виды знаний, необходимые для полноценного функционирования ИС, не могут быть представлены средствами одной модели. В связи с этим возникает необходимость в модели представления знаний, которая обладает набором методов и средств, достаточным для представления в ИС всех перечисленных выше видов знаний.

1.2. Состав интегрированной модели представления знаний

Исходя из рассмотренных выше требований, предлагается интегрированная модель представления знаний (ИМПЗ-модель), которая объединяет в рамках одного формализма различные взаимодополняющие друг друга средства и методы представления и обработки знаний следующих моделей – онтологической, сетевой, продукционной, императивной и недоопределенной. (Последняя включает аппарат недоопределенных типов данных и разработанный на его основе метод недоопределенных вычислительных моделей, который, как показано в [Нариньяни и др., 1998], фактически является универсальным методом программирования в ограничениях.)

Интегрированная модель представления знаний описывается шестеркой:

$$\langle ONT, SN, FN, PS, IM, GM \rangle \quad (1)$$

где ONT – прикладная онтология, задающая описание основных сущностей (понятий и отношений) предметной и проблемной областей приложения в виде классов объектов и отношений,

SN – объектно-ориентированная семантическая сеть (ОО-сеть), свойства объектов и отношений которой определяются онтологией ONT ,

FN – функциональная сеть (Ф-сеть), включающая экземпляры ограничений, заданные на значениях атрибутов объектов ОО-сети SN ,

PS – система правил-продукций, описанных в терминах классов и отношений онтологии ONT и операций над их экземплярами,

IM – средства императивного программирования,

GM – общий механизм вывода/обработки

знаний и данных, представленных в ИМПЗ-модели, объединяющий механизмы вывода/обработки каждой из интегрируемых моделей.

В качестве каркаса (базиса) интегрированной модели представления знаний выступает прикладная онтология, которая, являясь основным средством описания ПО и ПрО, определяет главные сущности (классы объектов и отношений между ними), которые затем используются остальными средствами модели.

Формально прикладная онтология ONT может быть представлена следующим образом:

$$\langle C, R_{in}, R_C, T, D, A_C, A_R, Cntr, P, R_{rev} \rangle \quad (2)$$

где $C = \{C_1, \dots, C_m\}$ – конечное непустое множество классов, описывающих понятия некоторых ПО и ПрО;

R_{in} , где $R_{in} \subseteq C \times C$ – строгий частичный порядок на множестве классов C , задающий отношение наследования;

$R_C = \{R_1, \dots, R_n\}$, $R_i \subseteq C \times C$ – конечное множество бинарных отношений, заданных на классах C ;

$T = \{T_1, \dots, T_t\}$ – множество типов данных;

$D = \{d_1, \dots, d_p\}$ – множество доменов, таких что $d_i = \{s_1, \dots, s_k\}$, где s_i – значение некоторого простого типа из T ;

$A_C = \{a_1, \dots, a_r\}$, $a_j \subseteq C \times (T \cup D)$ – конечное множество атрибутов, описывающих свойства классов C ;

$A_R = \{a_1, \dots, a_v\}$, $a_k \subseteq R \times (T \cup D)$ – конечное множество атрибутов, описывающих дополнительные свойства отношений R_C ;

$Cntr$ – множество ограничений, заданных на атрибутах классов, т.е. логических выражений вида $Exp_i(e_{i1}, \dots, e_{iw})$, где $e_{ik} \in A_i$ или $e_{ik} \in T_j$, т.е. e_{ik} либо имя атрибута, либо константа;

P – множество математических свойств (симметричность, рефлексивность, транзитивность и т.д.), которые могут быть заданы для отношений R_C ;

R_{rev} , где $R_{rev} \subseteq R \times R$ – множество пар взаимнообратных отношений из R_C .

Прикладная онтология ONT , задавая комплексное описание понятий и отношений моделируемых ПО и ПрО в виде классов объектов и отношений, инкапсулирующих в себе семантические свойства и ограничения на свои атрибуты и аргументы, тем самым определяет структуру семантической и функциональной сети, с

помощью которых представляются предметные знания и знания о задачах, а также задает множество сущностей, в терминах которых описываются продукционные правила.

С помощью отношения R_m на классах онтологии S может быть задана иерархия «общее-частное», по которой организуется наследование свойств (включая атрибуты, отношения и ограничения) вышестоящих классов нижестоящими.

Благодаря включению в ИМПЗ-модель аппарата недоопределенных типов данных, все простые типы данных, входящие в T , имеют недоопределенные расширения, обеспечивающие возможность представления недоопределенных значений (н-значений) в виде интервала или множества допустимых значений и оперирования ими. Это позволяет, в частности, задавать атрибутам объектов ОО-сети н-значения.

Функциональные зависимости между атрибутами сущностей $a_j \in A_C$ могут задаваться в описаниях классов объектов и отношений в виде ограничений $cntr_i \in Cntr$, связывающих значения атрибутов объектов. (В описаниях бинарных отношений ограничения задаются на атрибутах их аргументов.) Такие ограничения представляют собой множество логических выражений (формул), связывающих арифметическими, теоретико-множественными и логическими операциями значения атрибутов одного или нескольких объектов. Следует заметить, что благодаря включению в ИМПЗ-модель метода недоопределенных моделей с помощью ограничений задаются не только условия корректности значений связанных ими атрибутов, но и функциональные интерпретации каждого из этих значений относительно других, что обеспечивает потенциальную возможность вычисления (уточнения) неизвестных (неточных) значений атрибутов через известные (или более точные).

В зависимости от потребностей представления знаний о ПО любому бинарному отношению $R_k \in R$ может быть «приписано» одно или несколько не противоречащих друг другу математических свойств $P_k \in P$ из следующего набора: рефлексивность, симметричность, транзитивность, антирефлексивность, антисимметричность, а также задано обратное отношение.

Конкретные знания о ПО, как и знания о моделируемом в ИС фрагменте действительности или контексте задачи представляются в объектно-ориентированной семантической сети SN , включающей множество экземпляров классов (объектов) и отношений. Для представления этого вида знаний также служат экземпляры ограничений, накладываемых на атрибуты объектов ОО-сети и

храняемые в функциональной сети FN , связанной с ОО-сетью через атрибуты объектов.

Для уточнения значений слотов объектов, ограничения на которые представлены в функциональной сети, используется метод недоопределенных вычислительных моделей.

Описание предметных знаний в виде правил «ЕСЛИ-ТО» средствами системы продукционных правил PS является естественным и удобным для многих областей. В интегрированной модели система продукционных правил работает над семантической сетью, состав понятий и отношений которой определяется онтологией. В связи с этим посылка и заключение любого правила могут описываться в терминах классов и отношений онтологии. Это позволяет не только удобно представлять экспертные знания в виде правил «ЕСЛИ-ТО», но и строить довольно мощные по своим изобразительным и операционным возможностям правила и с их помощью описывать на достаточно высоком уровне довольно сложные процессы вывода и обработки информации.

Знания о пространстве задач могут быть представлены в прикладной онтологии, где (1) описываются типы задач и методов, предназначенных для их решения, и (2) задается разбиение задач на подзадачи и связь задач с методами их решения (знания второго вида также могут быть представлены средствами ОО-сети). При этом сами методы решения могут быть представлены продукционными правилами, объединенными в группы и/или модули. Для этих целей в модель включаются средства структурирования множества правил и управления им, такие как операторы активации и деактивации групп правил, которые могут сочетаться с императивными средствами управления.

Для поддержки эффективной реализации методов и алгоритмов, а также обеспечения создания полноценных прикладных ИС в ИМПЗ-модель включены традиционные для императивной модели средства. Они применяются для разработки алгоритмов в случаях, когда их реализация другими средствами ИМПЗ-модели (например, продукционной модели) была бы слишком неэффективной по времени и/или затратной по памяти, а также для реализации численных вычислений и обеспечения интерфейсных возможностей ИС.

В дополнение к традиционным императивным средствам, таким как процедуры и функции, операторы присваивания, ветвления, цикла и т.п., ИМПЗ-модель включает механизм альтернатив, который является мощным средством поддержки процессов вывода и обработки информации в условиях, когда неточно заданы не только объекты, но и отношения между ними. Данный механизм реализуется оператором альтернативы, который позволяет задавать и перебирать альтернативы (варианты вычислений), обеспечивая при

возникновении противоречия во время обработки очередной из них переход к следующей альтернативе.

Интеграция указанных методов и средств выполняется на основе объектно-ориентированного подхода. При этом:

- понятия и отношения описываемых ПО и ПрО представляются классами объектов и отношений онтологии *ONT*, инкапсулирующими в себе семантические свойства и ограничения на свои атрибуты и аргументы;
- объекты (экземпляры классов) и экземпляры отношений представляются в семантической сети *SN*, структура и свойства которой определены в онтологии;
- ограничения *Cntr* на значения атрибутов объектов представляются в функциональной сети *FN*, связанной через эти атрибуты с семантической сетью *SN*;
- все необходимые процессы вывода и обработки информации (реализующие процесс решения задачи) осуществляются системой продукционных правил *PS*, работающей над семантической сетью, механизмом удовлетворения ограничений, реализуемым в функциональной сети *FN*, а также средствами императивного программирования *IM*.

Взаимосвязи интегрируемых моделей показаны на рисунке 1. Здесь видно, что сетевая и продукционная модели используют сущности, определенные в онтологической модели. В частности, структура и свойства семантической сети определяются в онтологии, а посылки и заключения продукционных правил описываются в терминах классов и отношений онтологии. В свою очередь, средства оперирования, предоставляемые онтологической моделью, используют средства оперирования сетевой модели (для того, чтобы привести ОО-сеть в соответствие с онтологией). Средства оперирования продукционной модели используют и функционал сетевой модели (для сопоставления посылок правил с ОО-сетью и выполнения действий над ней, определенных в заключении правил), и средства оперирования, заданные в императивной модели. Кроме того, сетевая модель использует средства оперирования недоопределенной модели, а последняя – сущности, определенные в сетевой модели.

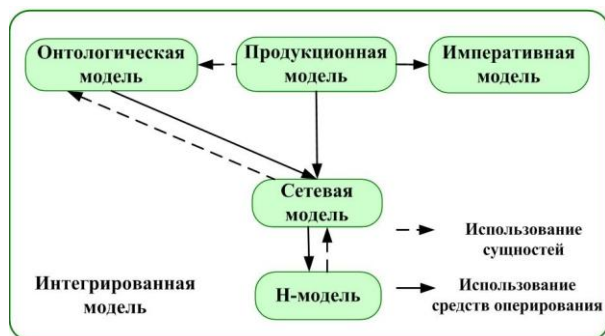


Рисунок 1 – Взаимодействие интегрируемых моделей

1.3. Схема реализации интегрированной модели

ИМПЗ-модель поддерживает представление и решение множества задач, которые в зависимости от назначения прикладной ИС могут быть условно разделены на вспомогательные (подготовительные), выполняемые на этапе настройки системы, и основные – ориентированные на конечного пользователя или получение конечного результата. Однако, такое разделение является условным, так как одна и та же задача может быть для одного приложения вспомогательной, а для другого – основной. В связи с этим в этом разделе рассматривается общий механизм решения задач независимо от их места в «общем разделении труда» в прикладной ИС.

Решение всех задач в ИМПЗ-модели обеспечивается виртуальной машиной (ВМ), реализующей общий механизм вывода/обработки знаний и данных *GM*. Данную виртуальную машину будем называть **главной виртуальной машиной** (ГВМ), потому что она управляет всеми процессами обработки знаний и данных и включает как команды, отвечающие за обработку сущностей онтологической, продукционной, сетевой и недоопределенной моделей, так и команды, реализующие императивные операторы.

ГВМ может быть представлена в виде иерархии виртуальных машин (рисунок 2), каждая из которых обеспечивает обработку определенного вида знаний и данных.

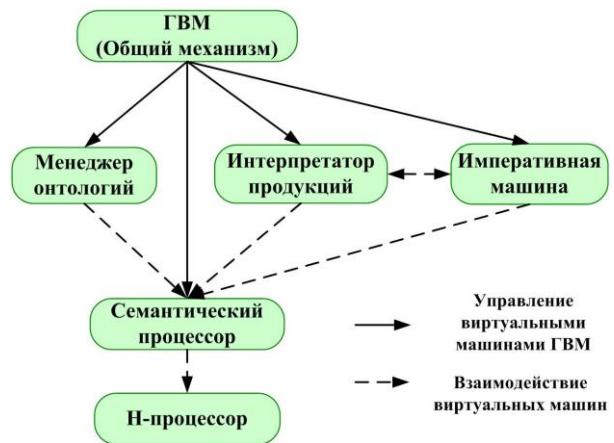


Рисунок 2 – Иерархия виртуальных машин

Выполнение команд создания и редактирования прикладной онтологии обеспечивается виртуальной машиной, которую мы будем называть **менеджером онтологий**. При этом следует иметь в виду, что редактирование онтологии не сводится к простому добавлению, удалению или модификации ее элементов. Семантика операций, выполняемых менеджером онтологий, должна гарантировать логическую целостность новой версии онтологии. Кроме того, если в приложении уже имеется непустая семантическая сеть, то после выполнения операций над онтологией она должна оставаться согласованной с онтологией, т.е. содержать в своем

составе экземпляры только тех классов и отношений и с такими свойствами, какие есть в онтологии.

Команды, обеспечивающие весь комплекс операций над семантической сетью, включая создание, удаление и редактирование объектов и экземпляров отношений, а также поиск фрагментов сети, удовлетворяющих заданному образцу, реализованы в **семантическом процессоре**. При этом нужно учитывать, что семантическая сеть тесно связана с функциональной сетью. В связи с этим при создании пустой ОО-сети создается пустая Ф-сеть, и выполнение операций над объектами и экземплярами отношений ОО-сети при наличии у последних ограничений на атрибуты и/или аргументы влечет за собой выполнение соответствующих операций над Ф-сетью.

Весь спектр операций над функциональной сетью, включая создание, модификацию и интерпретацию ограничений, обеспечивается виртуальным потоковым процессором, который в дальнейшем мы будем называть **Н-процессором**.

Интерпретатор продукций отвечает за работу системы продукционных правил, реализуя процессы вывода/обработки информации, представленные в виде структурированного множества правил.

«Императивная машина» обеспечивает выполнение команд обработки данных и управления, реализуемых в императивной парадигме программирования, а также поддерживает механизм альтернатив.

Взаимодействие виртуальных машин обеспечивается путем включения команд одних машин в реализации команд других. При этом часть команд рассматривается в качестве вызовов процедур или фрагментов, составленных из команд этой же или другой машины. Так, команды «императивной машины», интерпретатора продукций и менеджера онтологий включают вызовы команд семантического процессора, команды семантического процессора обращаются к командам Н-процессора, а команды интерпретатора продукций могут включать команды императивной машины.

Таким образом, процесс решения задачи в ИМПЗ-модели разворачивается в последовательность команд, исполняемую разными виртуальными машинами. Эта последовательность определяется типом приложения и не должна противоречить схеме взаимодействия виртуальных машин, представленной на Рис.2. В частности, в соответствии с этой схемой напрямую к Н-процессору может обращаться только семантический процессор, а к семантическому процессору могут обращаться все остальные ВМ, за исключением Н-процессора.

Построенная таким образом интегрированная модель предоставляет средства для комплексного описания как декларативных, так и вычислительных свойств понятий и отношений некоторой области

знаний, а также средства оперирования их экземплярами в продукционном стиле. Доступ к указанным средствам ИМПЗ-модели осуществляется посредством языка представления и обработки знаний (ЯПОЗ), синтаксические конструкции которого отображаются в команды и данные описанных выше виртуальных машин. На этом языке можно естественным образом описывать и решать задачи, в том числе, требующие сочетания логического вывода и вычислений над неточно заданными значениями.

Язык ЯПОЗ включает два типа средств – средства для описания предметной области (декларативных знаний) и средства для спецификации приложений (процедурных знаний), предназначенных для использования в рамках описанной предметной области. Первая группа средств позволяет описывать ПО в виде классов объектов и отношений онтологии, а также конструировать необходимые для этого новые типы данных и домены. Вторая группа средств предназначена для описания функциональности приложения, главным образом, в виде системы продукционных правил, работающей над семантической сетью, задавать на них структуры и порядок их активации. Кроме того, с целью поддержки разработки полноценных приложений в ЯПОЗ включены традиционные средства императивного программирования, а также средства, обеспечивающие разработку эргономичных пользовательских интерфейсов.

2. Инструментальные средства разработки ИС, основанные на интегрированной модели

Для ускорения и упрощения процесса разработки интеллектуальных систем широко применяется такой класс инструментальных систем, как оболочки ИС [Попов и др., 1996], представляющие собой ИС с пустой или частично заполненной базой знаний. Достоинством оболочек является то, что разработка прикладных ИС на их основе не требует участия программистов, так как наполнение базы знаний входит в компетенцию инженеров знаний и экспертов.

На основе средств, предоставляемых ИМПЗ-моделью, могут быть построены оболочки, предназначенные для разработки различных типов систем, основанных на знаниях, – от оболочек информационных систем, обеспечивающих пользователю содержательный доступ к знаниям и данным определенной области знаний, до оболочек экспертных систем и систем поддержки принятия решений. В зависимости от функциональности и назначения оболочка ИС включает в качестве своего программного ядра ту или иную конфигурацию виртуальных машин, а в качестве средств настройки ИС на ПО – соответствующий программному ядру набор средств редактирования базы знаний.

В рамках данной работы рассматриваются два

типа таких оболочек – оболочка систем, основанных на знаниях (СОЗ), и оболочка порталов знаний.

2.1. Оболочка системы, основанной на знаниях

Оболочка систем, основанных на знаниях, предназначена для построения интеллектуальных систем традиционного типа, т.е. таких СОЗ, за процессы вывода и обработки информации в которых в основном отвечает система продукций. В то же время благодаря тому, что такие СОЗ базируются на ИМПЗ-модели, они способны решать задачи требующие сочетания логического вывода и вычислений над неточно заданными значениями.

Оболочка СОЗ (рисунок.3) включает типовую СОЗ с пустой базой знаний и средства спецификации БЗ.

В типовую СОЗ входят следующие компоненты: интерфейс, база знаний и программное ядро, обеспечивающее реализацию процессов вывода и обработки информации в создаваемых СОЗ.

Интерфейс обеспечивает диалог (взаимодействие) полученной системы с конечным пользователем или внешними подсистемами. СОЗ может включать интерфейсы двух типов – стандартный и специализированный.

Стандартный интерфейс – это абстрактный интерфейс, который обеспечивает основные виды взаимодействия с СОЗ – ввод данных и получение результатов работы. Такое взаимодействие осуществляется на «языке семантической сети», посредством помещения в основную память программного ядра экземпляров объектов и отношений семантической сети или их образцов. При этом дальнейшая интерпретация введенных и полученных в результате работы СОЗ данных (например, на входные и выходные) возлагается на внешние системы, использующие данную СОЗ.

Специализированный интерфейс строится под конкретный тип приложения – для этого в языке ЯПОЗ есть специальные средства и библиотеки. При этом обмен данными с программным ядром также осуществляется через его основную память.

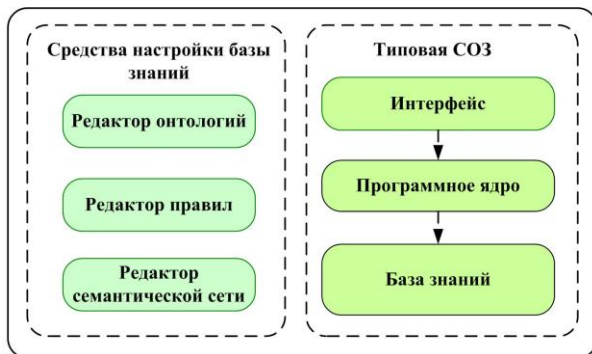


Рисунок 3 – Оболочка СОЗ традиционного типа

Программное ядро СОЗ включает интерпретатор продукционных правил, «императивную машину», семантический процессор и Н-процессор (порядок

взаимодействия между модулями ядра показан на рисунке 5).

Базу знаний прикладной СОЗ (рисунок 4) составляет онтология ПО и ПрО, для работы (моделирования) в которых предназначена СОЗ, семантическая сеть, функциональная сеть, а также система продукций, определяющая основные процессы вывода и обработки информации в СОЗ.

Связанные между собой общими данными семантическая и функциональная сети играют роль основной памяти системы. В семантической сети представлены декларативные знания об объектах и отношениях между ними, в функциональной сети – вычислительные отношения (ограничения), заданные на атрибутах объектов. Вид и свойства этих сетей определяется понятиями и отношениями, введенными инженером знаний на этапе конструирования онтологии.

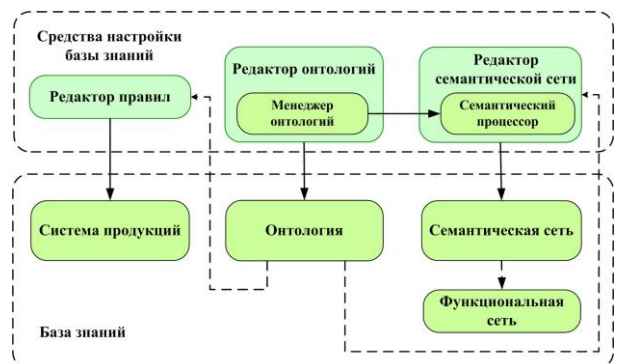


Рисунок 4 – База знаний СОЗ и средства ее настройки

Средства настройки БЗ включают редактор онтологий, предоставляющий удобный интерфейс к менеджеру онтологий, редактор семантической сети, обеспечивающий интерфейс к семантическому процессору, с помощью которого создается начальное наполнение семантической сети, задающее контекст решения задачи или исходное состояние объекта (системы) управления, а также редактор правил, с помощью которого с использованием сущностей, определенных в онтологии, создаются продукционные правила.



Рисунок 5 – Система виртуальных машин оболочки СОЗ традиционного типа

В целом оболочка СОЗ включает все виды рассмотренных выше виртуальных машин (рисунок 5), а также описанные выше редакторы. Кроме того,

в оболочке вводятся определенные ограничения на последовательность применения ВМ (т.е. на ВМ задается частичный порядок).

Возможны следующие последовательности применения виртуальных машин: 1) редактор онтологий, менеджер онтологий; 2) редактор правил; 3) редактор семантических сетей, семантический процессор, Н-процессор; 4) интерпретатор продукций либо семантический процессор, Н-процессор, либо «императивная машина», семантический процессор, Н-процессор. При этом номер на стрелках, выходящих из ГВМ (см. рисунок 5), определяет порядок, в котором эти последовательности могут выполняться.

В соответствии с приведенной схемой типовой сценарий работы с оболочкой следующий:

1. С помощью редактора онтологий создается онтология.

2. С помощью редактора правил с использованием понятий и отношений, определенных в онтологии, описываются и объединяются в группы и модули продукционные правила, которые затем компилируются и загружаются во внутреннюю память системы.

3. С помощью редактора семантической сети создается начальное наполнение семантической сети (фрагмент действительности), задающее контекст решения задачи или исходное состояние объекта (системы) управления.

4. Функционирование системы в рабочем режиме, инициируемое запуском системы продукций.

На последнем этапе в процессе обработки фрагмента действительности формируется результат решения задачи, который представляется в виде экземпляров классов и отношений, также помещаемых во фрагмент действительности.

СОЗ, построенная на основе предложенной оболочки, может быть либо самостоятельной прикладной ИС, либо выступать в качестве интеллектуального ядра другой системы.

2.2. Оболочка портала знаний

Интегрированная модель представления знаний может быть применена не только для создания широкого класса систем, основанных на знаниях, но и для решения другой важной задачи – проблемы эффективного доступа к знаниям и информационным ресурсам, представленным в сети Интернет. Для этих целей строится оболочка портала знаний (ПЗ), которая позволяет создавать специализированные системы – интернет-порталы, обеспечивающие интеграцию и систематизацию знаний и информационных ресурсов по требуемой тематике, а также содержательный доступ к ним.

Информационная модель такого портала базируется на интегрированной модели

представления знаний, что позволяет представлять в нем разнородные знания и данные. Для удобства разработки в информационной модели выделяется два уровня – первый уровень составляет онтология портала знаний, задающая понятия и существующие между ними отношения моделируемой области знаний, включая типологию соответствующих ей информационных ресурсов, а второй – объектно-ориентированная семантическая сеть, содержащая информационные объекты (экземпляры понятий онтологии), связанные отношениями, определенными в онтологии. На основе данной информационной модели организуются удобная навигация по знаниям и информационным ресурсам, интегрированным в портале знаний, и содержательный поиск требующихся информационных объектов.

Оболочка портала знаний (рисунок 6) представляет собой типовой ПЗ и включает пользовательский интерфейс, базу знаний, программное ядро, состоящее из менеджера онтологий, семантического процессора и Н-процессора, а также средства настройки БЗ.

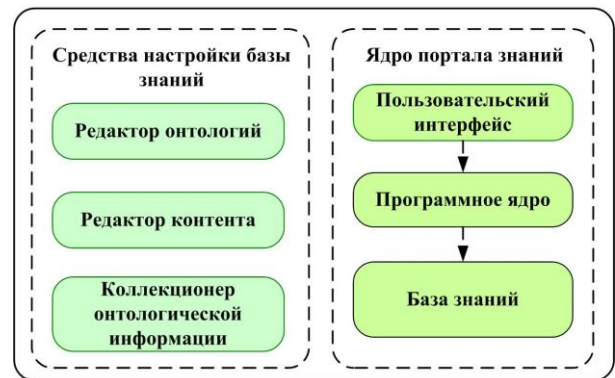


Рисунок 6 – Оболочка портала знаний

Управляемый онтологией пользовательский интерфейс, представляет собой веб-приложение, обеспечивающее навигацию по знаниям и информационным ресурсам, представленным в онтологии и контенте портала знаний, а также содержательный поиск требующейся информации. При выполнении своих команд пользовательский интерфейс обращается к менеджеру онтологий и семантическому процессору, «поставляющим» ему необходимые данные.

Редактор контента предоставляет более удобные и разнообразные средства для управления контентом портала знаний, чем редактор семантической сети, используемый в оболочках СОЗ. При работе он опирается на онтологию и использует команды поиска и редактирования, реализуемые в семантическом процессоре.

Вся работа по созданию конкретного портала знаний состоит в задании необходимых предметных знаний, т.е. онтологии области знаний и соответствующих ей информационных объектов и связей между ними, составляющих контент ПЗ, а также настройке пользовательского веб-интерфейса.

Для этих целей в оболочку ПЗ включены редакторы онтологий и контента, предназначенные для разработки и сопровождения соответственно онтологии и контента ПЗ.

Для повышения степени автоматизации разработки контента ПЗ оболочка предоставляет настраиваемые программные средства для сбора онтологической информации (расширенных метаданных) об информационных ресурсах и публикациях по тематике портала и согласованного пополнения ими контента портала знаний. Эти средства представлены коллекционером онтологической информации (коллекционер ОИ), который при выполнении своих функций использует знания, представленные в онтологии.

Заметим, что указанные редакторы и коллекционер ОИ непосредственно входят в состав портала знаний и, благодаря этому, могут использоваться не только на этапе его создания, но и во время эксплуатации, обеспечивая поддержку его системы знаний в актуальном состоянии.

Оболочка ПЗ включает следующие виртуальные машины (рисунок 7) – редактор онтологий, менеджер онтологий, семантический процессор, Н-процессор, редакторы онтологий и контента, а также коллекционер онтологической информации.

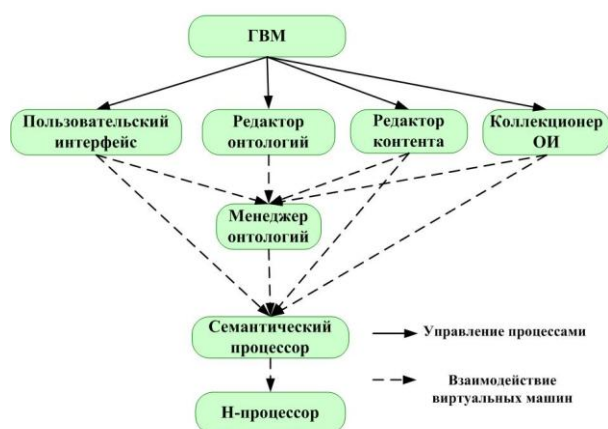


Рисунок 7 – Система виртуальных машин оболочки СОЗ традиционного типа

Возможны следующие последовательности применения ВМ оболочки ПЗ: 1) редактор онтологий, менеджер онтологий, семантический процессор, Н-процессор; 2) редактор контента, семантический процессор, Н-процессор; 3) коллекционер ОИ, семантический процессор, Н-процессор; 4) пользовательский интерфейс, семантический процессор.

Первая последовательность применения виртуальных машин соответствует деятельности по созданию и редактированию онтологии, вторая – деятельности по управлению контентом портала, третья – деятельности по автоматическому пополнению контента портала, четвертая – использованию портала знаний пользователем.

В отличие от оболочки СОЗ в оболочке ПЗ не вводится ограничений на порядок последовательностей применения ВМ, в частности, пользователь может редактировать контент и онтологию в любом порядке.

3. ТЕХНОЛОГИИ РАЗРАБОТКИ ИС, ОРИЕНТИРОВАННЫЕ НА ЭКСПЕРТОВ

Рассмотренные выше программные оболочки являются эффективными средствами построения интеллектуальных систем указанного типа. Однако есть еще одна нерешенная проблема, которая состоит в том, что оболочка ИС, даже основанная на описанной выше интегрированной модели представления знаний, является все еще трудной для непосредственного использования экспертом, так как содержит только универсальные средства представления знаний, которые далеки от области знаний эксперта и поэтому требуют участия в разработке ИС инженеров знаний.

В настоящее время наблюдается тенденция все большего вовлечения экспертов в разработку ИС с целью ее ускорения. Более того, существуют целые классы приложений, использующие большие объемы предметных знаний, при создании которых желательно возложить разработку и сопровождение их базы знаний целиком на эксперта. К таким приложениям можно отнести, например, медицинские экспертные системы, порталы научных знаний, электронные тезаурусы и т.п.

Представляется очевидным, что чем меньше будет разрыв между знаниями и представлениями эксперта и средствами представления знаний инструментальной системы, тем легче ими будет пользоваться эксперту. Поэтому специализация оболочки ИС на определенный класс областей знаний и задач, т.е. включение в ее состав знаниевых компонентов, содержащих базовый набор понятий, относящихся к заданному классу областей знаний, а также программных компонентов, поддерживающих требуемую в ИС функциональность, сделало бы ее более понятной экспертам и пригодной для использования ими без привлечения инженеров знаний [Артемьева, 2008].

На описанной выше идее специализированных оболочек строится концепция технологии разработки СОЗ и порталов знаний, ориентированная на экспертов. В соответствии с этой концепцией общая архитектура инструментальной поддержки разработки ИС выглядит, как показано на рисунке.8.

Базис инструментальной поддержки составляет базирующаяся на ИМПЗ-модели и языке ЯПОЗ интегрированная система представления и обработки знаний, которая представляет собой набор языковых и программных средств, реализующих интегрированную модель.

Следующий уровень включает оболочки, предназначенные для построения прикладных

интеллектуальных систем (СОЗ) и порталов знаний. Оболочки этого уровня предоставляют универсальные средства для построения систем данного типа и предназначены для инженеров знаний.



Рисунок 8 – Общая архитектура инструментальной поддержки разработки ИС

Самый верхний уровень образуют специализированные оболочки, ориентированные на определенные области знаний и классы задач, и, благодаря этому, пригодные для использования экспертами. Причем специализированные оболочки строятся инженерами знаний с помощью средств разработки баз знаний, предоставляемых универсальными оболочками. В соответствии с этим специализированная оболочка СОЗ включает базовые онтологии, на основе которых строится онтология области знаний СОЗ и онтология решаемых ею задач, а также набор специализированных продукционных модулей и библиотек процедур и функций (на языке ЯПОЗ), реализующих решение базовых задач.

Создание конкретной СОЗ на основе такой оболочки заключается в построении онтологии области знаний СОЗ путем расширения и развития базовых онтологий, построении системы продукций путем создания дополнительных продукционных модулей, реализующих задачи, специфичные для разрабатываемой СОЗ, а также наполнении декларативной части базы знаний (семантической и функциональной сети) необходимым набором экземпляров объектов и отношений, представляющих универсальные факты или начальное состояние объекта (системы) управления.

Примерами специализированных оболочек СОЗ, созданных на базе универсальной оболочки СОЗ,

являются оболочка СОЗ медицинской диагностики и оболочка СОЗ диагностики и мониторинга технологической инфраструктуры предприятия [Zagorulko et al., 2010]. Первая из этих оболочек включает общую онтологию медицинской диагностики и лечения, которая содержит понятия и отношения, общие для всех диагностических медицинских систем. Вторая оболочка включает верхний уровень онтологии технологической инфраструктуры предприятия, а также библиотеку процедур и функций для анализа временных рядов, которые могут использоваться в правилах, осуществляющих мониторинг и диагностику технологической инфраструктуры предприятия.

Аналогичным образом строятся и используются специализированные оболочки порталов знаний. Каждая такая оболочка представляет собой типовой портал знаний с набором базовых онтологий, описывающих основные сущности некоторого класса областей знаний, и пустым контентом. Такие специализированные оболочки строятся инженерами знаний путем дополнения универсальной оболочки соответствующим набором базовых онтологий.

Разработка конкретного портала знаний с помощью специализированной оболочки заключается, главным образом, в настройке его на требуемую область знаний, которая состоит в построении онтологии области знаний портала на основе имеющихся в оболочке базовых онтологий и заполнении контента портала под управлением построенной онтологии. Эту работу уже могут выполнять сами эксперты без привлечения программистов и инженеров знаний.

Средствами оболочки портала знаний были построены две специализированные оболочки – оболочка для разработки порталов научных знаний (ПНЗ-оболочка) [Загорулько, 2009] и оболочка для создания многоязычных тезаурусов (Т-оболочка) [Загорулько и др., 2012].

Эти оболочки отличаются, главным образом, набором базовых онтологий. Так, в ПНЗ-оболочку включены две базовые онтологии – онтология научного знания, содержащая базовые понятия, задающие структуры для описания области знаний портала, и онтология научной деятельности, включающая базовые понятия, служащие для описания различных аспектов научно-исследовательской деятельности. В то время как Т-оболочка включает онтологию представления тезауруса, основанную на стандартах построения многоязычных информационно-поисковых тезаурусов.

Следует заметить, что специализированные оболочки могут быть отправной точкой для построения еще более специализированных оболочек. Например, на основе ПНЗ-оболочки можно построить оболочку для разработки порталов знаний, предназначенных для поддержки инновационной деятельности, расширив набор

базовых онтологий ПНЗ-оболочки онтологией производственной и инновационной деятельности.

ЗАКЛЮЧЕНИЕ

В качестве базиса для представления знаний в ИС была предложена модель, интегрирующая взаимодополняющие друг друга методы и средства представления и обработки знаний следующих моделей: онтологической, сетевой, продукционной, недоопределенной и императивной.

Интеграция указанных методов и средств была выполнена на основе объектно-ориентированного подхода. В соответствии с этим подходом понятия и отношения предметной и проблемной областей представляются классами объектов и отношений, инкапсулирующими в себе семантические свойства и ограничения на свои атрибуты и аргументы, объекты этих классов и экземпляры отношений представляются в семантической сети, структура и свойства которой определены в онтологии, ограничения на значения атрибутов объектов – в функциональной сети, связанной через эти атрибуты с семантической сетью, а все необходимые процессы обработки информации (реализующие процесс решения задачи) осуществляются работающей над семантической сетью системой продукционных правил, механизмом удовлетворения ограничений, реализуемым в функциональной сети, а также средствами императивного программирования.

Благодаря такому способу интеграции ИМПЗ-модель позволяет представлять одновременно декларативные и процедурные знания, а также знания о вычислениях (т.е. знания, позволяющие уточнять или находить точные значения параметров описываемого объекта или ситуации). Кроме того, такой способ интеграции позволяет добиться эффективной реализации ИС и сделать доступными все интегрируемые средства в рамках одного языка представления и обработки знаний.

Набор языковых и программных средств, реализующих ИМПЗ-модель, образует интегрированную систему представления и обработки знаний, которая послужила базисом для инструментальных средств разработки интеллектуальных систем. В частности на основе этой системы были построены оболочка СОЗ традиционного типа [Загорюлько и др., 1995] и оболочка портала знаний [Загорюлько и др., 2009]. Обе оболочки могут использоваться инженерами знаний для разработки СОЗ и порталов знаний без привлечения программистов.

Несмотря на успешное применение указанных оболочек для разработки прикладных ИС, следует заметить, что они являются довольно трудными для освоения и непосредственного использования экспертами, так как содержат только универсальные средства представления знаний, далекие от областей знаний экспертов. В связи с этим технологию создания ИС, ориентированную на экспертов, было

решено базировать на специализированных оболочках, настроенных на определенные области знаний и классы задач. При этом сами специализированные оболочки могут создаваться инженерами знаний путем частичного заполнения баз знаний соответствующих универсальных оболочек с использованием имеющихся в ней средств описания предметных знаний.

Работа выполнена при финансовой поддержке Президиума РАН (интеграционный проект СО РАН № 15/10 «Математические и методологические аспекты интеллектуальных информационных систем»).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Артемьева, 2008] Артемьева, И.Л. Специализированные оболочки интеллектуальных систем для сложно-структурированных предметных областей / И.Л.Артемьева // Труды 11-й национальной конференции по искусственному интеллекту с международным участием КИИ-2008 (г.Дубна, Россия). – М: ЛЕНАНД, 2008. –Т.1. - С. 95-103.
- [Братко, 2004] Братко, И. Алгоритмы искусственного интеллекта на языке Prolog / И. Братко. –М.: Издательский дом «Вильямс», 2004. – 637 С.
- [Гаврилова и др., 2001] Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2001.
- [Гаврилова и др., 2006] Гаврилова, Т.А. Модели и методы формирования онтологий / Т.А. Гаврилова, Д.В. Кудрявцев, В.А. Горовой // Научно-технические ведомости СПбГПУ, № 4, 2006. – С.21-28.
- [Голенков и др., 2011] Голенков, В.В. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н.А. Гулякина // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2011): мат. Междунар. науч.-техн. конф. – Минск: БГУИР, 2011. – С. 21-58.
- [Загорюлько и др., 1995] Загорюлько, Ю.А. Интегрированная технологическая среда для создания систем обработки знаний / Ю.А. Загорюлько, И.Г. Попов, В.В. Щипунов // Известия РАН. Теория и системы управления. –1995. – № 5. – С. 210–213
- [Загорюлько, 2009] Загорюлько, Ю.А. Технология разработки порталов научных знаний / Ю.А. Загорюлько // Программные продукты и системы. – 2009. – № 4. –С.25-29.
- [Загорюлько и др., 2009] Загорюлько, Ю.А. Информационная модель портала научных знаний / Ю.А. Загорюлько, О.И. Боровикова // Информационные технологии. – 2009. – № 12. – С.2–7.
- [Загорюлько и др., 2012] Загорюлько, Ю.А. Программная оболочка для построения многоязычных тезаурусов предметных областей, ориентированная на экспертов / Ю.А. Загорюлько, О.И. Боровикова // Труды 13-й национальной конференции по искусственному интеллекту с международным участием КИИ-2012. – Белгород: Изд-во БГТУ, 2012. –Т.4. -С. 76-83.
- [Минский, 1978] Минский, М. Структура для представления знания / М. Минский // Психология машинного зрения. – М.: Мир. – 1978. – С. 249-338.
- [Нариньяни, 1986] Нариньяни, А.С. Недоопределенность в системах представления и обработки знаний / А.С. Нариньяни // Изв. АН СССР. Техн. кибернетика. – 1986. – №5. – С. 3-28.
- [Нариньяни и др., 1998] Нариньяни, А.С. Программирование в ограничениях и недоопределённые модели / А.С. Нариньяни, В.В. Телерман, Д.М. Ушаков, И.Е. Швецов // Информационные технологии. 1998. – №7. – С. 13-22.
- [Лозовский, 1984] Лозовский, В.С. Семантические сети / В. С. Лозовский // Представление знаний в человеко-машинных и робототехнических системах. – М.: ВИНТИ, 1984. –Т.А. – С. 84-121.
- [Лозовский, 1990] Лозовский, В.С. Сетевые модели / В. С. Лозовский // Искусственный интеллект. В 3-х кн. Кн.2. Модели и

методы: Справочник / Под. Ред. Д.А. Поспелова. – М.: Радио и связь, 1990. – С. 28-49.

[Попов и др., 1996] Попов, Э.В. Статические и динамические экспертные системы: Учеб. пособие. / Э.В. Попов, И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. –М.: Финансы и статистика, 1996. – 320 С.

[Поспелов, 1990] Поспелов, Д.А. Продукционные модели / Д. А. Поспелов // Искусственный интеллект. В 3-х кн. Кн.2. Модели и методы: Справочник / Под. Ред. Д.А. Поспелова. – М.: Радио и связь, 1990. – С. 49-56.

[Ушаков и др., 2000] Ушаков, Д.М. Системы программирования в ограничениях (обзор) / Д.М. Ушаков, В.В. Телерман // Системная информатика: Сб. науч. тр. Под. Ред. И.В. Поттосина. Вып.7: Проблемы теории и методологии создания параллельных и распределенных систем. – Новосибирск: Наука, 2000. –С. 275–310.

[Apt, 2000] Apt K. R. Principles of Constraint Programming / K.R. Apt. –New York: Cambridge University Press, 2003. 407 P.

[Baader et al., 2003] Baader, F. The Description Logic Handbook: Theory, Implementation, Applications / F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider – Cambridge, 2003. 574 P.

[Guarino. 1998] Guarino, N. Formal Ontology in Information Systems / N. Guarino // Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, 6–8 June 1998. – Amsterdam: IOS Press, 1998. – P. 3–15.

[OWL, 2004] OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004: [Electronic resource]. – 2004. – The University of Sheffield, 1995–2011. – Mode of access: <http://www.w3.org/TR/owl-guide/> – Date of access: 12.11.2012.

[Zagorulko et al., 2010] Zagorulko, Yury. An Approach to Development of the Decision Support System for Enterprise with Complex Technological Infrastructure / Yury Zagorulko, Galina Zagorulko. // Bulletin of NCC – Issue 31.– 2010.–P. 195–207.

TECHNOLOGIES OF DEVELOPMENT OF INTELLIGENT SYSTEMS BASED ON THE INTEGRATED KNOWLEDGE REPRESENTATION MODEL

Zagorulko Yu.A.

*A.P. Ershov Institute of Informatics Systems
Siberian Branch of the Russian Academy of
Sciences, Novosibirsk, Russia*

zagor@iis.nsk.su

The paper discusses technologies of development of intelligent systems (IS) based on the integrated knowledge representation model. The integrated model combines methods and means of knowledge representation and processing of ontological, network, production, subdefinite and imperative models. The knowledge base development tools built on the basis of this model are possessed of high expressiveness and flexibility and at the same time easy-to-use for both knowledge engineers and experts. These tools are foundation of intelligent system development technology oriented to experts.

INTRODUCTION

The most labor-consuming and crucial stage of intelligent system development is the building of its knowledge base, therefore there exists a vital need in knowledge representation model and knowledge base development tools built on their bases which are possessed of high expressiveness and flexibility and at the same time easy-to-use for both knowledge engineers and experts.

We propose the integrated knowledge representation model as such model and tools based on this model as such tools. Use of the model combining various mutually complementary methods and means of knowledge representation and processing as basis of intelligent system allows to meet the main requirements which are setting to technologies of development of IS.

MAIN PART

We have conducted comprehensive investigation of all kinds of knowledge which must be represented in intelligent systems. On the basis of results of this investigation a new knowledge representation model was developed. This model which was named the integrated knowledge representation model combines in one formalism methods and means of knowledge representation and processing of ontological, network, production, subdefinite and imperative models. Applied ontology is used as basis of the integrated model. This ontology, being a main means for description of subject and problem domains, defines principal entities (classes of objects and relations between them) which are later used other means of the model.

The integrated model built in such way provides facilities for complex description both declarative and procedural properties of concepts and relations of certain knowledge area as well as facilities for operation of their instances.

Several program shells intended for development of various types of system based on knowledge was designed on the basis of the integrated model. These are shell of knowledge based system and shell of knowledge portal. In turn on the basis of these universal shells the specialized shells which can be used by experts without resorting to the help of programmers and knowledge engineers can developed.

In accordance the suggested technology of IS development the specialized shells are built by knowledge engineers which perform a partial filling of knowledge bases of corresponding universal shells using for this aim knowledge base development tools of these shells.

CONCLUSION

Though shell of knowledge based system and shell of knowledge portal were successfully used for development of set of applied intelligent system, they are hard-to-use for experts. Therefore we have decided to base technology of IS development on specialized shells. On the base of the shell of knowledge based system was developed two specialized shells. First of them is a shell of medical diagnostics and cure, second is a shell of system of diagnostics and operational monitoring of the technological infrastructure of an industrial enterprise. Using the knowledge portal shell two specialized shells was built: shell of scientific knowledge portal and shell for development of multilingual electronic thesaurus.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89

ТЕХНОЛОГИЯ ФОРМАЛЬНО-ЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ И ПРОТОТИПИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Ануреев И.С., Атучин М.М.

*Институт систем информатики имени А.П. Ершова,
г. Новосибирск, Россия*

anureev@iis.nsk.su

atuchin.m@gmail.com

В работе представлена технология, позволяющая выбирать оптимальную концептуальную конфигурацию интеллектуальной системы на этапе ее проектирования и создавать ее прототип на базе формально-логических методов. В частности, использование формально-логического аппарата обеспечивает дедуктивную верификацию свойств безопасности проектируемой интеллектуальной системы. Технология иллюстрируется на примере спецификации концептуальной модели системы поддержки принятия решений.

Ключевые слова: операционно-онтологическая семантика, логика безопасности, системы переходов, интеллектуальные системы.

ВВЕДЕНИЕ

Применение формально-логических методов для обеспечения безопасности и эффективности программного обеспечения становится стандартом при проектировании и прототипировании промышленных программных систем (прежде всего критического программного обеспечения, примерами которого являются бортовые спутниковые системы, встроенные автомобильные системы, банковские системы, системы проведения выборов и т. п.). Наметившаяся в последнее время тенденция использования интеллектуальных компонент в таких системах делает актуальной задачу разработки соответствующих формальных методов для интеллектуальных систем (далее ИС) с учетом их специфики.

Следует также отметить, что в настоящее время происходит переход от разработки конкретных ИС к созданию технологий их разработки [Хорошевский, 2008; Хорошевский, 2009; Хорошевский, 2012а; Хорошевский, 2012б; Голенков, 2011; Грибова и др., 2012; Загоруйко, 2009]. Поэтому важной задачей является создание технологии на базе формально-логических методов (далее ФЛП-технологии) применительно к задачам проектирования и прототипирования ИС.

В статье предлагается подход к решению этих двух задач.

1. Базис ФЛП-технологии

ФЛП-технология базируется на специализированных помеченных системах переходов (далее П-системах) [Anureev, 2012], ориентированных на ускоренную разработку средств спецификации и верификации программных систем (далее ПС).

П-система строится над двумя базовыми множествами. Первое множество, называемое множеством элементов, определяет всевозможные синтаксические и семантические объекты, которые могут использоваться в специфицируемой ПС. Второе множество, называемое множеством атомов, определяет наименьшие синтаксические объекты языка П-систем, используемого для описания функциональностей ПС, правил перехода, имен объектов ПС и т. д. Элементы и атомы вместе называются атоменами, а язык П-систем — языком Atoment.

Одним из требований к языку Atoment является его близость по синтаксису к естественному языку. Основными синтаксическими объектами языка Atoment являются последовательности, списки и выражения. Атомы соответствуют словам естественного языка. Последовательности объектов (атомов, элементов и т. д.) — это упорядоченные наборы объектов, разделенных пробелами. Последовательности атомов соответствуют фразам (последовательностям слов) естественного языка. Списки — это последовательности, заключенные в круглые скобки (и). Списки атомов соответствуют

предложениям естественного языка, а последовательности списков атомов — текстам на естественном языке. Выражения — это либо атомы, либо списки выражений. Выражения можно рассматривать как вид иерархического текста со сложной вложенной структурой (определяемой скобками).

Особенностью П-систем является использование алгебраических систем в качестве состояний. Выбор подходящей сигнатуры алгебраической системы при проектировании и прототипировании ПС позволяет в какой-то степени решать проблему моделирования концептуального окружения ПС (системы понятий и категорий, которая позволяет сделать спецификации ПС обозримыми и емкими в смысловом аспекте за счет переноса части смысла на уровень терминов и обозначений.).

В отличие от стандартных алгебраических систем, используемых, например, в машинах абстрактных состояний [Gurevich, 2004], в которых символы сигнатуры — атомы, символы сигнатуры в П-системах — списки атомов. Использование списков атомов в качестве символов сигнатуры (далее символов) позволяет приблизить описание вызовов функций, обозначаемых этими символами, к описанию на естественном языке. Другой важной особенностью таких символов является использование в них специальных атомов `_` и `__`, которые делают эти символы шаблонами для соответствующих функций, обеспечивая некоторую информацию о вызовах этих функций. Эти атомы, называемые спецификаторами аргументов, означают места для аргументов функций. Атом `_`, входящий в символ сигнатуры, дополнительно указывает на то, что аргумент при вызове соответствующей этому символу функции нужно вычислить, а атом `__` — что вычислять аргумент не нужно. Например, символ `(_ is concept)` является шаблоном для логической функции, которая возвращает значение `true`, если ее (единственный) аргумент является понятием. Число входящих спецификаторов аргументов в символ называется его местностью (и местностью функции, обозначаемой этим символом), а `i`-е вхождение спецификатора аргумента в этот символ — `i`-м аргументом соответствующей функции.

Символы сигнатуры делятся на модифицируемые символы (например, символ `(_ is concept)`), семантика (интерпретация) которых может меняться при переходе П-системы из состояния в состояние, и предопределенные символы (например, конъюнкция `(_ and _)` и универсальный квантор `(forall __ __)`) с постоянной семантикой.

Конфигурации, в которых может находиться П-система, определяются обычным образом, как пары, состоящие из метки и состояния. Метками в П-системах являются программы, представляющие собой последовательности выражений на языке `Atoment`.

В [Anureev, 2012] выделены три вида специализированных систем переходов.

Ориентированные на операционную семантику системы переходов (далее ОС-системы) предназначены для разработки формальных спецификаций компьютерных языков и операционной семантики ПС.

Ориентированные на логику безопасности системы переходов (далее ЛБ-системы) предназначены для разработки дедуктивных методов обеспечения безопасности ПС.

Онтологические системы переходов (далее ОНТ-системы) предназначены для разработки операционно-онтологической семантики языков программирования [Ануреев, 2009] и ПС [Ануреев, 2008]. Эти системы являются специальным видом ориентированных на операционную семантику систем переходов, в которых в множестве модифицируемых символов выделяются два подмножества: множество онтологических символов и множество символов экземпляризации. Онтологические символы специфицируют элементы (понятия, атрибуты, отношения и т. п.) онтологии ПС. Символы экземпляризации специфицируют связи элементов онтологии с экземплярами.

Для каждого вида П-систем определен интерпретатор правил переходов, что делает спецификации ПС, описываемые с помощью П-систем, выполнимыми.

Специфика ИС состоит в том, что ФЛП-технология использует для проектирования и прототипирования ИС последние два вида П-систем и практически не использует ОС-системы, поскольку ИС, как правило, имеют сложную онтологическую структуру.

2. Этапы проектирования и прототипирования интеллектуальной системы в рамках ФЛП-технологии

Проектирование и прототипирование ИС на основе ФЛП-технологии состоит из следующих этапов.

На первых трех этапах строится прототип ИС.

Этап 1. Описывается множество состояний П-системы, которая специфицирует ИС. Поскольку для спецификации ИС используются ОНТ-системы, это описание должно включать множество онтологических символов и множество символов экземпляризации.

Этап 2. Определяется предметно-ориентированный язык (далее ПО-язык) как подмножество языка `Atoment`, описывающий желаемую функциональность ИС. Каждой функциональности соответствует конструкция ПО-языка. Также на этом шаге определяется неформальная семантика конструкций ПО-языка.

Этап 3. На основе ОНТ-систем, разрабатывается

операционно-онтологическая семантика ПО-языка.

Поскольку операционно-онтологическая семантика ИС, построенная на базе ОНТ-систем, является выполнимой, она задает прототип ИС.

На остальных этапах выбирается оптимальная концептуальная конфигурация ИС и доказывается ее безопасность.

Этап 4. На этом этапе, называемом этапом тестирования прототипа ИС, моделируются различные последовательности функционирования ИС. Целью такого тестирования является нахождение ошибок проектирования ИС и выбор оптимальной концептуальной конфигурации ИС.

Этап 5. Выполняется экспертная оценка «правильности» и «оптимальности» проектирования ИС посредством анализа текста формальных спецификаций ИС (соответствующей ОНТ-системы). Возможными действиями эксперта на этом этапе являются слияние и разделение функциональностей ИС, упрощение или обобщение элементов состояния ИС и т. д.

Этап 6. Разрабатывается логика безопасности для ПО-языка, основанная на ЛБ-системах. Особенность этого класса П-систем состоит в том, что, как правило, большая часть правил логики безопасности совпадает с соответствующими правилами операционно-онтологической семантики ПО-языка.

Этап 7. Выполняется расстановка свойств безопасности в программе на ПО-языке, которая специфицирует ИС, и с помощью логики безопасности выполняется проверка безопасности ИС относительно расставленных свойств. Фактически осуществляется дедуктивная верификация ИС с генерацией и последующим доказательством условий корректности. Если все сгенерированные условия корректности истинны, то ИС безопасна.

Заметим, что на каждом этапе может происходить возвращение к предыдущим этапам. Например, определение правил операционной семантики может потребовать уточнение набора функциональностей ИС или ее состояния.

Кроме того, ФЛП-технология допускает многовариантную разработку прототипа ИС, при которой на каждом этапе рассматривается набор вариантов. На шаге 1 вариантами будут различные описания состояния ИС, на шаге 2 — различные наборы функциональностей ИС и комбинирующих их конструкторов, и т. д.

3. ПРИМЕР ПРОЕКТИРОВАНИЯ СППР С ПОМОЩЬЮ ФЛП-ТЕХНОЛОГИИ

В этом разделе мы опишем идею спецификации концептуальной модели СППР СОМТИ [Загоруйко и др., 2010] (далее КМ) на языке Atoment, проиллюстрировав первые три этапа ФЛП-

технологии. Для экономии места считается, что читатель знаком с работой [Загоруйко и др., 2010].

На первом этапе определяется состояние ОНТ-системы, специфицирующей КМ, как интерпретация онтологических символов и символов экземпляризации.

Множество онтологических символов, используемых в состояниях ОНТ-системы, включает 4 символа. Символ (`_ is concept`) определяет множество понятий. Символ (`_ is attribute of _`) определяет множество атрибутов понятий, а его типизированный вариант (`_ is attribute of _ of _`) определяет множество типизированных атрибутов понятий. Типом является третий аргумент этого символа. Символ (`_ inherits _`) определяет отношение наследования на понятиях.

Интерпретация онтологических символов определяется следующим образом. Каждая сущность, используемая в КМ, описывается соответствующим понятием, которое добавляется в состояние предопределенным (из стандартной библиотеки языка Atoment) выражением (`A ::= B`), называемым модификацией символа или присваиванием. Это выражение меняет интерпретацию символа сигнатуры, которому соответствует вызов `A` на значение выражения `B` для аргументов, входящих в `A`. Например, добавление понятия `адаптер`, соответствующего сущности `адаптер`, осуществляющей взаимодействие СППР СОМТИ с окружением, определяется присваиванием (`((адаптер is concept) ::= true)`), изменяющим интерпретацию символа (`_ is concept`) в точке `адаптер` на `true`.

Атрибуты сущности, используемой в КМ, определяются как атрибуты соответствующего понятия. Так, например, понятие `задача` имеет атрибуты `имя`, `параметры`, `подзадачи`, `порождает` и `реализация`. Присваивание (`((имя is attribute of задача of string) ::= true)`) добавляет атрибут `имя` типа `string` понятию `задача`. Аналогичным образом определяются остальные атрибуты понятия `задача`. Заметим, что выбрано такое определение состояния, при котором бинарные отношения, используемые в КМ, также моделируются атрибутами. Например, бинарное отношение `подзадача` на задачах моделируется атрибутом `подзадачи` понятия `задача`. Можно было бы напрямую моделировать отношения, используемые в КМ, если расширить множество онтологических символов символом (`_ is relation on _ and _`). В этом случае, отношение `подзадача` определялось бы присваиванием (`((подзадача is relation on задача and задача) ::= true)`).

Таким образом, интерпретация онтологических символов, специфицирующая онтологию КМ,

определяется программой, которая является последовательностью присваиваний онтологическим символам. Как правило, такая онтология не меняется при функционировании ОНТ-системы.

Множество символов экземпляризации, используемых в состояниях ОНТ-системы, включает 3 символа. Символ (`_ is _`) определяет множество экземпляров (первый аргумент) понятия (второй аргумент). Символ (`_ of _`) определяет значение атрибута (первый аргумент) экземпляра понятия (второй аргумент). Его модифицированный вариант (`_ of _ of _`) определяет значение атрибута (первый аргумент) экземпляра (второй аргумент) понятия (третий аргумент). Он используется для разрешения конфликта в случае, когда экземпляр принадлежит нескольким понятиям, имеющим один и тот же атрибут.

На втором этапе определяется ПО-язык, специфицирующий функциональность КМ как некоторое конечное множество параметризованных выражений (далее ПО-выражений) языка *Atoment*. На третьем этапе определяются правила перехода ОНТ-системы, задающие семантику ПО-выражений.

Функциональность КМ определяется как объединение функциональности понятий КМ. Функциональность понятия есть множество ПО-выражений, которые содержат в качестве аргумента хотя бы один атомент, являющийся экземпляром этого понятия. Например, функциональность понятия *адаптер* включает три параметризованных выражения, специфицирующих действия *получить задание*, *загрузить данные* и *выгрузить данные*. Пусть *I* — экземпляр понятия *адаптер*. Выражение (*получить задание* `:: I`) запрашивает задание у операционной среды. Атоменты, входящие в выражение и следующие за спецификатором контекста `::`, называются контекстами этого выражения. Семантика ПО-выражения меняется в зависимости от используемых в ней контекстов. Выражение (*получить задание* `:: I`) задается правилом перехода (`if (получить задание :: I) var I then (assume (I is адаптер)) (modify ((: value) is задание))`).

Правило перехода представляет собой последовательность выражений языка *Atoment*. Часть `if` правила содержит образец, с которым сопоставляется ПО-выражение. Часть `var` содержит последовательность переменных образца, которые получают значения при сопоставлении ПО-выражения с образцом. Часть `then`, называемая телом правила, представляет собой последовательность выражений (программу), которая выполняется (после замены переменных образца соответствующими значениями) в случае, если ПО-выражение соответствует образцу.

Тело правила для ПО-выражения (*получить задание* `:: I`) представляет собой последовательность из двух выражений. Выражение (`assume (I is адаптер)`) выполняется тогда и только тогда, когда *I* — экземпляр понятия *адаптер*. Выполнение этого выражения не меняет состояния ОНТ-системы. В общем случае, предопределенное выражение (`assume A`), называемое условием продолжения, выполняется тогда и только тогда, когда истинна формула *A*. В противном случае, данная ветвь выполнения (последовательность конфигураций в терминах систем переходов) считается фиктивной и происходит откат (*backtracking*) в предыдущую конфигурацию и выбор другого правила для сопоставления. Выражение (`modify ((: value) is задание)`) изменяет интерпретацию нульместного символа (*value*) таким образом, что новое его значение является заданием (экземпляром понятия *задание*). В общем случае, предопределенное выражение (`modify A`), называемое условием модификации, преобразует состояние ОНТ-системы таким образом, что в новом состоянии истинна формула *A*. При этом могут изменяться только те символы, которые входят в *A* и которым предшествует атом `:`.

Таким образом, это выражение описывает некоторое декларативное действие, т. е., действие, которое определяет результат (с помощью логической формулы) и не определяет, как этот результат был получен. Использование декларативных действий позволяет выполнять проектирование на любом необходимом уровне абстракции, подбирая оптимальную «смесь» операционного и декларативного представлений. Выразительность декларативного представления определяется логикой (пропозициональная логика, логика предикатов, логика высшего порядка, табличная логика вида *условие–результат*, используемая при описании требований к проектируемой ИС и т. д.), которой принадлежит формула *A* в инструкциях `assume` и `modify`.

Синтаксис и семантика других ПО-выражений определяется аналогичным образом.

В заключение, определим функциональность основного модуля СППР СОМТИ — супервизора. Понятие супервизор имеет атрибуты *обработчик-заданий*, *интерпретатор-конфигураций* и *адаптер*, которые связывают соответствующие сущности КМ с супервизором. Например, присваивание (`(обработчик-заданий is attribute of супервизор of обработчик-заданий) ::= true`) связывает *обработчик-заданий* с супервизором.

Функциональность понятия супервизор включает ПО-выражение (*старт* `:: S`) запуска супервизора *S*, задаваемое следующим правилом:

```
if (старт :: S) var S
```

```

then (assume (S is супервизор))
  ((получить-задание ::
    (адаптер of S)))
  (if ((value) is задание)
    then
      (обработать (value) ::
        (обработчик-заданий of S)
        (while (not ((value) is пустая
          конфигурация решения))
          do
            (выполнить (value) ::
              (интерпретатор-конфигурации of
                S))))))
  (старт :: S)

```

Обработчик заданий получает на вход задание и возвращает конфигурацию решения, соответствующую этому заданию. Пустая конфигурация решения, определяемая ОП-выражением (`_ is пустая конфигурация решения`), соответствует решенной задаче. Функциональность понятия обработчик-заданий включает ПО-выражение (`обработать A :: I`), задаваемое правилом (`if (обработать A :: I) var A I then (assume ((A is задание) and (I is обработчик-заданий))) (modify ((: value) is конфигурация-решения))`). Заметим, что это правило расширяет функциональность обработчика заданий по сравнению с исходной КМ и, таким образом, оптимизирует ее, поскольку обработчик теперь может возвращать произвольную конфигурацию решения, а не только последовательность задач. Необходимость такой модификации КМ обоснована в [Ануреев, 2012].

Интерпретатор конфигурации получает на вход конфигурацию решения и возвращает модифицированную конфигурацию решения. Определим операционно-онтологическую семантику выражения (`выполнить A :: I`). Пусть ASSUME обозначает выражение (`assume ((A is конфигурация-решения) and (I is интерпретатор-конфигурации))`). Правило (`if (выполнить A :: I) var A I then ASSUME (assume (not (A is задача)) and (not (A is модуль))) (modify ((: value) is конфигурация-решения))`) определяет выполнение конфигурации решения в случае, когда она не является задачей или модулем. Заметим, что в этом правиле не накладывается никаких ограничений на результат выполнения конфигурации решения кроме того, что он снова является конфигурацией решения. При конкретизации КМ посредством

явного задания конструкторов конфигурации решения это правило разбивается на множество правил в зависимости от используемого конструктора. В случае, когда конфигурация решения является задачей или модулем, выполняются специализированные правила перехода, решающую конкретную задачу или выполняющие конкретный модуль.

4. ПРЕИМУЩЕСТВА ТЕХНОЛОГИИ И ОБЛАСТЬ ЕЕ ПРИМЕНЕНИЯ

ФЛП-технология обеспечивает следующие преимущества при проектировании и прототипировании ИС:

- создается формальная логическая модель ИС. Более точно, модель базируется на комбинации операционного (помеченные системы переходов), онтологического (ОНТ-системы) и логического (алгебраические системы и логика безопасности) подходов;
- комбинируются различные подходы к обеспечению безопасности и оптимальности ИС (тестирование, экспертный анализ и дедуктивная верификация);
- формальная модель ИС приближена к описанию на естественном языке, что позволяет использовать ее в качестве документации ИС и при согласовании требований к ИС с заказчиком.

Область применимости ФЛП-технологии ограничивается ИС, которые описываются с помощью систем переходов с конечной структурой состояний (само множество состояний при этом может быть бесконечным) и функциональность которых может быть представлена на языке Atoment.

ЗАКЛЮЧЕНИЕ

В статье описана технология проектирования и прототипирования ИС с акцентом на их безопасность и концептуальную оптимальность.

Обычно под открытостью технологии поддержки того или иного этапа разработки ИС, понимается наличие либо грамотно спроектированного и документированного API к средствам разработки ИС, либо предметно-ориентированного языка разработки ИС (Этот язык может иметь различное представление: текстовое, графовое, в виде форм-шаблонов и т. д.). ФЛП-технология относится к открытым технологиям второго типа. Она не требует программирования. Проектирование и прототипирование ИС проводится экспертами в предметной области, для которой создается ИС, на языке выполнимых спецификаций Atoment в сотрудничестве с менеджерами проекта и разработчиками. Первые обеспечивают взаимодействие с заказчиком (в частности, согласование с ним формальной модели ИС и демонстрацию прототипа ИС). Вторые оценивают разрабатываемый прототип с точки зрения эффективности его реализации с помощью

конкретных программных средств и языков и на конкретных платформах.

Работа выполнена при финансовой поддержке интеграционного проекта РАН № 15/10 «Математические и методологические аспекты интеллектуальных информационных систем» и гранта РФФИ № 13-07-00457 «Методы и средства поддержки разработчиков систем поддержки принятия решений».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Ануреев, 2009] Ануреев И.С. Операционно-онтологический подход к формальной спецификации языков программирования // Программирование. 2009. № 1. С. 1–11.

[Ануреев, 2012] Ануреев И.С. Применение операционно-онтологического подхода к концептуальному моделированию систем поддержки принятия решений // Информационные и математические технологии в науке и управлении. Труды XVII Байкальской Всероссийской конференции. 2012. Том 3. С. 13–19.

[Ануреев, 2008] Ануреев И.С. Язык описания онтологических систем переходов OTSL как средство формальной спецификации программных систем // Вестник НГУ, серия Информационные технологии. 2008. Т. 6, № 3. С. 24–34.

[Голенков, 2011] Голенков В.В. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем // Материалы I международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2011). Минск, 2011. С. 21–58.

[Грибова и др., 2012] Грибова В.В., Клещев А.С. Онтологическая парадигма программирования // Материалы II международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2012). Минск, 2012. С. 213–220.

[Загорюлько, 2009] Загорюлько Ю.А. Технология разработки порталов научных знаний // Программные продукты и системы. 2009. № 4. С. 25–29.

[Загорюлько и др., 2010] Загорюлько Ю.А., Ануреев И.А., Загорюлько Г.Б. Подход к разработке системы поддержки принятия решений на примере нефтегазодобывающего предприятия // Известия Томского политехнического университета. 2010. Т. 316, № 5. С. 127–131.

[Хорошевский, 2008] Хорошевский В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) // Искусственный интеллект и принятие решений. 2008. № 1. С. 80–97.

[Хорошевский, 2009] Хорошевский В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 2) // Искусственный интеллект и принятие решений. 2008. № 4. С. 15–36.

[Хорошевский, 2012а] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 3) // Искусственный интеллект и принятие решений. 2012. № 1.

[Хорошевский, 2012б] Семантические технологии: ожидания и тренды // Материалы II международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2012). Минск, 2012. С. 23–52.

[Anureev, 2012] Anureev I.S. Program specific transition systems // Joint NCC&IIS Bulletin, Series Computer Science. 2012. Vol. 33. P.1–21.

[Gurevich, 2004] Abstract State Machines: An Overview of the Project // Foundations of Information and Knowledge Systems (FoIKS): Proc. Third Internat. Symp. Lect. Notes Comput. Sci. 2004. Vol. 2942. P. 6–13.

OPEN TECHNOLOGY OF FORMAL LOGICAL DESIGN AND PROTOTYPING OF INTELLIGENT SYSTEMS

Anureev I.S., Atuchin M.M.

*A.P. Ershov Institute of Informatics Systems
Siberian Branch of the Russian Academy of
Sciences, Novosibirsk, Russia*

anureev@iis.nsk.su

atuchin.m@gmail.com

The paper presents a technology which allows one to choose an optimal conceptual configuration of an intelligent system at the stage of its design and to build its prototype on basis of formal logical methods. In particular, use of formal logical apparatus provides deductive verification of safety properties of the designed intelligent system. Use of the technology is illustrated by the example of specification of the conceptual model of a decision support system.

INTRODUCTION

The use of formal logical methods to provide the safety and efficiency of software becomes a standard for design and prototyping of industrial software systems (especially critical software). The recent trend to the use of intelligent components in such systems makes the challenge of developing formal methods for intelligent systems taking into account their specificity urgent.

It should also be noted that at present there is transition from the development of specific information systems to creation of technologies of their development. Therefore, an important task is to create technology based on formal logical methods (the FLP-technology) as applied to design and prototyping of information systems.

The paper proposes an approach to solve these two problems.

MAIN PART

The main part consists of four sections.

Section 1 describes the basis of the FLP-technology. This technology is based on specialized labeled transition systems aimed at accelerating the development of the specification and verification of software systems and the executable specification language Atoment used to specify these systems.

Section 2 describes the stages of the design and prototyping of information systems based on the FLP technology.

Section 3 illustrates the use of FLP-technology by the example of specification of the conceptual model of a decision support system.

Section 4 describes the advantages of FLP technology and defines its application domain.

CONCLUSION

The paper describes the technology of design and prototyping of information systems, with emphasis on their safety and conceptual-optimality properties.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

МЕТОД ОРГАНИЗАЦИИ СЛАБОСВЯЗАННЫХ ИНФОРМАЦИОННЫХ РЕСУРСОВ НА ПОРТАЛАХ ЗНАНИЙ

Глоба Л.С., Терновой М.Ю., Новогрудская Р.Л.

Национальный Технический Университет Украины «Киевский Политехнический Институт»

г. Киев, Украина

lgloba@its.kpi.ua

rinan@ukr.net

В работе представлен подход к построению порталов знаний, позволяющий улучшить доступ и обработку слабосвязанной гетерогенной информации представленной на порталах знаний. Метод организации информационных ресурсов на порталах знаний предусматривает интеграцию трех компонент: модели представления знаний, модели бизнес процессов и метода динамического формирования запроса.

Ключевые слова: Модель представления знание, онтология, поиск, модель бизнес процессов, метод динамического формирования запроса, портал знаний.

ВВЕДЕНИЕ

На сегодняшний день существует большое количество подходов к построению порталов знаний, которые базируются на использовании в качестве концептуальной модели онтологию. Целью создания таких порталов является предоставления большому количеству пользователей возможности работать с информацией определенных предметных областей. Такая информация должна быть систематизирована по различным направлениям, необходимо выполнить классификацию знаний для удобства пользования и организации эффективного поиска. В зависимости от предметной области, которой посвящён портал знаний, может проследиваться необходимость создания метамоделей позволяющих включать в состав ресурсов портала знаний разнообразные расчетные задачи, позволяющие получать новые знания, то есть выполнять любую необходимую обработку информации портала. Таким образом, возникает необходимость улучшить доступ к гетерогенной слабосвязанной информации, представленной на порталах знаний. Для повышения эффективности доступа к научным и инженерным знаниям, представленным на специализированных Интернет-порталах знаний, необходимо решить задачи: представление знаний на портале, систематизация и структуризация информации, формализация научных и инженерных знаний, эффективный содержательный поиск, описание сервисов, используемых для решения расчетных задач на портале.

На портале знаний в рассматриваемой работе выделяются 2 вида объектов информационные и функциональные:

- Информационные элементы E_i – статические объекты, не несущее функциональной нагрузки, которые сами по себе не являются процессом и не содержат последовательности взаимосвязанных работ.

- Функциональные элементы E_f – содержат алгоритмическую последовательность действий необходимых для выполнения расчетной задачи.

В процессе анализа и решения описанных выше задач возникли три главные проблемы создания портала знаний:

- Построение модели представления знаний на портале;
- Структуризация и систематизация информационных и функциональных ресурсов;
- Описание взаимодействия функциональных и информационных элементов портала;
- Организация контекстно-зависимого поиска.

Для решения описанных выше проблем необходимо создать метод организации слабосвязанных информационных и функциональных элементов, который будет представлен интеграцией трех компонент:

- Компонента 1 – «Концептуальное описание набора E_i и E_f портала» - служит для описание характеристик и свойств информационных и функциональных элементов портала. Реализуется средством построения модели представления знаний портала;

• Компонента 2 – «Концептуальное описание взаимосвязей E_i и E_f портала» - описывает процесс взаимодействия информационных и функциональных элементов. Выполняется за счет построения моделей бизнес процессов портала;

• Компонента 3 – «Формальное описание динамического формирования запроса» - описывает метод формирования динамического получения знаний согласно построенным моделям. Реализуется за счет построения метода динамического формирования поискового запроса.

Опишем каждую из компонент.

1. Модель представления знаний

Важным этапом построения портала является структуризация и систематизация информации и знаний портала, что позволит конечному пользователю осуществлять просмотр и поиск конкретных сведений в рассматриваемой сфере. Структуризация и размещение на портале организуется удобным для конечного пользователя образом с использованием онтологической модели в качестве модели представления знаний, и позволяет реализовать проблемно-ориентированные средства навигации и поиска по информационному пространству портала [Uschold M.] [Боровикова О.И и др., 2002].

Разработанная модель представления знаний включает такие онтологии как:

- Онтология портала – $O_{\text{портала}} = \{O_1 (O_2, O_3, O_4), O_5\}$.
- Онтология деятельности – O_2 .
- Онтология знания – O_3 .
- Онтология расчетов – O_4 .
- Онтология предметной области.

Для примера рассмотрим компоненту «Онтология расчетов» модели представления знаний для портала «Прочность материалов» [L.S. Globa].

Онтология Расчетов объединяет пять классов понятий, которые описывают расчетные возможности, реализованные на портале: Расчет, Результат, Сервис, Параметры. Данные классы связаны отношениями различных типов. Свойства и характеристики каждого понятия описываются с помощью атрибутов понятий и ограничений, наложенных на область значений. Онтология Расчетов играет важную роль при построении общей онтологии портала. С ее помощью становится возможным описать не только доступные пользователю на портале расчетные задачи – функциональные элементы, но и установить связи между конкретными сервисами (набором сервисов) для реализации конкретных функциональных элементов, а также связи сервисов с информационными ресурсами, что необходимо для решения производственных задач пользователя. Онтология Расчетов формально может быть задана

как $O_4 = \{C_{O_4}, A_{O_4}, R_{O_4}, T_{O_4}, F_{O_4}, D_{O_4}\}$, где $C_{O_4} = \{C_1, C_2, C_3, C_4, C_5\}$, $A_{O_4} = (A_C, A_C, A_C, A_C, A_C)$, $R_{O_4} = (R_{AS_1}(O_4), \dots, R_{AS_7}(O_4), R_{A_1}(O_4), R_{n_1}(O_4), R_{CD_1}(O_4), \dots, R_{CD_5}(O_4))$, $D_{O_4} = (D_{C_1}, D_{C_2}, D_{C_3}, D_{C_4}, D_{C_5})$ [F. Sowa].

Онтология Расчетов состоит из следующих классов понятий:

– Расчет (Расчет_1, Расчет_2, Расчет_2) – $C_1(O_4)$. К этому классу относятся типы расчетов, которые используются для проведения исследования (эксперимента) на портале. К атрибутам класса относятся название, направление и время расчета, методы, используемые при проведении расчетов $C_1(O_4) = (A_{C_1}, D_{C_1})$.

– Результат – $C_2(O_4)$. В этот класс входят результаты расчетов. (Результаты, которые может получить пользователь, проведя исследование (эксперимент) на портале). Результаты могут быть изображены (представлены пользователю) в виде графиков, текстовых описаний, формул. Класс, как бы аналог класса «научный результат», из онтологии знания, однако для Онтологии Расчетов он описывает не уже имеющиеся результаты проведенных ранее расчетов, а расчеты полученные от проводимых на портале расчетов, описанных классом «Расчет» Онтологии Расчетов. К атрибутам класса относятся название, описание вида представления, $C_2(O_4) = (A_{C_2}, D_{C_2})$.

– Сервис – $C_3(O_4)$. Этот класс описывает набор сервисов, которые используются для проведения расчетов. К атрибутам класса относятся название и характеристики сервиса, $C_3(O_4) = (A_{C_3}, D_{C_3})$.

– Интерфейс – $C_4(O_4)$, К этому классу относятся понятия, задействованные в реализации визуализации сервисов и расчетных задач. К атрибутам относится название интерфейса, его тип и вид $C_4(O_4) = (A_{C_4}, D_{C_4})$.

– Параметры (для расчета) – $C_5(O_4)$. Этот класс содержит параметры, учитываемые при расчете (температура, жесткость и т.д.). К атрибутам класса относится название параметра, его численная характеристика $C_5(O_4) = (A_{C_5}, D_{C_5})$.

Классы Онтологии Расчетов связаны следующими отношениями:

– Ассоциативные отношения – $R_{AS}(O_4) = \{C_i(O_4) \times C_j(O_4)\}$:

– «используется для» - связывает сервис с расчетом $R_{AS_1} = \{C_3(O_4) \times C_1(O_4)\}$.

– «находится в процессе» - связывает объект исследования из ОИЗ с расчетом

$$R_{AS_2} = \{C_2(O_3) \times C_1(O_4)\}.$$

– «быть результатом» - связывает результат и расчет $R_{AS_3} = \{C_2(O_4) \times C_1(O_4)\}.$

– «быть полученным при» - связывает результат и параметры расчета $R_{AS_4} = \{C_2(O_4) \times C_5(O_4)\}.$

– «служить параметрами» - связывает класс параметры и класс расчет $R_{AS_5} = \{C_5(O_4) \times C_1(O_4)\}.$

– «быть полученным с помощью» - связывает класс результат с классом сервис $R_{AS_6} = \{C_2(O_4) \times C_3(O_4)\}.$

– «использоваться для» - связывает класс объект исследования из Онтологии Знаний с классом расчет $R_{AS_7} = \{C_1(O_4) \times C_2(O_3)\}.$

Структурные отношения («часть-целое»):

Отношением «часть-целое» связан класс Расчет с классами Расчет_1 (Расчет предельных состояний), Расчет_2 (Расчет прочностных характеристик), Расчет_3 (Расчет деформации) – $C_1 \subset C_{1_1} \wedge C_{1_2} \wedge C_{1_3}.$

Отношение наследования – $R_n(O_4) = a_i, r_i | A_{C_m}(O_4) \rightarrow a_i, r_i | A_{C_k}(O_4)$

Класс Расчет и наследование атрибутов и отношений его подклассами:

$$A(C_1), R(C_1) \rightarrow A(C_{1_1}), R(C_{1_1}),$$

$$A(C_1), R(C_1) \rightarrow A(C_{1_2}), R(C_{1_2}),$$

$$A(C_1), R(C_1) \rightarrow A(C_{1_3}), R(C_{1_3}).$$

Отношения «класс-данные» - $R_{CD}(O_4) = C_j(O_4) \subseteq D_i(O_4):$

Отношение данного вида реализовано для всех классов данной онтологии $C_1(O_4) \subseteq D, A_{C_1} \subseteq A_D,$
 $C_2(O_4) \subseteq D, A_{C_2} \subseteq A_D, C_3(O_4) \subseteq D, A_{C_3} \subseteq A_D,$
 $C_4(O_4) \subseteq D, A_{C_4} \subseteq A_D, C_5(O_5) \subseteq D, A_{C_5} \subseteq A_D.$

Отношения различных видов могут связывать классы, находящиеся в рамках разных онтологий. Суть самих отношений при этом не меняется, а разбиение общей онтологии на несколько компонент реализовано в основном из-за смысловой нагрузки и необходимости разграничения (систематизации) большого количества элементов онтологической модели.

2. Модели бизнес процессов портала

Анализ и моделирование бизнес процессов портала занимает важное место в проектировании портала [J. Becker.].

Действия, которые необходимо выполнить при моделировании бизнес процессов:

1. провести исследование процессов, происходящих на портале во время его жизненного цикла;

2. проанализировать взаимодействие и кооперацию данных и сервисов, информационных и вычислительных ресурсов, баз данных и баз знаний портала, задействованных при его функционировании;

3. провести моделирование вычислительных процессов, происходящих на портале.

В результате проведенного анализа были выделены следующие бизнес процессы:

- БП_1 – Вычислительные задачи;
- БП_2 – Построение (графиков, диаграмм и зависимостей);
- БП_3 – Моделирование (устойчивости конструкций в зависимости от выбранного материала, зависимостей материала от материала);
- БП_4 - Подбор материала по определенному критерию;
- БП_5 - Выбор материала по заданным пользователем характеристикам;
- БП_6 - Расчет необходимых пользователю данных по формулам
- БП_7 – Просмотр (разнообразной теоретической информации, практических результатов, информации о событиях, имеющих отношение к данной предметной области);
- БП_8 – Поиска.

Для построения моделей бизнес процессов была использована нотация UML. Диаграмма вариантов использования нотации UML описывает функциональность интернет-портала знаний в области прочности материалов. Диаграмма кооперации представляет главные объекты портала, которые были выделены во время анализа процессов, происходящих на портале во время его функционирования: Интерфейс, Вычисления, Построение, Просмотр, Поиск, Сервис, Данные. Необходимость построения диаграмм последовательности при моделировании бизнес процессов, происходящих на портале во время его функционирования, следует из преимуществ, которые видны при ее построении [A. Sheer.]:

– легко проследить жизненный цикл определенного объекта;

– четко видно, какой из вызовов (сообщений) исходит от какого объекта и какому объекту передается;

– наглядно представлено, какой объект вызывает инициализацию какого объекта и в какой последовательности;

– легко определить какой вызов следует, за каким для определенного объекта по временной шкале;

– наглядно видна последовательность действий во время функционирования портала.

Определение правильной последовательности позволит:

- Улучшить качество выполнения вычислительных задач (поиска данных и сервисов для реализации различных вычислительных задач).

- Повысить эффективность поиска информации
- Максимизировать скорость связывания различных компонентов портала.

Построенные модели бизнес процессов полностью вписываются в функциональную структуру портала, и можно говорить об их адекватности. С функциональной точки зрения предлагаемый портал[4]:

- обеспечит доступ к базам данных, справочной литературе, мануалам, экспресс информации, ресурсам сети

- позволит организовать просмотр блоков новостей,

- предоставит возможность реализации различных расчетных и вычислительных задач:

Модель БП портала можно представить следующим формализмом:

$$\mu = (O, Z, Ec),$$

где O – объекты, Z – зависимости, Ec – контекстно-зависимые элементы.

Поскольку в качестве объектов БП портала можно выделить процессы так и статические элементы, важно учесть данную особенность с целью ее описания при моделировании БП. Таким образом, общие процессы, протекающие на портале, накопители информации и статические элементы могут быть представлены (2.2).

Множество объектов бизнес процессов портала представляется суммой функциональных и нефункциональных объектов:

$$O = \{Of\} \cup \{Onf\}.$$

Множество нефункциональных объектов является множеством трех параметров $\{O_{Id}^{nf}, Id = 1, 3\}$, где $Onf1 = I$ – Интерфейс, $Onf2 = D$ – Данные, $Onf3 = S$ – Сервис.

Множество функциональных объектов является множеством четырех параметров

$$\{O_{Id}^f, Id = 1, 4\},$$

где $Of1 = C$ – Вычисления, $Of2 = B$ – Построение, $Of3 = R$ – Просмотр, $Of4 = F$ – Поиск.

Определение функциональных объектов связано реальными возможностями, предоставленными пользователю на портале. Возможности портала могут быть определены как: f_{search} – функция поиска информации, f_{review} – функция просмотра информации, $f_{computation}$ – функция реализации различных вычислительных и расчетных задач.

Множество зависимостей бизнес процессов

портала это множество девяти параметров $\{P(V)Id, Id = 1, 9\}$, разбитых на группы в зависимости от переменной, для которой используется параметр $\{VId, Id = 1, 4\}$.

Параметры:

- $P(V)1$ Вход пользователя на портал,
- $P(V)2$ Запрос на $Vi, i = 1, 3$,
- $P(V)3$ Вызов сервиса для $Vi, i = 1, 4$,
- $P(V)4$ Реализация $Vi, i = 1, 4$,
- $P(V)5$ Запрос на $V4$,
- $P(V)6$ Реализация $V4$,
- $P(V)7$ Передача сервиса для $Vi, i = 1, 4$,
- $P(V)8$ Инициализация информации после $P(V)4, P(V)6$,
- $P(V)9$ Возврат информации для просмотра.

Переменные:

- $V1$ = Вычисление
- $V2$ = Построение
- $V3$ = Просмотр
- $V4$ = Поиск

Множество контекстно-зависимых элементов это множество n параметров $\{Ec Id, Id = 1, n\}$. Набор элементов данного множества напрямую зависит от предметной области проектируемого портала и используется для формального описания блоков контекстно-зависимой информации портала при их интеграции в модель БП портала.

3. Метод динамического формирования запроса

Метод динамического формирования запроса оперирует с понятиями определенными в модели представления знаний. Использование такого метода позволит организовать контекстно-зависимый поиск по пространству портала знаний используя формализованные термины предметной области заданные пользователем в поисковом запросе. Метод динамического формирования запроса включает следующую последовательность:

1. Описание понятий онтологической модели.
2. Переход к формальным структурам.
3. Логический вывод.
4. Описание результатов.

Таким образом метод состоит из этапов:

Этапы метода:

1 этап. Описание понятий заданных пользователем в терминах поискового запроса с помощью концептов онтологической модели.

2 этап. Представление терминов онтологической модели, используя формальные структуры.

3 этап. Поиск концепта связанного с формализмами терминов онтологической модели определенными на этапе 2.

4 этап. Вывод результата поиска.

Выполнение метода динамического формирования запроса происходит на нескольких уровнях:

Концептуальный уровень – 1 этап.

Формализованный уровень – 2 этап.

Логический уровень – 3 этап.

Уровень представления – 4 этап.

Рассмотрим пример поискового запроса и процесс поиска результата согласно построенной модели представления знаний на портале. Поисковый запрос: «Найти результаты исследования, проводимые со сталью направленные на определение ее свойств под воздействием высоких температур, полученные Гайном В. А. в 2008 году и описанные в книге “Поведение стали”, изданной институтом Материаловедения Германии», формально будет выглядеть следующим образом [Глоба Л.С., Новогрудская Р.Л.]:

Этап 1+ Этап 2

Класс «Результат исследования»

Отношение «проводится с»:

Класс «Объект исследования»

Атрибут «Объект исследования» = «сталь»

Отношение «направлено на»:

Класс «Назначение исследования»

Атрибут «Назначение исследования» = «воздействие высоких температур»

Отношение «полученные»:

Класс «Персона».

Атрибут «ФИО» = «Гайном В. А.»

&

Класс «Результат исследования»

Атрибут «год» = «2008»

Отношение «описанные»:

Класс «Литература»

Отношение «часть-целое»:

Подкласс «Учебные материалы»

Атрибут «книга» = «Поведение стали»

Отношение «выданной»:

Класс «Организация»

Атрибут «Институт» = «институтом Материаловедения Германии»

Покажем, как будут выглядеть термины поискового запроса в понятия онтологии, приведем их формальное описание. Зададим следующие переменные:

$C_1 = \{ \text{Результат исследования} \}$, $R_{AS_1} = \{ \text{проводится с} \}$, $C_2 = \{ \text{Объект исследования} \}$, $A_{C_2} = \{ \text{сталь} \}$, $R_{AS_2} = \{ \text{направлено на} \}$, $C_3 = \{ \text{Назначение исследования} \}$, $A_{C_3} = \{ \text{воздействию высокими температурами} \}$, $R_{AS_3} = \{ \text{полученные} \}$, $C_4 = \{ \text{Персона} \}$, $A_{C_4} = \{ \text{Гайном В. А.} \}$, $C_5 = \{$

Результат исследования $\}$, $A_{C_5} = \{2008\}$, $R_{AS_4} = \{ \text{описанные} \}$, $C_6 = \{ \text{Литература} \}$, $R_{AS_5} = \{ \text{Подкласс} \}$, $C_7 = \{ \text{Учебные материалы} \}$, $A_{C_6} = \{ \text{Поведение стали} \}$, $R_{AS_6} = \{ \text{выданной} \}$, $C_8 = \{ \text{Организация} \}$, $A_{C_8} = \{ \text{институтом Материаловедения Германии} \}$

Для реализации поискового запроса актуально следующее описание классов: $C_2(A_{C_2}, D_{C_2}, R_{C_2})$, $C_3(A_{C_3}, D_{C_3}, R_{C_3})$, $C_4(A_{C_4}, D_{C_4}, R_{C_4})$, $C_5(A_{C_5}, D_{C_5}, R_{C_5})$, $C_6(A_{C_6}, D_{C_6}, R_{C_6})$, $C_7(A_{C_7}, D_{C_7}, R_{C_7})$

Для реализации поискового запроса существуют следующие отношения (раздел 2.1): отношение вида класс-класс-данные: $R_{AS_6} \leftarrow C_5 \subset C_6$, ассоциативные

отношения: $R_{AS_1} = \{ C_1 \times C_2 \}$, $R_{AS_2} = \{ C_1 \times C_3 \}$, $R_{AS_3} = \{ C_1 \times C_4 \}$, $R_{AS_4} = \{ C_1 \times C_5 \}$, $R_{AS_5} = \{ C_5 \times C_7 \}$.

Этап 3

С помощью заданного формального описания понятий поисковый запрос будет выглядеть следующим образом.

Таблица 1 – Описание поискового запроса

Поисковый запрос	Описание запроса
$C_1 \{$	результаты исследования
$\xrightarrow{R_{AS_2}} A_{C_2}$,	проводимые со сталью
$\xrightarrow{R_{AS_3}} A_{C_3}$,	определение ее свойств при воздействии высокими температурами
$\xrightarrow{R_{AS_4}} A_{C_4} \& A_{C_5}$,	Полученные Гайном В. А. в 2008 году
$\xrightarrow{R_{AS_5}} A_{C_5}$ $\xrightarrow{R_{AS_6}} A_{C_7}$.	описанные в книге “Поведение стали” выданной институтом Материаловедения Германии

Этап 4

На рисунке 1 изображен процесс поиска по элементам онтологии портала. Экземпляры классов и отношения онтологии, заданные пользователем в терминах поискового запроса, выделены курсивом, элемент, который является результатом поиска, выделен жирным. Класс Результат Исследования связан соответствующими отношениями с классами Объект исследования, Персона и Литература. В этих классах находим экземпляры, заданные пользователем: Сталь, Гайн, книга «Поведение стали» и Институт материаловедения Германии, находим экземпляр класса Результат исследования, который им соответствует. Это – Теория прочности.

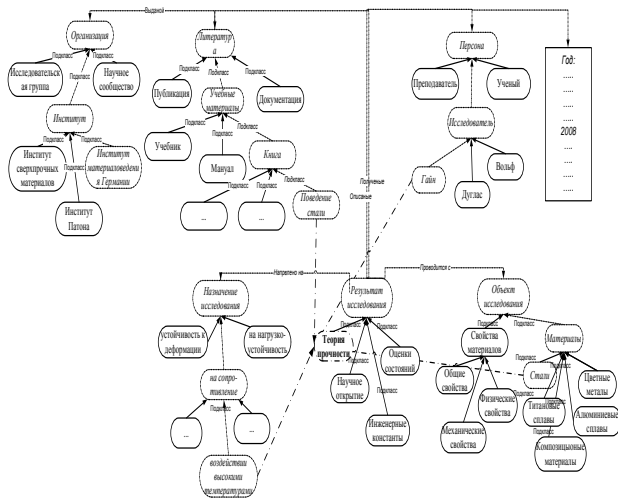


Рисунок 1 - Процесс поиска по элементам онтологии портала

ЗАКЛЮЧЕНИЕ

Целью метода организации информационных ресурсов на порталах знаний является структуризация и систематизация информационных ресурсов порталов знаний, организация контекстно-зависимого поиска а также, генерация новых знаний на основании имеющихся в среде портала. Такой метод базируется на интеграции модели представления знаний, модели бизнес процессов и метода динамического формирования запроса.

В качестве модели представления знаний портала используется онтологическая модель, что позволяет описать информационные и функциональные элементы портала знаний, модель бизнес процессов позволяет описать взаимосвязи таких элементов, а метод динамического формирования запроса – сформировать ответ на поисковый запрос заданный пользователем портала.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Uschold M.] Uschold M., Gruninger M. Ontologies: Principles, Methods and Applications // Knowledge Engineering Review11(2), 1996.
- [Боровикова О.И и др., 2002] Боровикова О.И., Загоруйко Ю.А. Организация порталов знаний на основе онтологий. // Труды международного семинара Диалог'2002 "Компьютерная лингвистика и интеллектуальные технологии". Протвино, 2002. Т.2, С.76-82.
- [L.S. Globa] L.S. Globa, I. Mamuzich, R. L. Novogradsk. Development of a model for the internet portal "strength of materials", Materials and technology 46 (2012) 4, pp. 407-410.
- [F. Sowa] John F. Sowa: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole: New York, 2000
- [J. Becker.] J. Becker. Process Management. A Guide for the Design of Business Processes / J. Becker, M. Kugeler, and M. Rosemann, editors. – Berlin: Springer-Verlag, 2003. – 337 p.
- [A. Sheer.] A. Sheer. Business-processes. Main notions. Methods, Moscow, MetaTechnology, 2000, p. 182
- [Глоба Л.С., Новогрудская Р.Л.] Глоба Л.С., Новогрудская Р.Л. Систематизация информационных ресурсов Интернет-портала «Прочность материалов», Вісник Харк. нац. ун-ту імені В. Н. Каразіна, серія "Математичне моделювання. Інформаційні технології. Автоматизовані системи управління". - Вип. 16, № 927, 2011.

THE METHOD OF KNOWLEDGE PORTAL INFORMATION ELEMENTS ORGANIZATION

Globa L., Ternovoy M., Novogradsk R.

National Technical University of Ukraine "Kyiv Polytechnic Institute"

Kyiv, Ukraine

lgloba@its.kpi.ua

rinan@ukr.net

Article presents the method of heterogeneous information available on portal structuring and systematizing. Such kind of method operates with the original conceptual model that integrates model of portal knowledge representation, model of portal business processes and methods of informational and computational resources collaboration.

The main task is to improve the efficiency of access to information and computing resources by heterogeneous weakly coupled information systematization. While analyzing the main task few problems appeared:

- qualitative knowledge representation;
- systematization of and structuring of information;
- formalization of knowledge based on mathematical models (for chosen subject field);
- a content-based search.

To solve mentioned problems it is in need to develop specific method of heterogeneous information structuring and systematizing while designing knowledge Internet portals in appropriate subject field . It is proposed to design such method with the help of two models and one method drafting:

- knowledge representation model that will form description structures for informational and functional element of the portal,
- business process model that allows to describe collaboration of such elements,
- method of dynamic knowledge generation must operate with notions defined in two above mentioned models.

The method of the knowledge Internet portals development is presented aimed on heterogeneous information available on such portals structuring and systematizing. Such method designing was made using original conceptual model (integrates knowledge representation model and business processes model) and dynamic knowledge generation method based on this model.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОТКРЫТЫЙ ПРОЕКТ, НАПРАВЛЕННЫЙ НА СОЗДАНИЕ ТЕХНОЛОГИИ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Голенков В.В., Гулякина Н.А.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

golen@bsuir.by

guliakina@bsuir.by

В работе рассматривается открытый проект, направленный на создание и развитие технологии компонентного проектирования интеллектуальных систем. Также рассматривается Метасистема IMS.OSTIS, предназначенная для комплексной поддержки проектирования интеллектуальных систем.

Ключевые слова: интеллектуальная система, технология проектирования интеллектуальных систем, семантическая сеть, предметная область.

ВВЕДЕНИЕ

Анализ современных информационных технологий показывает, что наряду с весьма впечатляющими достижениями они имеют целый ряд серьезных недостатков. К числу таких недостатков, в частности, относятся:

- Отсутствует общее унифицированное решение проблемы семантической совместимости компьютерных систем, что порождает высокую трудоемкость создания комплексных интегрированных компьютерных систем. Наиболее остро проблема совместимости компьютерных систем проявляется себя при разработке web-ориентированных систем.

- Высока степень зависимости архитектур компьютерных систем от платформ, на которых они реализованы, что порождает высокую трудоемкость переноса компьютерных систем на новые платформы.

- Отсутствует хорошо продуманная методика конструктивного использования опыта завершенных разработок компьютерных систем, что порождает высокую степень дублирования разработок различных компонентов этих систем. Многократная повторная разработка уже имеющихся технических решений обусловлена (1) либо тем, что известные технические решения плохо интегрируются в разрабатываемую систему, (2) либо тем, что эти технические решения трудно найти.

- Отсутствует хорошо продуманная унифицированная методика совершенствования различных архитектурных уровней компьютерных

систем в процессе их сопровождения, что порождает высокую трудоемкость такого совершенствования. Обновление компьютерных систем (в частности, обновление баз данных, обновление контента web-порталов) часто сводится к разработке различного рода "заплаток", которые устраняют не причины выявленных недостатков обновляемых компьютерных систем, а только некоторые следствия этих причин.

- Современные информационные технологии не ориентированы на широкий круг разработчиков прикладных компьютерных систем.

Перечислим некоторые общие подходы, направленные на устранение указанных выше недостатков современных информационных технологий.

- Все, что в настоящее время делается в информационных технологиях необходимо фундаментально переосмыслить и максимально возможным образом упростить (!), привести в стройную систему и сделать доступным любому (!) грамотному человеку, не являющемуся специалистом в области информационных технологий, но желающему самостоятельно (!) сделать прототип компьютерной системы в интересующей его предметной области.

- Необходимо минимизировать необходимость участия специалистов в области информационных технологий в разработке прикладных компьютерных систем и полностью исключить такое участие в разработке прототипов этих систем. Т.е. между разработчиками прототипа прикладной компьютерной системы (каковыми должны быть эксперты по соответствующей

предметной области) и самим этим прототипом не должно быть никаких посредников-программистов. Такое посредничество нередко приводит к снижению качества разрабатываемой прикладной системы. Специалисты в области информационных технологий должны, в основном, заниматься не разработкой приложений, а разработкой и совершенствованием технологий (моделей, средств и методов), создающих комфортные условия для разработчиков приложений.

- Необходимо ориентироваться на разработку таких моделей представления и обработки информации, которые бы носили для любого грамотного человека максимально комфортный (понятный, человекоподобный) характер. Т.е. необходимо не только пользовательский интерфейс компьютерных систем, но и само внутреннее устройство этих систем сделать понятным и комфортным для пользователя.

1. Актуальность интеллектуальных систем в контексте развития современных информационных технологий

Одним из направлений устранения указанных выше недостатков современных информационных технологий является расширение применения методов и средств искусственного интеллекта в традиционных компьютерных системах. Интеллектуализация компьютерных систем делает их более гибкими, открытыми, дает возможность перевести проектирование компьютерных систем на более высокий уровень.

Сама по себе трансформация традиционных компьютерных систем в интеллектуальные системы, основанные на знаниях, не решает указанных выше проблем современных информационных технологий, но создает серьезные предпосылки для их решения, т.к. интеллектуальные системы по сравнению с традиционными системами являются более гибкими и открытыми. Кроме того, для интеллектуальных систем (в отличие от традиционных) можно более конструктивно решать проблему их семантической совместимости.

Трансформация традиционных компьютерных систем в интеллектуальные – важная тенденция эволюции компьютерных систем, направленная на повышение их эффективности и конкурентоспособности. Такая трансформация предполагает:

- унифицированную систематизацию и структуризацию всей обрабатываемой в системе информации, что означает переход от субъективно структурированных данных к базе знаний, т.е. к принципиально новому качеству представления и структуризации обрабатываемой информации;

- переход к программированию, ориентированному на обработку баз знаний (семантически структурированных данных), что предполагает расширение работ по созданию

принципиально новых моделей обработки информации и языков программирования.

Высокая актуальность интеллектуальных систем для конечного пользователя обусловлена не только тем, что требуется автоматизация решения все более и более сложных задач анализа, синтеза, принятия решения, но и тем, что повышение эффективности эксплуатации компьютерных систем настоятельно требует включения в состав каждой компьютерной системы help-подсистемы, которая должна фактически стать интеллектуальным консультантом и учителем по эксплуатации соответствующей компьютерной системы.

Тем не менее, основные проблемы повышения эффективности интеллектуальных систем касаются не конечных пользователей, а их разработчиков. Современным требованиям не удовлетворяет не столько качество разрабатываемых интеллектуальных систем, сколько недостаточное их количество, высокая трудоемкость и наукоемкость их разработки и, соответственно, недопустимо большие сроки их проектирования.

Подчеркнем при этом, что технологии проектирования интеллектуальных систем, как и информационных технологий в целом, имеют во многом похожие проблемы и требуют существенного переосмысления.

2. Основные предпосылки создания эффективных технологий проектирования интеллектуальных систем

К числу основных условий, необходимых для создания в ближайшем будущем эффективных технологий проектирования интеллектуальных систем, можно отнести:

- (1) Осознание того, что основным практическим результатом искусственного интеллекта являются не сами интеллектуальные системы, а мощные и эффективные технологии их разработки.

- (2) Осознание того, что создание конкурентоспособных технологий разработки интеллектуальных систем требует согласованного конструктивного взаимодействия представителей всех научных дисциплин, входящих в состав искусственного интеллекта. Как минимум необходимо разработать и постоянно совершенствовать согласованную *онтологию проектирования интеллектуальных систем*.

- (3) Осознание того, что без активного и массового привлечения молодых участников создания нового поколения проектирования интеллектуальных систем, указанные технологии не только не будут созданы, но и не смогут в дальнейшем динамично развиваться. Это обусловлено тем, что создание новых технологий – процесс наукоемкий, долгосрочный, требующий

согласованной деятельности большого количества участников и обеспечения преемственности. Очевидно, что для организации эффективного привлечения молодых участников разработки новых технологий проектирования интеллектуальных систем необходима хорошо продуманная система соответствующих организационных форм и методов.

3. Основные положения предлагаемого подхода к созданию эффективной технологии проектирования интеллектуальных систем

К числу ключевых положений, лежащих в основе предлагаемой технологии проектирования интеллектуальных систем, относятся следующие положения:

(1) Применяется методика компонентного проектирования, которая является фактором зрелости любых технологий и которая основывается на постоянно расширяемых библиотеках многократно используемых компонентов (типовых технических решений). Создание технологии компонентного проектирования требует:

- обеспечения совместимости (интегрируемости) компонентов интеллектуальных систем на основе унификации представления этих компонентов;
- разработки онтологии компонентного проектирования интеллектуальных систем;
- создание библиотек многократно используемых (типовых) компонентов интеллектуальных систем и уточнения типологии таких компонентов (предметные онтологии, многократно используемые фрагменты баз знаний, машины вывода, интерфейсные компоненты и т.д.);
- разработка языка спецификации многократно используемых компонентов интеллектуальных систем;
- создание средств компьютерной поддержки синтеза интеллектуальных систем из имеющихся компонентов.

(2) Формальные модели проектируемых интеллектуальных систем строятся на основе унифицированных семантических сетей, что создает необходимые условия для обеспечения семантической совместимости интеллектуальных систем и их компонентов (т.е. для решения задач их интеграции).

(3) Для снижения трудоемкости разработки и модификации (обновления) создаваемых и постоянно совершенствуемых интеллектуальных систем обеспечивается максимально возможная независимость процесса обновления базы знаний интеллектуальной системы от процесса обновления моделей и методов обработки знаний и от процесса обновления средств технической реализации интеллектуальной системы (включая переход на новые платформы). Суть предлагаемого подхода заключается в создании такого способа

представления знаний в интеллектуальных системах, который бы зависел только от семантики описываемой предметной области и никак не зависел ни от виртуальных машин обработки знаний, ни от различных вариантов и платформ технической реализации интеллектуальной систем. В качестве основы такого способа представления знаний предлагаются абстрактные (!) унифицированные семантические сети с базовой теоретико-множественной интерпретацией.

Рассматриваемое положение, лежащее в основе предлагаемой технологии компонентного проектирования интеллектуальных систем, позволяет декомпозировать процесс проектирования интеллектуальной системы на несколько достаточно независимых и, следовательно, параллельно (!) выполняемых процессов. К числу таких подпроцессов, в частности, относятся:

- разработка и обновление базы знаний проектируемой интеллектуальной системы;
- разработка и обновление абстрактной модели (виртуальной машины) обработки знаний;
- разработка и обновление абстрактной модели интерфейса интеллектуальных систем с различными внешними субъектами (в первую очередь, - с пользователями);
- создание и обновление средств технической реализации памяти для хранения обрабатываемой базы знаний и средств технической реализации разработанных абстрактных моделей обработки знаний и моделей интерфейсов интеллектуальных систем.

(4) Предлагаемая технологий компонентного проектирования интеллектуальных систем оформляется как интеллектуальная метасистема, которая строится по той же предлагаемой технологии и аккумулирует в себе все накопленные к текущему моменту модели, средства (в том числе библиотеки типовых компонентов) и методы, входящие в состав предлагаемой технологии.

(5) Постоянное развитие технологии компонентного проектирования интеллектуальных систем осуществляется в рамках open-source-проекта. Массовая технология проектирования интеллектуальных систем должна быть доступной и постоянно открытой. Более того, к ее развитию должны быть привлечены разработчики различных приложений, передавая свой опыт.

(6) В рамках предлагаемой технологии проектирования интеллектуальных систем особое внимание необходимо обратить:

- на технологию обновления (совершенствования) интеллектуальных систем в процессе их эксплуатации;
- на метатеchnологию постоянного обновления (совершенствования) самой технологии компонентного проектирования интеллектуальных систем (т.е. метасистемы, предназначенной для поддержки проектирования интеллектуальных систем).

4. Достоинства и особенности методики компонентного проектирования

Использование методики компонентного проектирования, в основе которой лежат постоянно (!) пополняемые библиотеки многократно используемых (типовых) компонентов проектируемых систем, является признаком достаточно высокого уровня развития соответствующей технологии проектирования.

Без мощных, хорошо структурированных библиотек типовых совместимых технических решений (многократно используемых компонентов) проектируемых систем невозможно добиться:

- ни существенного сокращения сроков проектирования, т.к. невозможно вписаться в разумные сроки, если не использовать человеческий опыт аналогичных разработок;
- ни повышения качества проектирования, т.к. в каждой разрабатываемой системе всегда есть компоненты, имеющие хорошо (!) сделанные аналоги в других разработках, превзойти которые в разумные сроки невозможно.

Подчеркнем при этом, что интеграция указанных аналогов в разрабатываемую систему даже при наличии хорошо оформленных исходных текстов этих аналогов всегда приводит к трудноисправляемым ошибкам, если заранее на уровне самой технологии не предусмотрена полная (!) совместимость таких аналогов (типовых технологических решений, многократно используемых компонентов) с проектируемыми системами. Это означает, что типовые технологические решения в технологии компонентного проектирования "вставляются" в разрабатываемую систему без каких бы то ни было их доработок.

Технология компонентного проектирования может существовать и развиваться только в форме открытого проекта, в котором стирается грань между разработчиками прикладных систем и теми, кто разрабатывает, совершенствует и сопровождает саму технологию. Каждый разработчик прикладной системы может без особого труда выделить в разрабатываемой им системе те компоненты (фрагменты), которые могут быть использованы в каких-либо других системах. Если библиотеки многократно используемых компонентов будут активно пополняться самими разработчиками приложений, передающими в такой конструктивной форме свой опыт другим разработчикам, то технология компонентного проектирования будет иметь практически неограниченный жизненный цикл.

Что же препятствует созданию технологии компонентного проектирования. Только человеческий фактор, а, точнее, примитивный эгоистический подход к конкуренции разрабатываемых систем. Конкурировать должны сами разрабатываемые системы, а не частные

технологические решения, используемые в этих разработках.

Эффективная технология компонентного проектирования появится только тогда, когда сформируется "критическая масса" разработчиков прикладных систем, участвующих в пополнении библиотек многократно используемых компонентов проектируемых систем.

Важная особенность методики компонентного проектирования интеллектуальных систем заключается в том, что совокупность всех компонентов, входящих в состав библиотеки многократно используемых компонентов баз знаний и библиотеки многократно используемых компонентов виртуальных машин обработки знаний, удобно трактовать как формальную модель глобальной (интегрированной) интеллектуальной системы, обладающей всеми человеческими знаниями и навыками, которые к текущему моменту удалось формализовать. В предельном случае в состав такой глобальной интеллектуальной системы должны входить базы знаний и виртуальные машины обработки знаний всех разработанных прикладных интеллектуальных систем. Это означает, что компонентное проектирование каждой новой прикладной системы можно трактовать как выделение соответствующего фрагмента глобальной интеллектуальной системы с последующим "дописыванием" тех фрагментов, которые отсутствуют в текущем состоянии этой глобальной системы.

Аналогичным образом можно трактовать не только методику компонентного проектирования формальных моделей интеллектуальных систем, но и методику компонентного проектирования их баз знаний.

Методика компонентного проектирования совместимых баз знаний исходит из того, что:

- существует глобальное абстрактное семантическое пространство всех знаний, накопленных человечеством к текущему моменту времени (назовем это глобальной базой знаний);
- существует унифицированный способ кодирования (представления) этих знаний;
- существует достаточно большая часть человеческих знаний, которая представлена указанным унифицированным способом и структурирована на предмет выделения наиболее часто используемых (востребованных) фрагментов этих знаний. Заметим при этом, что ценность знаний далеко не всегда определяется частотой их использования.

5. Структура проекта, направленного на создание и развитие предлагаемой технологии компонентного проектирования интеллектуальных систем.

Для того, чтобы говорить о создании предлагаемой технологии компонентного проектирования интеллектуальных систем (которую будем называть *Технологией OSTIS* – Open Semantic Technology for Intelligent Systems), необходимо, прежде всего, уточнить, что мы будем называть такой технологией. С одной стороны, указанную технологию можно трактовать как специальным образом организованную деятельность по проектированию интеллектуальных систем. Такую комплексную деятельность будем называть *Проектом OSTIS*.

С другой стороны, *Технологию OSTIS* можно трактовать как комплекс моделей, средств и методов и средств, предназначенных для разработки интеллектуальных систем, а также для постоянного обновления и совершенствования самой этой технологии. Актуальность и принципы, лежащие в основе предлагаемой нами технологии компонентного проектирования интеллектуальных систем, рассматривались в докладах на 1-ой и 2-ой конференциях OSTIS [Голенков, 2011], [Голенков, 2012].

В основе *Технологии OSTIS* лежит применение в качестве способа представления знаний унифицированных семантических сетей с базовой теоретико-множественной интерпретацией их элементов. Такой способ представления знаний назван *SC-кодом* (Semantic Code), а семантические сети, представленные в SC-коде, названы *sc-графами*. Унифицированные семантически модели интеллектуальных систем и их компоненты, в основе которых лежат sc-графы, названы *sc-моделями*. Частными видами sc-моделей являются:

- *sc-модели интеллектуальных систем;*
- *sc-модели различного вида знаний;*
- *sc-модели баз знаний;*
- *sc-модели машин обработки знаний;*
- *sc-модели памяти машин обработки знаний* (sc-память, sc-хранилище);
- *sc-модели агентов обработки знаний;*
- *sc-модели пользовательских интерфейсов;*

В предлагаемой технологии указанный комплекс моделей, средств и методов реализуется в виде интеллектуальной метасистемы, которая строится по той же самой предлагаемой технологии проектирования интеллектуальных систем, и, соответственно этому, названа *Метасистемой IMS.OSTIS* (Intelligent Metasystem of Open Semantic Technology for Intelligent Systems). Это означает, что комплекс моделей, средств и методов, предназначенных для постоянного обновления и совершенствования предлагаемой технологии, есть не что иное, как комплекс моделей, средств и

методов постоянного обновления и совершенствования указанной интеллектуальной метасистемы, который может быть использован не только для развития и сопровождения этой метасистемы, но и любых других интеллектуальных систем, построенных на основе предлагаемой технологии.

Если говорить в целом о *Проекте OSTIS*, направленном на создание и развитие предлагаемой *Технологии OSTIS*, то, кроме разработки самой этой технологии в виде *Метасистемы IMS.OSTIS*, в состав такого комплексного проекта должен входить целый спектр подпроектов, направленных на расширение контингента разработчиков интеллектуальных систем, использующих предлагаемую технологию, на расширение спектра разрабатываемых приложений, на расширение контингента разработчиков самой технологии (точнее, *Метасистемы IMS.OSTIS*) и, в первую очередь, на расширение библиотеки многократно используемых (типовых) компонентов интеллектуальных систем.

Таким образом, комплексный *Проект OSTIS* включает в себя:

(1) *Проект IMS.OSTIS*, направленный на создание технологии OSTIS в форме интеллектуальной *Метасистемы IMS.OSTIS*, содержащей:

- комплекс моделей, методов и средств, осуществляющих комплексную поддержку проектирования интеллектуальных систем;
- семейство библиотек многократно используемых компонентов интеллектуальных систем (типовых технических решений);

(2) Неограниченное семейство прикладных проектов, направленных:

- на разработку интеллектуальных систем;
- на пополнение библиотек многократно используемых компонентов (выделяемых из разрабатываемых систем);
- на тестирование используемой технологии проектирования интеллектуальных систем.

Важным дополнительным результатом таких проектов является формирование открытого сообщества разработчиков, участвующих в развитии технологии OSTIS.

(3) Семейство проектов, направленных на организацию подготовки специалистов по проектированию интеллектуальных систем, на подготовку и проведение мероприятий, способствующих развитию технологий проектирования интеллектуальных систем (конференций, семинаров, выставок, конкурсов проектов). Примерами таких проектов, в частности, являются:

- Проект развития кафедры интеллектуальных информационных технологий БГУИР, в которой подготовка студентов осуществляется на основе

их активного привлечения к разработке различных интеллектуальных систем (*Проект ИТ.BSUIR*). Важнейшим подпроектом этого проекта является *Проект IS.ИТ.BSUIR*, направленный на разработку и постоянное совершенствование интеллектуальной системы автоматизации деятельности указанной кафедры и управления этой деятельностью;

- Проект подготовки и проведения ежегодных конференций OSTIS (*Проект Conf.OSTIS*), важнейшим подпроектом которого является *Проект IS.Conf.OSTIS*, направленный на разработку интеллектуальной системы поддержки подготовки и проведения конференций OSTIS;

6. Структура интеллектуальной метасистемы поддержки проектирования интеллектуальных систем

Метасистема IMS.OSTIS, предназначенная для комплексной поддержки проектирования интеллектуальных систем, представляет собой интеллектуальную систему, которая построена на основе технологии OSTIS и, соответственно этому, состоит из следующих основных компонентов:

- унифицированной семантической модели (sc-модели) базы знаний *Метасистемы IMS.OSTIS*;
- унифицированной семантической модели (sc-модели) машины обработки знаний *Метасистемы IMS.OSTIS*;
- унифицированной семантической модели (sc-модели) пользовательского интерфейса *Метасистемы IMS.OSTIS*;
- технической реализации интерпретатора указанных унифицированных семантических моделей. Сюда входит техническая реализация sc-памяти для хранения унифицированных семантических сетей (sc-графов) и техническая реализация интерпретатора программ, описывающих обработку информации в указанной sc-памяти.

Кроме этого, *Метасистема IMS.OSTIS* представляет собой результат интеграции целого комплекса взаимосвязанных самостоятельных (!) интеллектуальных подсистем, построенных на основе общей *sc-памяти* и ориентированных на поддержку решения разных классов проектных задач. К числу таких подсистем, в частности, относятся:

- Подсистема *Метасистемы IMS.OSTIS*, предназначенная для поддержки проектирования унифицированных семантических моделей баз знаний разрабатываемых интеллектуальных систем;
- Подсистема *Метасистемы IMS.OSTIS*, предназначенная для поддержки проектирования унифицированных семантических моделей машин обработки знаний разрабатываемых интеллектуальных систем;

- Подсистема *Метасистемы IMS.OSTIS*, предназначенная для поддержки проектирования унифицированных семантических моделей пользовательских интерфейсов разрабатываемых интеллектуальных систем;
- Подсистема *Метасистемы IMS.OSTIS*, предназначенная для поддержки выбора и компоновки подходящего варианта технической реализации интерпретатора унифицированных семантических моделей разрабатываемых интеллектуальных систем;
- Подсистема *Метасистемы IMS.OSTIS*, предназначенная для поддержки проектирования интеллектуальных help-систем для конечных пользователей разрабатываемых интеллектуальных систем;
- Подсистема *Метасистемы IMS.OSTIS*, предназначенная для информационного обслуживания и обучения пользователей *Метасистемы IMS.OSTIS*;
- Подсистема *Метасистемы IMS.OSTIS*, предназначенная для автоматизации сопровождения и обновления *Метасистемы IMS.OSTIS*;
- Подсистема *Метасистемы IMS.OSTIS*, предназначенная для поддержки проектирования подсистем разрабатываемых интеллектуальных систем, осуществляющих управление проектированием, сопровождением и обновлением разрабатываемых интеллектуальных систем;
- Подсистема *Метасистемы IMS.OSTIS*, предназначенная для управления проектированием, сопровождением и обновлением *Метасистемы IMS.OSTIS*;
- Подсистема *Метасистемы IMS.OSTIS*, предназначенная для поддержки проектирования интеллектуальных справочных систем различного назначения, осуществляющих комплексное информационное обслуживание пользователей в заданных предметных областях;
- Подсистема *Метасистемы IMS.OSTIS*, предназначенная для поддержки проектирования интеллектуальных обучающих систем, реализующих различные стратегии и методы управления обучением.

Все перечисленные подсистемы интеллектуальной *Метасистемы IMS.OSTIS* являются продуктами соответствующих подпроектов *Проекта OSTIS*.

Таким образом, *Метасистема IMS.OSTIS* – это интеллектуальная система, которая:

- (1) построена на основе технологии OSTIS;
- (2) обеспечивает комплексную поддержку проектирования интеллектуальных систем в соответствии с *Технологией OSTIS*;
- (3) обладает всеми необходимыми для этого умениями и навыками (в том числе знаниями о соответствующих моделях, методах и средствах проектирования);

(4) является формой материального воплощения указанно *Технологии OSTIS*;

(5) осуществляет автоматизацию управления проектом, направленным на постоянное совершенствование самой себя.

В целом *Метасистему IMS.OSTIS* можно считать:

- порталом знаний, необходимых разработчикам интеллектуальных систем;
- интеллектуальной системой (средой) автоматизированного проектирования интеллектуальных систем различного назначения;
- интеллектуальной системой, осуществляющей не только информационную поддержку проектирования интеллектуальных систем, но и поддержку управления коллективным проектированием этих систем (в том числе самой себя).

Пользователями *Метасистемы IMS.OSTIS* являются:

- читатели, желающие познакомиться с технологией OSTIS (для них нужна навигация по базе знаний IMS.OSTIS и средства отображения);
- разработчики прикладных интеллектуальных систем, проектируемых на основе Технологии OSTIS (им нужны методика и средства проектирования, а также библиотеки многократно используемых компонентов);
- разработчики самой *Метасистемы IMS.OSTIS*
 - разработчики *sc-модели базы знаний IMS.OSTIS* (им нужны средства интеграции, редактирования, верификации знаний);
 - разработчики *sc-модели машины обработки знаний IMS.OSTIS*;
 - разработчики *sc-модели пользовательского интерфейса IMS.OSTIS*;
 - разработчики средств технической реализации *Метасистемы IMS.OSTIS*.

7. Унифицированная семантическая модель базы знаний интеллектуальной Метасистемы, предназначенной для поддержки проектирования интеллектуальных систем

Для того, чтобы при просмотре базы знаний *Метасистемы IMS.OSTIS* обеспечить пользователю комфортную навигацию по семантическому пространству этой базы знаний, не требующую априорного знания внешних идентификаторов (имен) искомым элементов семантической сети, представляющей указанную базу знаний, целесообразно выделить компактный набор узлов этой семантической сети, от которых достаточно легко (за небольшое число шагов) можно по связям, имеющим четкую семантическую интерпретацию, "добраться" до любого искомого элемента семантической сети.

Указанный набор узлов базы знаний Метасистемы IMS.OSTIS будем называть ключевыми узлами навигации по базе знаний Метасистемы IMS.OSTIS. К числу таких ключевых узлов относятся:

- *Проект OSTIS*
- *Проект IMS.OSTIS*
- *Метасистема IMS.OSTIS*
- *Проект Conf.OSTIS*
- *Документация Проекта IMS.OSTIS*
- *Документация IMS.OSTIS*
- *Подпроекты Проекта OSTIS*
- *Продукты Проекта OSTIS*
- *Горячие проектные задачи Проекта OSTIS*
- *Исполнители Проекта OSTIS*
- *Партнеры Проекта OSTIS*
- *Новости Проекта OSTIS*
- *Форумы Проекта OSTIS*

Рассмотрим семантические окрестности перечисленные узлов.

Проект OSTIS

= *Технология OSTIS как определенным образом организованная деятельность, направленная на компонентное проектирование интеллектуальных систем на основе унифицированных семантических сетей, а также на совершенствование и распространение самой этой технологии.*

= *Специальным образом организованная деятельность, направленная на разработку интеллектуальных систем различного назначения, на постоянное совершенствование этой деятельности, на расширение контингента участников этой деятельности и повышение эффективности их участия*

=> *примечание**:

[Основным продуктом Проекта OSTIS является сам этот проект, т.е. сама соответствующая деятельность]

=> *подпроект**:

- *Проект IMS.OSTIS*
- *Проект IS.GraphTheory*
- *Проект IS.Geometry*
- *Проект Conf.OSTIS*
=> *подпроект**:
Проект IS.Conf.OSTIS
- *Проект ИТ.BSUIR*
=> *подпроект**:
Проект IS.ИТ.BSUIR

Проект IMS.OSTIS

= *Технология OSTIS как определенный комплекс моделей, методов и средств разработки интеллектуальных систем на основе библиотеки многократно используемых компонентов*

<= *основной подпроект**:

Проект OSTIS

=> *пояснение**:

[Это метапроект, направленный на разработку, сопровождение и обновление интеллектуальной

Метасистемы *IMS.OSTIS*, для комплексного проектирования интеллектуальных систем. Указанная метасистема рассматривается нами как конструктивная форма воплощения предлагаемой нами технологии компонентного проектирования интеллектуальных систем.]

Метасистема *IMS.OSTIS*

- = *IMS.OSTIS*
- = *Технология OSTIS*
- = *SC-технология проектирования интеллектуальных систем*
- = *Intelligent Meta System for Open Semantic Semantic Technology for Intelligent Systems*
- = www.ostis.net
- = *Технология OSTIS как комплекс моделей, методов и средств проектирования интеллектуальных систем, оформленный в виде интеллектуальной метасистемы*
- = *Комплекс моделей, методов и средств проектирования интеллектуальных систем на основе унифицированных семантических сетей (sc-графов)*
- = *Комплекс моделей, средств и методов, предназначенных для компонентного проектирования интеллектуальных систем на основе унифицированных семантических сетей с базовой теоретико-множественной интерпретацией и оформленных в виде интеллектуальной метасистемы, осуществляющей информационную и инструментальную поддержку такого проектирования*
- = *Интеллектуальная метасистема комплексной поддержки проектирования интеллектуальных систем согласно технологии OSTIS*
- = *Форма материализации технологии OSTIS в виде интеллектуальной метасистемы*
- <= продукт*:
Проект IMS.OSTIS
- ∈ *технология компонентного проектирования интеллектуальных систем*
 - ⊂ *технология проектирования интеллектуальных систем*
 - = *Семейство всевозможных технологий проектирования интеллектуальных систем*
 - ⊂ *технология*
- => *пояснение**:
[Это инфраструктура, позволяющая разработчикам интеллектуальных систем ускорять процесс разработки таких систем и самим участвовать в развитии этой инфраструктуры (библиотеки ip-компонентов) в рамках *Проекта IMS.OSTIS*.]
- => *пояснение**:
[Это комплекс моделей, методов (в том числе организационных) и средств (в том числе компьютерной поддержки), предназначенных для быстрого проектирования интеллектуальных систем и развивающейся как wiki-проект в рамках *Проекта IMS.OSTIS*.]

технология

=> *пояснение**:

[Это комплекс моделей, методов и средств, предназначенных для осуществления некоторого вида целенаправленной деятельности. Каждая такая деятельность имеет свой продукт (результат, сухой остаток) и свою цель (требования, предъявляемые к этому проекту – условия, которым он должен удовлетворять).]

Проект *Conf.OSTIS*

- = *Проект, направленный на подготовку и проведение ежегодных конференций по открытым семантическим технологиям проектирования интеллектуальных систем*
- => *подпроект**:
Проект IS.Conf.OSTIS
 - = *Проект, направленный на разработку интеллектуальной системы, предназначенной для информационной поддержки подготовки и проведения ежегодных конференций по открытым семантическим технологиям проектирования интеллектуальных систем*
- => *продукт**:
Система IS.Conf.OSTIS
 - = *IS.Conf.OSTIS*
 - = www.conf.ostis.net
- => *подпроект**:
Проект Conf.OSTIS-2013
 - => *продукт**:
Конференция OSTIS-2013

Документация *Проекта IMS.OSTIS*

<= *декомпозиция**:

- {
 - *Введение. Проект IMS.OSTIS*
 - = *Краткая характеристика Проекта IMS.OSTIS*
 - *Документация Метасистемы IMS.OSTIS*
 - = *Описание конечного продукта Проекта IMS.OSTIS*
 - *Раздел. Текущая стадия разработки IMS.OSTIS*
 - *Раздел. Основные направления работ по созданию очередной версии Метасистемы IMS.OSTIS*
 - *Раздел. Стратегия дальнейшего развития разрабатываемой Метасистемы IMS.OSTIS*
 - *Раздел. Бизнес-модель и бизнес-план развития Проекта IMS.OSTIS*
 - *Раздел. Сведения о команде Проекта IMS.OSTIS*
 - *Раздел. Файл презентации Проекта IMS.OSTIS*

=> *аннотация**:

[*Документация Проекта IMS.OSTIS*, направленного на разработку и постоянное совершенствование Метасистемы *IMS.OSTIS*, является важнейшим разделом базы знаний этой

метасистемы. Указанный раздел базы знаний является основой для организации управления проектированием. Проект IMS.OSTIS рассматривается как иерархическая система его проектов.]

Документация IMS.OSTIS

= Документация Метасистемы IMS.OSTIS

= Документация Технологии OSTIS как комплекса моделей, средства методов проектирования интеллектуальных систем на основе

унифицированных семантических моделей

<= декомпозиция*:

- {
- Раздел. Основные характеристики Метасистемы IMS.OSTIS
- Раздел. Теория унифицированных семантических моделей интеллектуальных систем и их компонентов
- Раздел. Способы представления унифицированных семантических сетей
- Раздел. Библиотека интерпретаторов унифицированных семантических моделей интеллектуальных систем
- Раздел. Подсистема Метасистемы IMS.OSTIS для поддержки проектирования унифицированных семантических моделей баз знаний
- Раздел. Подсистема Метасистемы IMS.OSTIS для поддержки проектирования унифицированных семантических моделей процедурных и непроцедурных программ, ориентированных на обработку унифицированных семантических моделей баз знаний
- Раздел. Подсистема Метасистемы IMS.OSTIS для поддержки проектирования унифицированных семантических моделей машин обработки баз знаний
- Раздел. Подсистема Метасистемы IMS.OSTIS для поддержки проектирования унифицированных семантических моделей интерфейсов интеллектуальных систем с внешней средой (в том числе с пользователями)
- Раздел. Подсистема Метасистемы IMS.OSTIS для поддержки проектирования унифицированных семантических моделей help-подсистем разрабатываемых интеллектуальных систем для консультационного обслуживания и обучения конечных пользователей этих систем
- Раздел. Подсистема Метасистемы IMS.OSTIS для поддержки проектирования унифицированных семантических моделей подсистем разрабатываемых интеллектуальных систем, предназначенных для информационной поддержки и автоматизации обновления этих интеллектуальных систем, в том числе в ходе их эксплуатации

- Раздел. Подсистема Метасистемы IMS.OSTIS для поддержки проектирования унифицированных семантических моделей подсистем разрабатываемых интеллектуальных систем, предназначенных для информационной поддержки и автоматизации управления проектированием этих интеллектуальных систем, в том числе и в ходе их эксплуатации

- Раздел. Подсистема Метасистемы IMS.OSTIS для поддержки комплексного проектирования различных классов интеллектуальных систем на основе их унифицированных семантических моделей

- Раздел. Подсистема Метасистемы IMS.OSTIS для автоматизации обучения проектированию интеллектуальных систем на основе их унифицированных семантических моделей

- Публикации. Метасистема IMS.OSTIS = Публикации по моделям, средствам и методам проектирования интеллектуальных систем на основе унифицированных семантических моделей

- Библиография. Метасистема IMS.OSTIS = Библиография по моделям, средствам и методам проектирования интеллектуальных систем

}

Раздел. Основные характеристики Метасистемы IMS.OSTIS

<= декомпозиция*:

- {
- Подраздел. Тип разрабатываемой Метасистемы IMS.OSTIS
- Подраздел. Состав и объем базы знаний Метасистемы IMS.OSTIS
- Подраздел. Структуризация и систематизация знаний в Метасистеме IMS.OSTIS
- Подраздел. Структура и состав машины обработки знаний Метасистемы IMS.OSTIS
- Подраздел. Структура и состав пользовательского интерфейса Метасистемы IMS.OSTIS
- Подраздел. Интерфейс Метасистемы IMS.OSTIS с другими компьютерными системами
- Подраздел. Help-подсистема Метасистемы IMS.OSTIS для консультации и обучения ее пользователей
- Подраздел. Подсистема Метасистемы IMS.OSTIS для автоматизации ее проектирования
- Подраздел. Подсистема Метасистемы IMS.OSTIS для управления ее проектированием
- Подраздел. Подсистема Метасистемы IMS.OSTIS для управления ее информационной безопасностью

- Подраздел. Характеристика технической реализации Метасистемы IMS.OSTIS
- Подраздел. Аналоги Метасистемы IMS.OSTIS
- Подраздел. Проекты, аналогичные Проекту IMS.OSTIS и выполняемые в настоящее время. Организации и команды, выполняющие эти проекты
- Подраздел. Характеристики, используемые для сравнительного анализа Метасистемы IMS.OSTIS
- Подраздел. Сравнительный анализ Метасистемы IMS.OSTIS
- Подраздел. Конкурентные преимущества и достоинства Метасистемы IMS.OSTIS
- Подраздел. Пример диалога пользователя с Метасистемой IMS.OSTIS, демонстрирующий возможности и достоинства этой Метасистемы
- Подраздел. Новизна и оригинальность решений, реализуемых в Метасистеме IMS.OSTIS
- Подраздел. Новизна и эффективность технологий, используемых при разработке Метасистемы IMS.OSTIS

раздел документации

= Множество всевозможных разделов, входящих в состав различных документации

=> примечание*:

[В базе знаний могут быть представлены документации различного вида – документации различных проектов, документации различных технических систем, документации теорий различных предметных областей]

подпроект Проекта OSTIS

=> пояснение*:

[Это множество элементами которого являются сам Проект OSTIS, все непосредственные (собственные) подпроекты Проекта OSTIS, все собственные подпроекты всех собственных проектов Проекта OSTIS и т.д.]

- ▷ подпроект Проекта OSTIS, направленный на разработку целостной интеллектуальной системы
 - ≡ Проект IMS.OSTIS
 - ≡ Проект IS.GraphTheory
 - ≡ Проект IMS.Geometry
- ▷ подпроект Проекта OSTIS по разработке подсистемы для соответствующей интеллектуальной системы
- ▷ подпроект Проекта OSTIS по разработке пользовательского интерфейса интеллектуальной системы /* для соответствующей целостной интеллектуальной системы */
- ▷ подпроект Проекта OSTIS по разработке подсистемы управления диалогом с пользователем соответствующей интеллектуальной системы

- ▷ подпроект Проекта OSTIS по разработке подсистемы автоматического проектирования интеллектуальной системы /* для соответствующей целостной интеллектуальной системы */
- ▷ подпроект Проекта OSTIS по разработке подсистемы управления проектированием соответствующей интеллектуальной системы

проект

=> пояснение*:

[Это целенаправленная деятельность, результатом которой является определенный ранее не существовавший продукт]

=> примечание*:

[описание (спецификация) каждого проекта включает в себя:

- указание продукта, являющегося результатом выполнения этого проекта;
- указание проектной задачи (проектной цели, технического задания) – требований, которым должен удовлетворять разрабатываемый продукт;
- срок выполнения проекта;
- команда исполнителей проекта (с указанием роли каждого исполнителя);
- план выполнения проекта (для неатомарного проекта это его декомпозиция на подпроекты, а для атомарного проекта – декомпозиция на проектные задания)]

продукт*

= быть продуктом соответствующей деятельности (в том числе и явно обозначенного, явно заявленного проекта)

▷ основной продукт*

= быть основным продуктом данной деятельности (в том числе проекта)

Продукты Проекта OSTIS

- ▷ Библиотека многократно используемых компонентов интеллектуальных систем, проектируемых по Технологии OSTIS
- ▷ Унифицированные семантические модели интеллектуальных систем и их компонентов, разработанные по Технологии OSTIS
- ▷ Интеллектуальные системы, разработанные по Технологии OSTIS
- ▷ Языки внутреннего представления знаний, разрабатываемых по Технологии OSTIS
- ▷ Языки внешнего представления знаний, разрабатываемых по Технологии OSTIS
- ▷ Инструментальные средства Технологии OSTIS
- ▷ Методы Технологии OSTIS

Библиотека многократно используемых компонентов интеллектуальных систем, проектируемых по Технологии OSTIS

= многократно используемый компонент интеллектуальных систем проектируемых по Технологии OSTIS

- ⊃ Библиотека sc-моделей многократно используемых компонентов баз знаний
- ⊃ Библиотека sc-моделей многократно используемых неатомарных и атомарных агентов обработки знаний
- ⊃ Библиотека sc-моделей многократно используемых компонентов интерфейсов интеллектуальных систем
- ⊃ Библиотека sc-моделей типовых подсистем интеллектуальных систем
- ⊃ Библиотека различных вариантов технических реализаций sc-памяти
- ⊃ Библиотека различных вариантов технических реализаций scr-машины
- ⊃ Библиотека платформенно ориентированных вариантов реализации агентов обработки знаний

Библиотека sc-моделей многократно используемых компонентов баз знаний

- ⊃ Библиотека sc-моделей онтологий
- ⊃ Библиотека описаний используемых языков
- ⊃ Библиотека различных базовых знаний
 - ⊃ Базовые знания о множествах и отношениях
 - ⊃ Базовые знания о графовых структурах
 - ⊃ Базовые знания о числах и числовых моделях
 - ⊃ Базовые знания о пространстве и пространственных формах
 - ⊃ Базовые знания о времени, динамических системах, ситуациях, событиях
 - ⊃ Базовые знания об информационных целях, задачах и способах их решения
 - ⊃ Базовые знания о целенаправленной деятельности различных субъектов

Унифицированные семантические модели интеллектуальных систем и их компонентов, разработанные по Технологии OSTIS

- = sc-модель интеллектуальной системы или ее компонента
- ⊃ sc-модель интеллектуальной системы
- ⊃ sc-модель базы знаний
- ⊃ sc-модель машины обработки знаний
- ⊃ sc-модель пользовательского интерфейса

Интеллектуальные системы, разработанные по Технологии OSTIS

- = Множество всевозможных интеллектуальных систем, разработанных по Технологии OSTIS
 - ⊃ главный элемент*:
Метасистема IMS.OSTIS
 - ⊃ IS.Geometry
 - ⊃ IS.GraphTheory
 - ⊃ IS.Conf.OSTIS
 - ⊃ IS.IT.BSUIR

Языки внутреннего представления знаний, разрабатываемых по Технологии OSTIS

- = Семейство всевозможных языков внутреннего представления знаний, разрабатываемых по Технологии OSTIS
- = язык представления знаний в виде

- унифицированных семантических сетей
- = язык представления определенного вида знаний с помощью sc-графов
- = язык, текстами которого являются sc-графы
- = sc-язык
 - ⊃ Логический sc-язык
 - ⊃ Теоретико-множественный sc-язык
 - ⊃ SC-язык информационных целей и задач
 - ⊃ Семейство sc-языков программирования
 - ⊃ Язык SCP
 - ⊃ SC-код
 - = Базовый язык sc-графов
 - = Базовый язык унифицированных семантических сетей

Языки внешнего представления знаний, разрабатываемых по Технологии OSTIS

- = Способы изображения sc-графов
- = язык внешнего представления sc-графов
 - ⊃ SCg-код
 - ⊃ SCs-код
 - ⊃ SCn-код

Инструментальные средства Технологии OSTIS

- = Инструментальные средства автоматизации проектирования интеллектуальных систем на основе Технологии OSTIS
 - => примечание*:
[Такие средства становятся подсистемами проектируемых интеллектуальных систем, т.к. проектирование считается перманентным процессом, осуществляемым одновременно с эксплуатацией интеллектуальной системы]
 - ⊃ Семейство инструментальных средств проектирования sc-моделей баз знаний /*верификация, редактирование, анализ качества*/
 - ⊃ Семейство инструментальных средств проектирования sc-моделей хранимых программ /* в том числе scr-программ */
 - ⊃ Семейство инструментальных средств проектирования машин обработки знаний
 - ⊃ Семейство инструментальных средств проектирования пользовательских интерфейсов

проектная задача*

- = быть проектной задачей соответствующего проекта*
- ⊃ основная проектная задача*
 - = быть основной проектной задачей заданного проекта*
- ⊃ горячая проектная задача*
 - = быть срочной проектной задачей заданного проекта*

Горячие проектные задачи Проекта OSTIS

- => пояснение*:
[Это множество всех проектных задач, решаемых в рамках всевозможных проектов Проекта OSTIS и являющихся в текущий момент горячими (срочными) задачами]
- => пояснение*:

[Это множество всех проектных задач, связанных с *Проектом OSTIS* отношением *горячая проектная задача**]

исполнитель*

- = *быть индивидуальным участником некоторой целенаправленной деятельности** /*в том числе некоторого проекта*/
- = *быть исполнителем**
- ▷ *основной исполнитель**
- ▷ *координатор**
- /*каждому проекту должен соответствовать единственный координатор*/

Исполнители Проекта OSTIS

= *Рабочая группа Проекта OSTIS*

= *исполнитель Проекта OSTIS*

=> *пояснение**:

[Это множество всех персон, связанных с *Проектом OSTIS* отношением *быть исполнителем**]

=> *процедура присоединения**:

[Для того, чтобы войти в число исполнителей *Проекта OSTIS*, необходимо:

- выбрать интересующий Вас подпроект *Проекта OSTIS* (лучше, если это будет атомарный подпроект, не разбиваемый на подпроекты);
- направить свои предложения координатору выбранного подпроекта]

▷ *Основные исполнители Проекта OSTIS*

партнер проекта*

= *быть партнером соответствующего проекта**

= *быть коллективным участником соответствующего проекта**

Партнеры проекта OSTIS

= *Коллективные участники Проекта OSTIS*

▷ *Основные партнеры Проекта OSTIS*

новость*

= *быть новостью в соответствующей динамической системе**

= *быть событием в заданной динамической системе**

Новости проекта OSTIS

=> *пояснение**:

[*Новости Проекта OSTIS* представляют собой хронологическую последовательность событий, определяющих развитие *Проекта OSTIS*. К такого рода событиям, в частности, относятся:

- корректировка системы подпроектов *Проекта OSTIS*;
- корректировка системы разделов *Документации проекта OSTIS*;
- завершение выполнения какого-либо подпроекта *Проекта OSTIS* или *проектной задачи*, входящей в план выполнения такого подпроекта (завершение разработки очередной версии какого-либо продукта *Проекта OSTIS*,

какого-либо раздела *Документации Проекта OSTIS*);

- корректировка текущего состава рабочей группы *Проекта OSTIS* или распределения работ (по всем подпроектам *Проекта OSTIS*);
- обновление *sc-языков* (добавление и удаление ключевых узлов, модификация их семантики);
- обновление внешних языков представления *sc-графов* (*SCg, SCs, SCn*);
- обновление правил построения внешних идентификаторов *sc-элементов*;
- замена основных и системных идентификаторов конкретных *sc-элементов* (в том числе ключевых узлов);
- обновление (добавление, удаление или реконфигурация) различных фрагментов *Базы знаний IMS.OSTIS* (в т.ч. – различных разделов *Документации Проекта OSTIS*). Наиболее значимые новости такого рода можно считать аналогом научно-технических статей, а сам *Проект IMS.OSTIS* условно можно считать редакцией электронного семантически структурированного научно-технического журнала, посвященного созданию и развитию *Технологии OSTIS*;
- обновление библиотек многократно используемых (типовых) компонентов интеллектуальных систем;
- обновление инструментальных средств разработки интеллектуальных систем;
- обновление рекомендуемых методов разработки интеллектуальных систем (различных классов интеллектуальных систем и различных их компонентов);
- обновление машины обработки знаний *Метасистемы IMS.OSTIS*;
- обновление пользовательского *Метасистемы IMS.OSTIS*;
- обновление технической реализации *Метасистемы IMS.OSTIS*;
- основные новости *Проекта Conf.OSTIS*, направленного на подготовку и проведение ежегодных конференций *OSTIS*.]

форум*

= *быть форумом**

=> *пояснение**:

[Это бинарное ориентированное отношение, каждая связка которого связывает некий узел семантической сети с форумом, направленным на обсуждение сущности, обозначаемой указанным узлом. Такой обсуждаемой сущностью, в принципе, может быть все, что угодно, и, в частности, фрагмент семантической сети (в том числе раздел некоторой документации), проект, проектное задание, какая-либо конкретная новость, какое-либо понятие]

Форумы Проекта OSTIS

=> *пояснение**:

[Это множество всевозможных форумов,

имеющих отношение к Проекту OSTIS. К таким форумам, в частности, относятся:

- форумы, обсуждающие различные проекты, входящие в состав Проекта OSTIS;
- форумы, обсуждающие различные разделы *Документации Проекта IMS.OSTIS* и, в том числе, *Документации IMS.OSTIS*;
- форумы, обсуждающие различные проектные задачи Проекта OSTIS;
- форумы, обсуждающие различные новости *Проекта OSTIS*.]

8. Машина обработки знаний интеллектуальной Метасистемы, предназначенной для поддержки проектирования интеллектуальных систем

Машина обработки знаний текущей версии *Метасистемы IMS.OSTIS* включает в себя:

- средства навигационного поиска в *sc*-памяти;
- средства интеграции вводимых в память *sc*-графов;
- средства верификации *sc*-модели хранимой базы знаний;
- средства редактирования *sc*-модели хранимой базы знаний.

9. Пользовательский интерфейс интеллектуальной Метасистемы, предназначенной для поддержки проектирования интеллектуальных систем

Пользовательский интерфейс текущей версии *Метасистемы IMS.OSTIS* включает в себя:

- описание синтаксиса и семантики всех внешних языков, используемых для изображения (визуализации) *sc*-графов. К числу таких языков относятся *SCg-код* (Semantic Code graphical – способ графической визуализации *sc*-графов), *SCs-код* (Semantic Code string – способ неформатированной текстовой визуализации *sc*-графов), *SCn-код* (Semantic Code natural – способ форматированной текстовой визуализации *sc*-графов);
- редактор *sc.g-текстов* (тестов *SCg*-кода) и средства трансляции их в *sc*-графы, хранимые в *sc*-памяти;
- редактор *sc.s-текстов* (тестов *SCs*-кода) и *sc.n-текстов* (тестов *SCn*-кода) и средства трансляции их в *sc*-графы, хранимые в *sc*-памяти;
- средства упрощенного задания типовых навигационно-поисковых запросов с помощью соответствующих командных элементов управления пользовательским интерфейсом;
- средства визуализации *sc*-графов, являющихся ответами на пользовательские навигационно-поисковые запросы;

- средства задания режима и стиля визуализации ответов на пользовательские запросы (язык визуализации, естественный язык, являющийся основой идентификации отображаемых *sc*-элементов);
- средства задания различного вида запросов к элементам управления пользовательским интерфейсом;
- средства задания команд редактирования хранимой в *sc*-памяти базы знаний.

10. Техническая реализация интеллектуальной метасистемы, предназначенной для проектирования интеллектуальных систем

Основой технической реализации интеллектуальной *Метасистемы IMS.OSTIS* является web-ориентированный вариант интерпретатора унифицированных логико-семантических моделей (*sc*-моделей) интеллектуальных систем. В состав данного интерпретатора входит (1) web-ориентированный вариант реализации (на языке C) *sc*-хранилища – памяти для хранения *sc*-графов и (2) реализация интерпретатора виртуальной машины обработки *sc*-графов, которая задает операционную семантику *Метасистемы IMS.OSTIS*.

Подчеркнем, что указанный web-ориентированный интерпретатор *sc*-моделей интеллектуальных систем может быть использован для технической реализации не только *Метасистемы IMS.OSTIS*, но и многих других интеллектуальных систем, разрабатываемых на основе предлагаемой технологии компонентного проектирования интеллектуальных систем.

11. Архитектура интеллектуальных систем, построенных на основе предлагаемой технологии проектирования интеллектуальных систем

Если рассматривать полностью укомплектованную интеллектуальную систему на самом верхнем уровне, то она представляет собой результат интеграции целого ряда интеллектуальных систем, специализирующихся на решении различного вида задач. К числу таких подсистем относятся:

- основная подсистема;
- пользовательский интерфейс;
- help-подсистема;
- подсистема управления диалогом;
- подсистема автоматизации интеллектуальной системы;
- подсистемы управления процессом обновления;
- подсистемы управления информационной безопасностью.

Архитектура каждой из перечисленных

интеллектуальных систем задается (1) архитектурой ее базы знаний и (2) архитектурой ее машины обработки знаний (решателя задач).

На верхнем уровне архитектура базы знаний представляет собой систему взаимосвязанных между собой первичных и вторичных предметных областей, а также онтологий и формальных теорий, соответствующих этим предметным областям. Т.е. на этом уровне структура базы знаний задается:

- семейством рассматриваемых первичных предметных областей;
- семейством используемых вторичных предметных областей;
- семейством конкретных онтологий, описывающих понятия всех введенных предметных областей (подчеркнем, что эти онтологии являются фрагментами вторичной предметной области всевозможных онтологий);
- семейством формальных теорий, описывающих свойства введенных предметных областей (заметим также, что такие формальные теории также являются фрагментами соответствующей вторичной предметной области всевозможных формальных теорий).

На более низком уровне структура базы знаний представляет собой систему фрагментов предметных областей, онтологий и формальных теорий (таких фрагментов, как определения, утверждения, семантические окрестности, сравнительные описания и т.п.).

На самом нижнем уровне база знаний интеллектуальной системы рассматривается как система взаимосвязанных элементарных фрагментов *sc-графов (sc-элементов)*. Таким образом, в основе этого уровня рассмотрения (уровня структуризации) базы знаний лежат следующие три компонента:

- *Предметная область sc-элементов*, оперирующая основными типами *sc-элементов* и основными отношениями, заданными на множестве *sc-элементов*. К указанным типам *sc-элементов* относятся и те, которые задаются синтаксически в помощью соответствующих элементов *Алфавита sc-элементов*;
- *Онтология Предметной области sc-элементов*, которая описывает все понятия, используемые в указанной предметной области;
- *Теория Предметной области sc-элементов*, которая описывает свойства и закономерности, имеющие место в указанной предметной области.

Указанные три компонента базы знаний есть не что иное, как базовый уровень представления (кодирования) унифицированных семантических сетей (*sc-графов*) и описания синтаксиса и семантики такого представления. Это то, что мы называем *SC-кодом*, его синтаксисом и семантикой.

Очевидно, что и *Онтология Предметной области sc-элементов*, и *Теория Предметной области sc-элементов*, а также онтологии многих

других предметных областей, используемых в базе знаний, должны быть включены в библиотеку многократно используемых компонентов баз знаний.

На верхнем уровне рассмотрения архитектуры машины обработки знаний интеллектуальной системы задается семейством совместных (интегрируемых) специализированных виртуальных машин обработки знаний. К числу таких машин, в частности, относятся:

- машина дедуктивного вывода;
- машина информационного поиска;
- машина интерпретации программ, принадлежащих языкам программирования высокого уровня;
- машина поиска ошибок в базе знаний;
- машина удаления информационного мусора;
- и т.д.

На более низком уровне рассмотрения архитектура машины обработки знаний каждой интеллектуальной системы задается:

- частью базы знаний, используемой для управления процессом решения задач (прежде всего – это процедурные знания высокого уровня);
- семейством агентов обработки знаний, реализующих различные операции логического вывода и осуществляющих интерпретацию процедурных знаний высокого уровня.

Часть базы знаний, используемая для управления процессом решения задач, включает в себя:

- Предметную область информационных целей и задач, а также соответствующую ей онтологию и формальную теорию;
- Предметную область поведенческих целей и задач, а также соответствующую ей онтологию и формальную теорию;
- Предметную область информационных программ высокого уровня и процессов их реализации, а также соответствующую этой предметной области онтологию и формальную теорию;
- Предметную область поведенческих программ и планов решения конкретных поведенческих задач, а также соответствующую этой предметной области онтологию и формальную теорию;
- Предметную область ситуаций и событий в *sc-памяти* (виртуальной памяти для хранения *sc-графов*), а также соответствующую этой предметной области онтологию и формальную теорию.

Если рассматривать машину обработки знаний еще на более низком уровне, то это будет машина, осуществляющая интерпретацию программ, описывающих поведение агентов обработки знаний на базовом языке программирования – **Язык SCP** (Semantic Code Programming). База знаний такой машины включает в себя предметную область *scp-программ* (программ базового языка программирования, ориентированного на обработку *sc-графов* – языка SCP), и *scp-процессов* (процессов

выполнения *scr-программ*), а также соответствующую этой предметной области онтологию и формальную теорию.

На следующем уровне детализации машины обработки знаний:

- уточняется техническая реализация памяти для хранения *sc*-графов (*sc*-памяти, *sc*-хранилища);
- уточняется техническая реализация агентов, осуществляющих интерпретацию *scr-программ*, хранимых в *sc*-памяти.

На рисунке 1 представлена система уровней детализации укомплектованных интеллектуальных систем, построенных на основе предлагаемой технологии.

Коллектив интегрированных интеллектуальных систем	
Обрабатываемая база знаний интеллектуальной системы	
Система предметных областей, онтологий и формальных теорий	
Система семантически целостных фрагментов базы знаний	
Система взаимосвязанных <i>sc</i> -элементов	
Коллектив интегрированных решателей задач	
Решатель задач	
Знания решателя задач используемые для управления процессом обработки знаний на основании используемой модели решения задач	Коллектив неатомарных агентов над обрабатываемой базой знаний Коллектив атомарных агентов над обрабатываемой базой
Хранимые <i>scr-программы</i> , описывающие поведение атомарных агентов над базой знаний Библиотека <i>scr-программ</i> , описывающих базовые преобразования обрабатываемых знаний	Неатомарный агент интерпретации хранимых <i>scr-программ</i> Коллектив атомарных агентов интерпретации хранимых <i>scr-программ</i>
Техническая реализация <i>sc</i> -памяти	Техническая реализация атомарных агентов интерпретации хранимых <i>scr-программ</i>

Рисунок 1. Уровни детализации интеллектуальной системы

12. Достоинства интеллектуальных систем, построенных на основе предлагаемой технологии проектирования интеллектуальных систем

К числу достоинств интеллектуальных систем, построенных на основе предлагаемой технологии можно отнести:

- (1) Многообразие видов знаний, хранящихся в базе знаний интеллектуальной системы;
- (2) Многообразие типов вопросов, на которые система может отвечать;
- (3) Многообразие видов решаемых задач и многообразие универсальных моделей решения

задач, используемых в самых различных предметных областях. То есть речь идет о предметно независимых моделях решения задач, с помощью которых можно решать задачи в любых (!) предметных областях. Подчеркнем при этом, что принципиальная возможность использования указанных моделей решения задач в любых предметных областях вовсе не гарантирует успешности такого использования;

(4) Способность решать не только предметные задачи, но и задачи управления обучением пользователя (вид метазадачи – сделать так, чтобы пользователь перешёл в состояние умения решать заданное многообразие задач, в том числе информационно-поисковых);

(5) Наличие встроенной интеллектуальной *help*-системы для конечных пользователей, обеспечивающей существенное повышение эффективности эксплуатации системы;

(6) Возможность использования терминологии различных естественных языков и различных слэнгов (в том числе и профессиональных). Возможность быстрого переключения диалога на разные терминологические режимы;

(7) Наличие развитых средств визуализации знаний, в том числе разных стилей визуализации фрагментов семантического пространства и удобных средств навигации по этому семантическому пространству;

(8) Наличие встроенной интеллектуальной подсистемы управления коллективной разработкой (совершенствованием), которая осуществляется разработчиками и конечными пользователями системы непосредственно в ходе её эксплуатации, это существенно продлевает жизненный цикл системы;

(9) Возможность легко расширять (наращивать) знания и умения системы руками разработчиков;

(10) Интегрируемость системы с другими смежными системами и системами, построенными на основе технологии OSTIS;

(11) Наличие средств самодиагностики, самоанализа и самосовершенствования.

13. Состав общей онтологии проектирования интеллектуальных систем

Создание технологии компонентного проектирования интеллектуальных систем невозможно без обеспечения совместимости многократно используемых компонентов интеллектуальных систем, накапливаемых в соответствующих библиотеках. Но для этого необходима унификация, согласованное представление формальных моделей не только компонентов интеллектуальных систем, но и интеллектуальных систем в целом. Такая

унификация, прежде всего, требует разработки согласованной общей онтологии проектирования интеллектуальных систем.

Говоря об интеллектуальных системах и о технологиях их проектирования, мы часто используем понятия, не утруждая себя уточнением их семантики и даже точным пониманием их смысла, мы часто говорим об одном и том же, не всегда чувствуя необходимость приходить к одинаковому пониманию даже, на первый взгляд, понятных вещей.

Казалось бы, очевидные вещи далеко не всегда бывают очевидными и, уж тем более, далеко не всегда одинаково трактуются разными специалистами. Но создавать эффективные массовые технологии проектирования интеллектуальных систем без привлечения достаточно широкого круга специалистов и, соответственно, без согласования точек зрения таких специалистов невозможно.

В состав общей онтологии проектирования интеллектуальных систем входят следующие частные онтологии:

- **Онтология общей теории интеллектуальных систем;**
- **Онтология библиотек многократно используемых компонентов интеллектуальных систем;**
- **Онтология средств проектирования интеллектуальных систем (средств автоматизации решения некоторых проектных задач);**
- **Онтология методов проектирования интеллектуальных систем;**
- **Онтология управления проектами, направленными на разработку интеллектуальных систем.**

Онтологию общей теории интеллектуальных систем, в свою очередь, можно разбить на:

- **Онтологию общей теории знаний;**
- **Онтологию машин обработки знаний и, в частности, интеллектуальных решателей задач;**
- **Онтологию пользовательских интерфейсов интеллектуальных систем и их компонентов.**

В качестве примера приведем перечень некоторых понятий, описываемых в *Онтологии общей теории интеллектуальных систем* и, в частности, в *Онтологии общей теории знаний*.

интеллектуальная система

= компьютерная система, основанная на знаниях

=> *примечание**:

[Каждая интеллектуальная система задается (1) своей базой знаний и (2) своей машиной обработки знаний.]

укомплектованная интеллектуальная система

⊂ интеллектуальная система

=> *примечание**:

[Укомплектованная интеллектуальная система

представляет собой результат интеграции целого ряда интеллектуальных систем (подсистем), каждая из которых ориентируется на решение определенного вида задач. К числу таких подсистем относятся:

- (1) основная подсистема укомплектованной интеллектуальной системы;
- (2) пользовательский интерфейс укомплектованной интеллектуальной системы, решающий задачи трансляции сообщений с внутреннего языка (*SC-кода*) на внешний и обратно, а также выполняющий пользовательские команды по управлению интерфейсом, а также по редактированию и вводу пользовательских сообщений;
- (3) help-подсистема, предназначенная для консультирования и обучения пользовательским навыкам эффективной эксплуатации основной интеллектуальной системы;
- (4) подсистема управления адаптивным диалогом с пользователями;
- (5) подсистема автоматизации обновления и совершенствования укомплектованной интеллектуальной системы;
- (6) подсистема управления процессом обновления и совершенствования укомплектованной интеллектуальной системы;
- (7) подсистема управления информационной безопасностью укомплектованной интеллектуальной системы.]

база знаний*

= быть базой знаний соответствующей интеллектуальной системы*

=> *пояснение**:

[Это бинарное ориентированное отношение, каждая связка которого связывает некоторую интеллектуальную систему с ее базой знаний. База знаний интеллектуальной системы представляет собой динамическую (нестационарную) знаковую конструкцию, которая содержит все (!) знания, необходимые указанной интеллектуальной системе для ее функционирования.]

база знаний

= второй домен отношения быть базой знаний*

=> *пояснение**:

[Это целостная (!) совокупность знаний, достаточных (!) для функционирования некоторой интеллектуальной системы. Здесь принципиальным является факт достаточности (полноты) этих знаний для того, чтобы указанная интеллектуальная система могла решать соответствующий класс задач.

Базу знаний можно трактовать как некоторое семантическое пространство, в котором функционирует указанная интеллектуальная система.]

=> *примечание**:

[Важнейшим свойством базы знаний является структуризация и систематизация входящих в нее знаний. Это предполагает переход от знаний

к метазнаниям путем явного выделения различных фрагментов, входящих в состав базы знаний, и явного представления различных связей, связывающих такие выделенные фрагменты базы знаний с другими фрагментами или с другими сущностями.]

машина обработки знаний

=> *пояснение**:

[Это многоагентная система над общей памятью, в которой хранится база знаний соответствующей интеллектуальной системы.]

знаковая конструкция

= *текст*

= *множество знаков, связанных между собой синтаксическими отношениями* /*в частности отношениями инцидентности*/

знак

= *фрагмент знаковой конструкции, обозначающий некоторую описываемую сущность*

▷ *первичный знак*

= *знак фрагмента внешней среды*

▷ *знак связи* /*между описываемыми сущностями*/

▷ *знак структуры* /*которая состоит из некоторого множества знаков описываемых сущностей и некоторого множества знаков связей между ними*/

▷ *знак понятия*

знак понятия

= *понятие*

▷ *концепт*

= *класс сущностей, не являющихся однотипными связками*

= *абсолютное понятие*

= *неотносительное понятие*

= *понятие, не являющееся отношением*

▷ *отношение*

= *класс однотипных связей*

= *относительное понятие*

▷ *ролевое отношение*

=> *примечание**:

[Каждому понятию ставится в соответствие:

(1) объем понятия – это то множество, которое обозначается этим понятием;

(2) содержание понятия – это семантическая окрестность этого понятия, описывающая его основные свойства;

(3) термин (имя) понятия – это строка символов, представляющая (изображающая) это понятие в различных сообщениях.]

семантическая сеть

=> *пояснение**:

[Это в общем случае нелинейная знаковая конструкция (нелинейный текст), атомарные (элементарные) фрагменты которой являются знаками описываемых сущностей либо знаками сущностей, используемых в таком описании.

При этом в указанной знаковой конструкции отсутствуют пары синонимичных знаков (знаков, обозначающих одну и ту же сущность). Это значит, что каждый знак, входящий в семантическую сеть входит в нее однократно (!). Из этого следует, что некоторые семантические связи между текстами (знаковыми конструкциями), которые представлены семантическими сетями, имеют достаточно простую теоретико-множественную интерпретацию, если каждую семантическую сеть трактовать как множество знаков описываемых сущностей и знаков связей между сущностями, – семантическое пересечение семантических сетей есть не что иное, как их теоретико-множественное пересечение, семантическое включение для семантических сетей трактуется как теоретико-множественное включение, а семантическая эквивалентность – как равенство соответствующих множеств. При этом следует отличать семантическую эквивалентность от логической эквивалентности (взаимного логического следования).]

знание

= *Множество всевозможных знаний*

=> *пояснение**:

[Это стационарная знаковая конструкция обладающая некоторой семантической целостностью. Можно говорить о предметной области, в которой множеством исследуемых объектов является Множество всевозможных знаний. Концептами такой предметной области являются различные классы (типы, виды) знаний, а отношениями указанной предметной области являются различные отношения, заданные на Множестве всевозможных знаний. Соответственно этому можно говорить об онтологии указанной предметной области]

вид знаний

= *тип знаний*

= *Семейство подмножеств Множества всевозможных знаний*

≡ *предметная область*

▷ *стационарная предметная область*
= *статическая система*

▷ *нестационарная предметная область*
= *динамическая система*

≡ *семантическая окрестность*

≡ *описание*

≡ *сравнение*

≡ *определение*

≡ *онтология*

▷ *теоретико-множественная онтология*

▷ *логическая онтология*

▷ *терминологическая онтология*

≡ *цель*

▷ *вопрос*

= *информационная цель*

= *цель, направленная на изменение состояния базы знаний*

- ▷ поведенческая цель
 - = цель, направленная на изменение внешней динамической системы (внешней среды)
 - ▷ цель отображения пользователю заданного знания
- ≡ задача
 - = формулировка задачи
 - = цель+контекст
 - = что требуется+что дано
- ≡ обобщенная формулировка задач, принадлежащих соответствующему классу однотипных задач
- ≡ программа
- ≡ предметная область
- ≡ фактографическое знание
- ≡ формальная теория
 - = компонент предметной области формальных теорий
- ≡ высказывание
 - = компонент формальной теории
- ≡ логическая формула
 - = компонент высказывания

предметная область

=> пояснение*:

[Понятие предметной области является важнейшим методологическим приемом, позволяющим выделить из всего многообразия исследуемого Мира только определенный класс исследуемых сущностей и только определенное семейство отношений, заданных на указанном классе. То есть осуществляется локализация, фокусирование внимания только (!) на этом, абстрагируясь от всего остального исследуемого Мира]

=> примечание*:

[Понятие предметной области рассматривается нами как обобщение понятия алгебраической системы. При этом семантическая структура базы знаний рассматривается нами как иерархическая система различных предметных областей]

<= разбиение*:

- {
 - **первичная предметная область**
 - ≡ Предметная область геометрических фигур и точек
 - = Геометрическая предметная область
 - = Предметная область геометрии
 - ≡ Предметная область множеств и отношений
 - = Предметная область теории множеств и отношений
 - ≡ Предметная область графовых структур и алгебраических отношений
 - ≡ Предметная область чисел, числовых отношений и числовых метаотношений
 - ≡ Предметная область текстов русского языка
 - ≡ Предметная область текстов всевозможных строчковых языков

- **вторичная предметная область**
 - = предметная область, состоящая из определенного вида надстроек над различными предметными областями /*не обязательно первичными*/
 - = предметная метаобласть
 - ≡ Предметная область формальных теорий и высказываний
 - ≡ Предметная область теоретико-множественных онтологий
 - ≡ Предметная область логических онтологий
 - ≡ Предметная область терминологических онтологий
 - ≡ Предметная область ситуаций и событий во всевозможных нестационарных предметных областях

}
<= разбиение*:

- {
 - **стационарная предметная область**
 - = статическая предметная область
 - => пояснение*:
[Это предметная область, в которой связи между сущностями, входящими в ее состав, не зависят от времени (не меняются во времени). При этом некоторые из указанных сущностей могут иметь конечное время "жизни" (конечное время существования).]
 - **нестационарная предметная область**
 - = динамическая предметная область
 - => пояснение*:
[Это предметная область, в которой некоторые связи между сущностями, входящими в ее состав, меняются со временем (то есть носят ситуационный, нестационарный характер)]

ситуация

=> пояснение*:

[Это квазистационарный фрагмент нестационарной предметной области, сохраняющий свою целостность в течении соответствующего отрезка времени]

событие

=> пояснение*:

[Это упорядоченная пара ситуаций, удовлетворяющая следующим требованиям:

- эти ситуации имеют несколько общих сущностей, являющихся экземплярами концептов соответствующей предметной области;
- момент начала "жизни" первой из указанных ситуаций предшествует моменту начала "жизни" второй ситуации.]

▷ событие иницирования ситуации

=> пояснение*:

[Это упорядоченная пара ситуаций, в которая удовлетворяет следующим дополнительным

требованиям:

- расстояние между моментами начала "жизни" первой и второй ситуации пренебрежительно мало и определяется соответствующими переходными процессами;
- первая ситуация не прекращает свою "жизнь" после начала второй ситуации.]

⊃ *событие трансформации ситуации*

=> *пояснение**:

[Это упорядоченная пара ситуаций, являющаяся событием, которое удовлетворяет следующему дополнительному требованию: момент конца "жизни" первой ситуации предшествует моменту начала "жизни" второй ситуации. При этом расстояние между указанными моментами пренебрежимо мало и определяется соответствующими переходными процессами]

система ситуаций и событий нестационарной предметной области

=> *пояснение**:

[Идея перехода от нестационарной предметной области к семейству ситуаций и событий этой области, заключается в том, чтобы перейти от нестационарного текста к такому стационарному тексту, который позволит с любой степенью детализации рассмотреть все процессы, происходящие в нестационарной предметной области]

Предметная область ситуаций и событий во всевозможных нестационарных предметных областях

=> *пояснение**:

[Это предметная область, обеспечивающая описание всевозможных нестационарных предметных областей как стационарных – как систем ситуаций (состояний) и событий, имеющих место в описываемых нестационарных областях]

онтология

= *система понятий соответствующей предметной области*

= *концептуальный каркас (скелет) описания некоторой предметной области*

= *концептуальная (семантическая) основа различных языков, обеспечивающих описание объектов исследования, принадлежащих заданной предметной области*

= *семантический интерфейс для интеграции знаний по заданной предметной области и для согласованного понимания различными субъектами этих знаний*

=> *эпиграф**:

[Находясь в плену синонимов, омонимов и субъективно трактуемых терминов невозможно добиться качественного взаимопонимания]

=> *эпиграф**:

[Основная проблема инженерии знаний – иллюзия понимания]

= *онтология соответствующей предметной области*

= *описание концептов и отношений заданной предметной области*

⊃ *теоретико-множественная онтология*

=> *пояснение**:

[Это описание теоретико-множественных связей между концептами и отношениями заданной предметной области (включение, разбиение, объединение, пересечение, разность множеств, область определения, домен, функция)]

⊃ *логическая онтология*

=> *пояснение**:

[Это описание системы определений концептов и отношений заданной предметной области]

⊃ *терминологическая онтология*

=> *пояснение**:

[Это описание системы основных и неосновных терминов (имен, внешних обозначений), соответствующих концептам и отношениям заданной предметной области, а также описание правил построения терминов для сущностей, являющихся элементами (экземплярами) указанных концептов и отношений]

теория

= *формальная теория соответствующей предметной области*

=> *пояснение**:

[Это система высказываний, описывающих свойства и закономерности некоторой заданной предметной области]

⊃ *теория стационарной предметной области*

⊃ *теория нестационарной предметной области*

программа

= *процедурное знание*

= *процедурное или декларативное описание*

способа решения любой или почти любой задачи из соответствующего класса однотипных задач

<= *разбиение**:

{

• *информационная программа*

= *обобщенная информация, достаточная для решения произвольной информационной задачи из некоторого класса однотипных информационных задач*

⊃ *процедурная информационная программа*
= *обобщенный план действий, обеспечивающий решение произвольной задачи из заданного класса информационных задач*

⊃ *алгоритм*

⊃ *логическая программа*

⊃ *функциональная программа*

=> *пояснение**:

[Это описание оптимизированного способа решения любой задачи из заданного класса задач. Каждой программе соответствует своя формальная теория, в рамках которой может быть доказана логико-семантическая корректность этой программы. Подчеркнем также, что все программы считаются фрагментами соответствующей вторичной предметной области всевозможных программ.]

- поведенческая программа

}

описание*

=> пояснение*:

[Это бинарное ориентированное отношение, каждая пара которого связывает узел семантической сети с фрагментом базы знаний, который является достаточно детальным (полным) описанием сущности, обозначаемой указанным узлом. Наиболее полным вариантом такого описания является документация (например, *Документация Проекта IMS.OSTIS*, *Документация Метасистемы IMS.OSTIS*)]

документация*

= быть документацией*

⊂ описание*

авторы*

= быть авторами*

=> пояснение*:

[Это бинарное ориентированное отношение, каждая пара которого связывает узел семантической сети, обозначающий продукт некоторой деятельности (в частности, продукт некоторого проекта), с узлом, обозначающим соответствующий авторский коллектив. Заметим, что частным видом указанного продукта является некоторое знание, входящее в состав базы знаний (например, раздел какой-либо документации)]

14. Состав Онтологии, которая соответствует предложенной технологии компонентного проектирования интеллектуальных систем

Подчеркнем, что разным технологиям проектирования интеллектуальных систем в общем случае могут соответствовать разные онтологии проектирования, определяемые разными вариантами формального уточнения ключевых понятий общей онтологии проектирования интеллектуальных систем.

В этом ничего удивительного нет. Например, фундаментальное понятие алгоритма имеет несколько вариантов его формального уточнения, а разным системам программирования соответствуют разные варианты формального уточнения понятия

программы.

Онтология предлагаемой технологии компонентного проектирования интеллектуальных систем Технологии OSTIS включает в себя:

- **Онтологию sc-моделей интеллектуальных систем;**
- Онтологию библиотеки многократно используемых компонентов sc-моделей интеллектуальных систем;
- Онтологию инструментальных средств проектирования sc-моделей интеллектуальных систем и их компонентов;
- Онтологию методов проектирования sc-моделей интеллектуальных систем и их компонентов;
- Онтологию технических средств интерпретации sc-моделей интеллектуальных систем;
- Онтологию управления проектами, направленными на разработку интеллектуальных систем на основе технологии OSTIS (в том числе и Метасистемы IMS.OSTIS).

В качестве примера приведем перечень некоторых понятий, описываемых в *Онтологии sc-моделей интеллектуальных систем*.

sc-граф

= sc-текст

= текст SC-кода

= Множество всевозможных текстов SC-кода

= SC-код

= Semantic Code

= Вид абстрактного унифицированного представления семантических сетей на основе базовой теоретико-множественной семантической интеграции всех элементов (узлов и коннекторов), входящих в состав этих семантических сетей

= унифицированная семантическая сеть с базовой теоретико-множественной интерпретацией ее элементов

⊂ знаковая конструкция

sc-элемент

= атомарный фрагмент sc-графа

<= разбиение*:

- sc-узел
- sc-ребро
- sc-дуга

}

<= разбиение*:

- sc-константа
- sc-переменная

}

<= разбиение*:

- sc-коннектор
- sc-связка
- sc-структура
- sc-концепт

- *sc-отношение*
- *первичный sc-узел*
- *sc-ссылка*

}

▷ *sc-дуга принадлежности*

sc-дуга принадлежности

<= *разбиение**:

```
{
  • позитивная sc-дуга принадлежности
  • негативная sc-дуга принадлежности
  • нечеткая sc-дуга принадлежности
}
```

<= *разбиение**:

```
{
  • стационарная sc-дуга принадлежности
  • нестационарная sc-дуга принадлежности
  • sc-дуга принадлежности неизвестной стационарности
}
```

sc-модель

=> *пояснение**:

[Это абстрактная модель интеллектуальной системы или какого-либо ее компонента, в основе которой лежит использование sc-графов.]

sc-модель предметной области

= *предметная область, представленная в виде sc-графа*

=> *примечание**:

[Понятия (концепты и отношения) предметной области в sc-модели предметной области задаются ее ключевыми sc-узлами, обозначающими эти понятия. Связь между указанными ключевыми sc-узлами и sc-узлом, обозначающим соответствующую sc-модель предметной области, задается с помощью соответствующих ролевых отношений.]

=> *примечание**:

[sc-модель предметной области практически всегда является бесконечным (!) sc-графом. Это значит, что указанная sc-модель не может быть полностью представлена в памяти интеллектуальной системы. Но практическую ценность имеет не вся такая sc-модель, а только некоторые ее фрагменты. Причем, в разные моменты времени могут быть востребованы разные фрагменты sc-модели предметной области.]

sc-язык

= *множество всевозможных фрагментов соответствующей sc-модели некоторой предметной области*

=> *примечание**:

[Каждому *sc-языку* взаимно однозначно соответствует своя sc-модель предметной области.]

=> *примечание**:

[SC-код, строго говоря, также следует считать одним из *sc-языков*. SC-коду соответствует

предметная область, оперирующая целым рядом понятий, обозначающих различные классы sc-элементов и различные отношения, заданные на множестве *sc-элементов*. К числу таких ключевых понятий, в частности, относятся: *sc-элемент* (как класс объектов исследования), *sc-узел*, *sc-коннектор*, *sc-дуга*, *sc-ребро*, *sc-константа*, *sc-переменная*, *sc-дуга принадлежности*, *позитивная sc-дуга принадлежности*, *негативная sc-дуга принадлежности*, *нечеткая sc-дуга принадлежности*, *стационарная sc-дуга принадлежности*, *нестационарная sc-дуга принадлежности*. Единственное отличие SC-кода от других sc-языков заключается в том, что некоторые ключевые понятия в нем задаются синтаксически с помощью алфавита sc-элементов.]

sc-модель онтологии

= *sc-онтология*

=> *пояснение**:

[Это онтология некоторой предметной области, представленной в SC-коде (т.е. в виде sc-графа).]

sc-модель нестационарной предметной области

=> *примечание**:

[Данная *sc-модель* носит принципиально нестационарный характер, поэтому практического значения эта модель не имеет, поскольку хранить ее в sc-памяти невозможно. Но можно хранить в sc-памяти (1) стационарную спецификацию этой модели и (2) семейство ситуаций и событий указанной динамической модели.]

sc-модель ситуаций и событий в нестационарной предметной области

=> *пояснение**:

[Это *sc-граф*, описывающий последовательность ситуаций и событий в некоторой нестационарной предметной области. Другими словами, это протокол изменения состояния указанной предметной области.]

SC-модель Предметной области ситуаций и событий во всевозможных нестационарных предметных областях

= *SC-модель Предметной области, описывающей динамику всевозможных нестационарных предметных областей*

= *SC-модель предметной области, описывающей протоколы изменения состояний всевозможных нестационарных предметных областей*

sc-память

=> *пояснение**:

[Это виртуальная память, предназначенная для хранения обрабатываемых sc-графов]

= *виртуальное хранилище sc-графов*

= *sc-модель памяти*

= *абстрактное sc-хранилище*

sc-модель ситуаций и событий в заданной sc-памяти

=> *примечание**:

[Не следует путать нестационарность описываемой предметной области и нестационарность самого sc-графа, описывающего эту предметную область и хранимого в некоторой sc-памяти. Во-первых, нестационарная предметная область может иметь описание стационарного вида (например, семейство ситуаций и событий). Во-вторых, нестационарность описания нестационарной предметной области чаще всего обусловлена (1) самим процессом обработки знаний, который в конечном счете сводится к генерации и удалению sc-элементов, хранимых в sc-памяти, а также к изменению (уточнению) их семантического типа, и (2) ограниченным объемом sc-памяти, что предполагает периодическое удаление всяческого рода информационного мусора.]

SC-модель Предметной области ситуаций и событий во всевозможных sc-хранилищах

- = SC-модель Предметной области, описывающей динамику sc-графов, хранимых в sc-памяти различных интеллектуальных систем
- = SC-модель Предметной области, в которой описываются протоколы изменения состояния sc-графов, хранимых в запоминающих средах различных интеллектуальных систем

SC-онтология Предметной области ситуаций и событий во всевозможных sc-хранилищах

<= онтология*:

SC-модель Предметной области ситуаций и событий во всевозможных sc-хранилищах

=> описываемые сущности*:

- {
- логически удаленный sc-элемент
- событие удаления заданного множества sc-элементов
- событие генерации заданного множества sc-элементов
- }

sc-модель базы знаний

⊂ база знаний

- = уточнение базы знаний, принятое в Технологии OSTIS
- = база знаний, представленная в виде унифицированной семантической сети с базовой теоретико-множественной семантической интерпретацией ее элементов
- = база знаний, представленная в виде sc-графа
- = база знаний, представленная в виде SC-коде
- = база знаний некоторой интеллектуальной системы, полностью представленная в виде sc-графа

sc-модель машины обработки знаний

=> пояснение*:

[Это полное описание поведения всех агентов, входящих в состав соответствующей машины обработки знаний, представленное в виде sc-графа с помощью ключевых узлов специального базового языка программирования SCP (Semantic Code Programming), ориентированного на обработку sc-графов, хранимых в структурно перестраиваемой (графодинамической) ассоциативной памяти, которую будем называть sc-памятью.]

scp-машина

- = Виртуальная машина, осуществляющая интерпретацию scp-программ, хранимых в sc-памяти
- = Базовая виртуальная машина, обеспечивающая интерпретацию sc-моделей любых (!) машин обработки знаний
- = sc-модель машины, обеспечивающей интерпретацию базового языка программирования, ориентированного на обработку sc-графов

sc-модель интеллектуальной системы

- = унифицированная логико-семантическая модель интеллектуальной системы, основанная на использовании sc-графов
- = полное описание логико-семантического уровня интеллектуальной системы, оформленное в виде sc-графа

ЗАКЛЮЧЕНИЕ

Таким образом, создание описанной в данной статье технологии позволит решить ряд существенных проблем, рассмотренных в данной работе, что, в свою очередь, радикально изменит возможности разработчика, проектирующего интеллектуальные системы и расширит спектр решаемых подобными системами задач.

Библиографический список

[Айзерман и др., 1988] Айзерман, М.А. Динамический подход к анализу структур, описываемых графами (основы графодинамики) / М. А. Айзерман, Л. А. Гусев, С. В. Петров, И. М. Смирнова, Л. А. Тененбаум // Исследования по теории структур. - М. : Наука, 1988. - С. 5-76.

[Гаврилова, 2008] Гаврилова, Т. А. Визуальные методы работы со знаниями: попытка обзора / Т. А. Гаврилова, Н. А. Гулякина // Искусственный интеллект и принятие решений, 2008, № 1, С. 15-21

[Гаврилова и др., 2000] Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2000.

[Глоба и др., 2012a] Глоба, Л. С. Подход к хранению баз нечетких знаний / Л. С. Глоба, М. Ю. Терновой, О. С. Штогрин // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012

[Глоба и др., 2012b] Глоба, Л. С. Модели и методы интеграции информационных и вычислительных ресурсов / Л. С. Глоба, Р.Л. Новогрудская // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012, с.447-452

[Голенков, 2011] Голенков, В.В., Гулякина, Н.А. Принципы построения массовой семантической технологии компонентного

- проектирования интеллектуальных систем . – В кн Междунар. научн.-техн. конф. . «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2011). Материалы конф. [Минск, 10-12 февр. 2011 г.]. – Минск: БГУИР, 2011, с. 21-59.
- [**Голенков, 2012**] Голенков, В.В., Гулякина Н.А. Принципы Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования . – В кн Междунар. научн.-техн. конф. . «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2012). Материалы конф. [Минск, 16-18 февр. 2012 г.]. – Минск: БГУИР, 2012, с. 23-52.
- [**Гулякина, 2012**] Гулякина, Н. А. Языки и технологии программирования, ориентированные на обработку семантических сетей/ Н. А. Гулякина, О. В. Пивоварчик, Д. А. Лазуркин// Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012. –С. 221-228
- [**Гуляева, 1989**] Гуляева, Д.М. Решение прикладных задач на расширенных семантических сетях. / Д.М. Гуляева // Математическое обеспечение ЭВМ и систем программирования. - М., 1989.
- [**Грибова, 2011**] Грибова, В. В. Облачная платформа для разработки и управления интеллектуальными системами / В. В. Грибова, А. С. Клещев, Д. А. Крылов, Ф. М. Москаленко, С. В. Смагин, В. А. Тимченко, М. Б. Тютюнник, Е. А. Шалфеева //Материалы международной научно-технической конференции OSTIS-2011, 5-14 стр. Минск БГУИР
- [**Евгеньев, 2008**] Евгеньев, Г.Б. Технология создания многоагентных прикладных систем / Г. Б. Евгеньев // Одиннадцатая национальная конференция по искусственному интеллекту : Труды конференции. Т.2. – М., 2008. – С. 306-312.
- [**Загорюлько и др., 2009**] Загорюлько Ю.А. Технология разработки порталов научных знаний // Программные продукты и системы. – 2009. – № 4. –С.25-29.
- [**Кандрашина и др., 1989**] Кандрашина, Е.Ю. Представление знаний о времени и пространстве в интеллектуальных системах / Е. Ю. Кандрашина, Л. В. Литвинцева, Д. А. Поспелов - М.: Наука, 1989.
- [**Карабеков и др., 2008**] Карабеков, Б.А. Система «Бинарная Модель Знаний» как инструмент для концептуального моделирования бизнес-процессов // Одиннадцатая национальная конференция по искусственному интеллекту : Труды конференции. Т.2. – М., 2008. – С. 282-291.
- [**Касьянов, 2003**] Касьянов, В.Н. Графы в программировании: обработка, визуализация и применение/ В. Н. Касьянов, В. А. Евстигнеев // ВNH–Санкт-Петербург, 2003.–1104 с.
- [**Клещев и др., 2001а**] Клещев А.С., Артемьева И.Л. Математические модели онтологий предметных областей. Часть 1. Существующие подходы к определению понятия "онтология" //НТИ.Серия 2 "Информационные процессы и системы", 2001, № 2
- [**Клещев и др., 2001б**] Клещев А.С. Артемьева И.Л. Математические модели онтологий предметных областей. Часть 2. Компоненты модели //НТИ.Серия 2 "Информационные процессы и системы", 2001, № 3
- [**Клещев и др., 2001с**] Клещев А.С. Артемьева И.Л. Математические модели онтологий предметных областей. Часть 3. Сравнение разных классов моделей онтологий //НТИ.Серия 2 "Информационные процессы и системы", 2001, № 4
- [**Кузнецов, 2009**] Кузнецов, О.П. Интеллектуализация поддержки управляющих решений и создание интеллектуальных систем / О.П. Кузнецов // Теория и методы разработки программного обеспечения систем управления. Спецвыпуск журнала «Проблемы управления» №3.1, 2009.
- [**Маклаев, 2012**] Маклаев, В.А. Соснин, П.И. Прецедентно-ориентированная база опыта проектной организации / В.А. Маклаев, П.И. Соснин, Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012
- [**Мальковский, 2012**] Мальковский, М.Г. Соловьев, С.Ю. Терминологические сети / М.Г. Мальковский, С.Ю. Соловьев, Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012, с. 77-82.
- [**Мартынов, 1977**] Мартынов, В. В. Универсальный семантический код / В. В. Мартынов. – Минск : Наука и техника, 1977.
- [**Мельчук, 1974**] Мельчук, И.А. Опыт теории лингвистических моделей «Смысл-Текст». Семантика, синтаксис/ И. А. Мельчук. – М. : Наука, 1974.
- [**Нариньяни, 1994**] Нариньяни, А.С. НЕ-факторы и инженерия знаний: от наивной формализации к естественной грамматике / А. С. Нариньяни //КИИ-94. Сборник трудов Национальной конференции с международным участием по ИИ. «Искусственный интеллект-94»; в 2-х т. – Т. 1. – Тверь : АИИ, 1994.- С. 9-18.
- [**Плесневич, 2008**] Плесневич, Г.С. Бинарные модели знаний / Г. С. Плесневич // Труды Международных научно-технических конференций «Интеллектуальные системы» (AIS'08) и «Интеллектуальные САПР» (CAD-2008). Научное издание в 4-х томах. – М : Физматлит, 2008, Т.2. – С. 424 – 135-146.
- [**Поспелов, 1986а**] Поспелов, Д.А. Представление знаний. Опыт системного анализа / Д. А. Поспелов. // Системные исследования. Методологические проблемы. Ежегодник. - М.: Наука, 1986. - с. 83-102.
- [**Поспелов, 1986б**] Поспелов, Д.А. Ситуационное управление. Теория и практика / Д. А. Поспелов. – М : Наука, 1986.
- [**Скореходько, 1989**] Скореходько, Э.Ф. Семантические сети и автоматическая обработка текста. / Э. Ф. Скореходько. – Киев: Наук. думка, 1983.
- [**Смирнов, 2008**] Смирнов, С.В. Прагматика онтологий: объектно-ориентированная модель знаний о предметной области // КИИ-2008. Труды конференции. Т. 3, С. 208-216.
- [**Смирнов, 2012**] Смирнов, С.В. Онтологическое моделирование в ситуационном управлении/ С.В.Смирнов //Онтология проектирования, 2012, №2. – С16-24
- [**Соснин, 2011**] Соснин, П.И. Вопросно-ответное программирование человеко-компьютерной деятельности / П.И. Соснин – Ульяновск: УлГТУ, 2010. – 240 с.
- [**Тыгу, 1989**] Тыгу, Э.Х. Интеграция знаний / Э. Х. Тыгу // Изв. АН СССР. Техн. кибернет. - 1989. - № 5. - с. 3-13.
- [**Хорошевский, 2008**] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В. Ф. Хорошевский. // Искусственный интеллект и принятие решений, 2008, №1. – С.80-97.
- [**Шенк, 1980**] Шенк, Р. Обработка концептуальной информации / Р. Шенк. – Москва: Энергия, 1980.
- [**OSTIS, 2012**] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2012. - Режим доступа: <http://ostis.net>. – Дата доступа: 18.11.2012

OPEN PROJECT, AIMED AT CREATION OF TECHNOLOGY OF COMPONENT DESIGN OF INTELLECTUAL SYSTEMS

Golenkov V.V., Guliakina N.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

golen@bsuir.by

guliakina@bsuir.by

This article is devoted to the open project, aimed at creation of technology of component design of intellectual systems. Also this article considers Metasystem IMS.OSTIS, designed for complex support of intellectual systems design.

INTRODUCTION

Analysis of modern information technologies can show that together with very impressive achievements they have a number of serious shortcomings:

- There is not common unified solution for computer systems semantic compatibility problem, which leads to a high laboriousness of complex integrated computer systems creation.
- Level of computer systems platform (on which these systems are implemented) dependency is very high, which leads to a high laboriousness of computer systems transition from one platform to another.
- There is not well elaborated technique of constructive using of the experience of completed development of computer systems, which leads to high level of duplication of different components developed for these systems.
- There is not well elaborated and unified technique of computer systems different architecture levels improvement during the maintenance process, which leads to a high laboriousness of such a improvement.
- Modern computer technologies are not oriented on a wide circle of computer systems developers.

MAIN PART

One of the directions to solve modern information technologies problems shown above is an expansion of artificial intelligence methods and means using in traditional computer systems.

Transformation of traditional computer systems into intelligent is one of the most important problems of computer systems evolution, directed on their efficiency and competitiveness improvement. Such a transformation includes:

- Unified systematization and structuring of all information processed in the system, which means a transition from subjectively structured data to a knowledge base, i.e. to a fundamentally new quality of presentation and structuring of processing information;

- Transition to a programming, oriented on knowledge bases (semantically structured data) processing, what assumes expanding the development of fundamentally new models of information processing and programming languages.

Among the key provisions underlying proposed technology of intelligent systems design, the following provisions apply:

(1) Component design method is used, which is a factor of maturity of any technology and is based on permanently expandable libraries of reusable components (standard solutions).

(2) Formal models of designed intelligent systems are based on the unified semantic networks, which creates the necessary conditions for the semantic interoperability of intelligent systems and their components (i.e., to meet the challenges of integration).

(3) To reduce the complexity of the design and modification (updating) created and permanently improved intelligent systems, the maximum possible independence of the intelligent system knowledge base update process from the intelligent system knowledge processing methods update process and the intelligent system technical implementation means update process (including the transition to the new platform).

(4) The proposed technology of component design of intelligent systems is made as intelligent metasystem, which is built on the same proposed technology and accumulates all accumulated to this time models, means (including the library of standard components) and methods that are the part of the proposed technology.

(5) The permanent development of technology of component design of intelligent systems is performed within the scope of opensource-project.

(6) In the proposed design technology of intelligent systems special attention should be paid:

- To the technology of upgrade (improvement) of intelligent systems during their operation process;
- A metatechnology of updating (improvement) of the technology of component design of intelligent systems (ie, metasystem intended to support the design of intelligent systems).

CONCLUSION

Thus, the creation of the technology, which is described in this article, will solve a number of important issues discussed in this work, which, in turn, will radically change the capabilities of developer, designing intelligent systems and expand the range of such systems functions.

СЕКЦИЯ 2.

**БАЗОВЫЕ СЕМАНТИЧЕСКИЕ МОДЕЛИ ИНТЕЛЛЕКТУАЛЬНЫХ
СИСТЕМ И ИХ РЕАЛИЗАЦИЯ**

SECTION 2.

**BASIC SEMANTIC MODELS OF INTELLIGENT SYSTEMS AND THEIR
IMPLEMENTATIONS**



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

МЕТОД ПРЕОБРАЗОВАНИЯ КЛАССОВ СЕМАНТИЧЕСКИХ СЕТЕЙ

Тимченко В.А.

*Федеральное государственное бюджетное учреждение науки Институт автоматизации и процессов управления Дальневосточного отделения Российской академии наук,
г. Владивосток, Россия*

rakot2k@mail.ru

В работе представлен метод преобразования классов семантических сетей, определяющий, как по заданной спецификации преобразования на основе исходного текстового или структурного представления семантической информации получить целевое текстовое или структурное представление этой информации, удовлетворяющее этой спецификации. Данный метод можно представить как совокупность трех базовых методов преобразования. Спецификации преобразований описываются в терминах моделей классов семантических сетей и их преобразований, являющихся инвариантными по отношению к технологическим пространствам.

Ключевые слова: методы преобразования семантических сетей, представление информации семантическими сетями, семантические сети, структурные проекции.

ВВЕДЕНИЕ

В настоящее время семантические сети успешно используются в качестве наглядной и универсальной структуры для представления информации (преимущественно декларативной) общего вида в разных предметных областях. Разные виды информации, независимо от конкретного синтаксиса языка их представления, в абстрактном синтаксисе представляют собой в общем случае (мульти-) графы, возможно, типизированные [Agrawal A. et al., 2005] [Ehrig K. et al., 2005]. Поэтому многие задачи преобразования семантической информации, возникающие в различных областях профессиональной деятельности, технологических пространствах [Kurtev, 2002] [Bezivin, 2005], а также на стыке разных дисциплин и технологических пространств, можно сформулировать в терминах преобразования семантических сетей. К таким задачам среди прочих можно отнести: задачу отображения онтологий, преобразования баз знаний, преобразование программ, преобразование моделей в модельно ориентированном подходе к инженерии информационных систем. Задача преобразования семантических сетей в общем случае может быть сформулирована следующим образом: по исходной семантической сети и спецификации преобразования сформировать новую семантическую сеть, удовлетворяющую этой спецификации.

Вместе с тем, в разных технологических пространствах исторически независимо сформировались свои эффективные модели и основанные на них методы преобразования информации; разработаны программные системы их поддержки. Однако, как показывает практика и опыт использования, методы и формализмы, применяемые в одном технологическом пространстве, зачастую по многим причинам напрямую не могут быть “спроецированы” на другое [Czarnecki, 2006].

Основным недостатком существующих методов преобразования разных видов семантической информации, таким образом, является их ориентированность на некоторое технологическое пространство или один вид преобразований – эндогенных или экзогенных, либо вовсе на конкретную задачу или класс задач; а также то, что они рассчитаны на круг специалистов, обладающих необходимыми знаниями в конкретной области.

В данной работе представлен метод преобразования классов семантических сетей, основанный на модели классов семантических сетей и их преобразований, которые являются инвариантными по отношению к технологическим пространствам [Тимченко, 2012]. Данный метод можно представить как совокупность трех базовых методов преобразования: “текст – семантическая сеть”, “семантическая сеть – текст” и “семантическая сеть – семантическая сеть”.

Причина, по которой должна обеспечиваться

возможность выполнять преобразования вида “текст – семантическая сеть” и “семантическая сеть – текст”, состоит в том, что во многих предметных областях текстовое представление информации (представление в конкретном синтаксисе) зачастую более привычно и понятно человеку, нежели структурное, в то время как последнее (представление в абстрактном синтаксисе), как правило, существенно удобнее для обработки программными системами. Таким образом, исходная информация может иметь как текстовое, так и структурное представление (в виде семантической сети). Целевую информацию также может потребоваться представить как в текстовом, так и в структурном виде.

Еще один аргумент состоит в том, что часто бывает быстрее и проще сформировать представление информации в виде сети понятий автоматически по текстовому представлению, чем вручную, с помощью средства редактирования породить такую сеть под управлением описания класса сетей понятий для этой информации. Примером является формирование программы на языке программирования в виде дерева абстрактного синтаксиса по описанию грамматики этого языка.

Спецификацией для метода преобразования “семантическая сеть – семантическая сеть” является описание структурной проекции на языке описания структурных проекций. Спецификацией для методов преобразования “текст – семантическая сеть” и “семантическая сеть – текст” является описание синтаксических ограничений (в терминах модели синтаксических ограничений) [Тимченко, 2012].

1. Метод преобразования “текст – семантическая сеть”

Существует множество разнообразных алгоритмов для анализа текста КС-языков разных классов. Наибольший интерес с точки зрения универсальности представляют общие методы анализа языков, не ограниченных требованиями к наличию тех или иных свойств у грамматики, их описывающей (LL(k), LR(k), наличие определенного порядка (грамматики предшествования), линейность и т.д.), то есть языков, являющихся контекстно-свободными в самом широком смысле [Хомский, 1966]. Такие языки описываются КС-грамматиками, правила которых имеют вид $A \rightarrow \alpha$, где A – понятие языка, или нетерминал, а α – строка символов из объединенного словаря терминалов и нетерминалов, либо пустая строка. В рамках данной работы решается задача анализа текста, который может быть описан на произвольном КС-языке.

Для решения задачи анализа выбран алгоритм Эрли, который подходит для осуществления разбора текста на КС-языке, задаваемом КС-грамматикой без ограничений. Алгоритм Эрли считается одним

из самых эффективных универсальным методом синтаксического анализа текстов, записанных на КС-языках. И хотя известен алгоритм Томиты и его новейшие модификации, которые претендуют на еще лучшие показатели эффективности КС анализа, отмечается, что реализация этих алгоритмов в десять, а то и более, раз сложнее, чем реализация алгоритма Эрли. Описание алгоритма Эрли можно найти, например, в [Ахо, 1978].

Вход:

- текстовое представление информации (*Text*);
- описание класса семантических сетей информации (S).

Выход:

- представление информации в виде семантической сети (S_{inst}).

Описание:

Этап 1. Формирование КС-грамматики по описанию синтаксических ограничений и описанию класса сетей понятий (S_{sr} и S_{nc} из S). Данный этап включает в себя:

– преобразование в соответствии с семантикой метасимволов синтаксических ограничений, содержащих факультативные элементы, а также ограничений, представляющих альтернативы и перечисления, в множество правил грамматики вида $A_i \rightarrow \alpha_i$, где A_i – нетерминальное понятие из описания класса сетей понятий, а α_i – последовательность, состоящая, в общем случае, из нетерминальных и терминальных понятий класса сетей понятий, а также элементов конкретного синтаксиса;

– формирование таблицы элементов конкретного синтаксиса, а также таблицы, содержащей ограничения на вид лексем.

Этап 2. Анализ текстового представления (*Text*) по сформированной КС-грамматике и построение последовательности списков ситуаций Эрли. На данном этапе используется алгоритм Эрли [Ахо, 1978].

Этап 3. Формирование семантической сети (S_{inst}), соответствующей описанию класса сетей понятий (S_{nc}), на основе полученной последовательности списков ситуаций и КС-грамматики.

Основная идея этапа состоит в том, чтобы, двигаясь от вершины сети (начального понятия) рекурсивно достраивать сеть понятий в соответствии с правилами грамматики и построенной последовательностью списков ситуаций Эрли.

Данный этап представлен рекурсивной функцией (*BuildNetwork*), входными параметрами которой являются: указатель на ситуацию Эрли, обрабатываемую на очередном шаге рекурсии (*state*); номер списка ситуации (*number*); указатель на вершину в формируемой S_{inst} , в которой находится вычислительный процесс при

выполнении очередного шага рекурсии (*concept*). Результат выполнения функции (*result*) содержит указатель на ситуацию Эрли, которую нужно обработать для корректной обработки перечислимых правил, а также – номер списка ситуаций, в котором эта ситуация находится, либо *нусто* – если правило, соответствующее текущей обрабатываемой ситуации, не является перечислимым (т.е. вида $\{;\dots\}$). При первом вызове аргументами функции *BuildNetwork* являются:

– указатель на ситуацию, являющуюся допускающей для входной строки символов. Допускающей является ситуация, присутствующая в списке ситуаций (из последовательности списков ситуаций) с наибольшим номером и содержащая правило грамматики, в котором нетерминал в левой части есть начальный нетерминал грамматики, точка в правой части – крайняя справа, и номер ситуации есть 0;

– длина входной строки КС-текста (*Text*) в лексемах (в качестве номера списка ситуации);

– начальной вершиной формируемого S_{inst} является левая часть правила, содержащегося в допускающей ситуации.

Шаг рекурсии состоит в следующем:

1. Из текущей ситуации Эрли (переданной в параметре вызова *BuildNetwork*) *state* берется правило грамматики *rule*.

2. В цикле, пока не достигнут первый его элемент из правой части ($k \geq 1$), выполняются следующие действия:

2.1. Берется k -ый элемент из правой части правила *rule* – *grammarElem* (перед циклом k равен номеру позиции последнего элемента правой части правила *rule*);

2.2. если *grammarElem* является терминалом, то:

2.2.1. создать терминальное понятие в S_{inst} и сделать его потомком *concept* – понятия S_{inst} , переданного в параметре вызова *BuildNetwork*: взять из *таблицы терминалов* тот, который был считан при формировании данного списка ситуаций, и добавить его в качестве понятия-потомка. При этом если для данного терминала уже было создано понятие в S_{inst} , то к нему создается только отношение от *concept*, иначе – создается новое понятие в S_{inst} и отношение к нему от *concept*;

2.2.2. перейти к предыдущему элементу из правой части правила ($k = k - 1$), а также к предыдущему списку ситуаций Эрли;

2.3. если *grammarElem* является нетерминалом, то:

2.3.1. создать нетерминальное понятие в S_{inst} (*childConcept*), присвоить ему в качестве значения значение нетерминала грамматики *grammarElem* и сделать его потомком понятия *concept*;

2.3.2. получить подходящую ситуацию Эрли для следующего шага рекурсии (*nextState*) по текущему номеру списка ситуаций Эрли и рассматриваемому элементу правила грамматики *grammarElem*;

2.3.3. если текущее правило является перечислимым, то необходимо сформировать *result* по текущему номеру списка ситуаций Эрли и текущей ситуации Эрли (*state*), чтобы следующий вызов *BuildNetwork* относился к этому же нетерминалу *grammarElem*, но соответствовал предыдущей итерации для перечислимого правила;

2.3.4. рекурсивно вызывается функция *BuildNetwork* с аргументами *nextState*, текущий номер списка ситуаций Эрли и *childConcept*. Далее для обработки перечислимого правила в цикле, пока результат вызова *BuildNetwork* (обозначим его *res*) не *нусто*, данная функция вызывается рекурсивно с аргументами: значение из *res*, содержащее указатель на ситуацию Эрли, значение из *res*, содержащее номер списка ситуаций, в котором эта ситуация находится, и *childConcept*;

2.3.5. перейти к предыдущему элементу из правой части правила ($k = k - 1$), а текущим номером списка ситуаций Эрли становится номер списка, в котором находится родительская ситуация ситуации *nextState*.

3. Выполняется выход из *BuildNetwork* с возвратом результата *result*.

2. Метод преобразования “семантическая сеть – текст”

Задача отображения некоторой информации, представленной в структурном виде, в текстовое представление является задачей синтеза текста. В рамках данной работы решается задача синтеза текста, который также может быть описан на произвольном КС-языке. Вариантом решения задачи синтеза является также недоопределенное текстовое представление, в котором не все слова (лексемы) являются элементами конкретного синтаксиса или значениями терминальных понятий. Некоторые понятия из семантической сети, представляющей описание класса сетей понятий, могут быть недоопределены в сети понятий, являющейся ее экземпляром. В этом случае в результирующем тексте должны быть синтезированы имена этих недоопределенных конструкций.

Вход:

- представление информации в виде семантической сети (S_{inst});
- описание класса семантических сетей информации (S).

Выход:

- текстовое представление информации (*Text*).

Описание:

Этап 1. Формирование КС-грамматики по описанию синтаксических ограничений и описанию класса сетей понятий (S_{sr} и S_{nc} из S). Данный этап

полностью совпадает с аналогичным этапом метода преобразования “текст – семантическая сеть”, который описан в предыдущем разделе.

Этап 2. Рекурсивный обход S_{inst} в глубину с параллельным просмотром правил сформированной КС-грамматики и синтезом фрагментов результирующего текста. Данный этап представлен рекурсивной функцией (*SynthesiseText*), входными параметрами которой являются: указатель на элемент (нетерминал) сформированной КС-грамматики (*grammarElem*); указатель на вершину в S_{inst} (*concept*). При первом вызове аргументами функции *SynthesiseText* являются аксиома грамматики и корневая вершина S_{inst} . Шаг рекурсии состоит в следующем:

1. Если *concept* является терминальным понятием, то к *Text* добавляется строковое представление значения понятия (при этом, если тип значения понятия *concept* есть “строковое”, то перед и после строкового представления значения ставится символ-ограничитель строковых констант, заданный в описании синтаксических ограничений).

2. Если *concept* является нетерминальным понятием, то:

2.1. получить правило грамматики (*rule*), у которого в левой части стоит *grammarElem* (если таких правил нет, то синтез текста не может быть выполнен);

2.2. если таких правил больше одного, то это одно из альтернативных правил. В этом случае выбирается нужная альтернатива (по имени метапонятия для понятия-потомка *concept*) и рекурсивно вызывается функция *SynthesiseText* с аргументами: правая часть альтернативного правила, понятие-потомок понятия *concept*;

2.3. если это не альтернативное правило, то в цикле по элементам правой части правила *rule* (одновременно выполняется проход по потомкам понятия *concept*):

2.3.1. элемент конкретного синтаксиса добавляется к *Text*;

2.3.2. если в S_{inst} нет понятий, соответствующих текущему элементу грамматики, то синтез текста не может быть выполнен;

2.3.3. если текущий потомок понятия *concept* (*child*) соответствует текущему элементу грамматики (*grElem*) (т.е. их типы совпадают, и имя метапонятия *child* совпадает с *grElem*), то рекурсивно вызывается функция *SynthesiseText* с аргументами: *grElem*, *child*; в противном случае синтез текста не может быть выполнен;

2.3.4. если *rule* является перечислимым правилом, то в цикле перебираются только потомки понятия *concept*, а текущий элемент грамматики *grElem* остается неизменным; к *Text* при этом добавляется символ-разделитель (заданный в описании синтаксических ограничений), которым в

тексте разделены повторяющиеся элементы; выполняется п. 2.3.3.

3. Метод преобразования “семантическая сеть – семантическая сеть”

Данный метод основан на операционной семантике языка описания структурных проекций, разработанного для формирования пользователем множества правил в терминах модели описания структурных проекций [Тимченко, 2012].

Вход:

- представление исходной информации в виде семантической сети (SS_{inst});
- описание класса сетей понятий целевой информации (TS_{nc});
- описание структурной проекции (μ).

Выход:

- представление целевой информации в виде семантической сети (TS_{inst}).

Описание:

Этап 1. Кэширование описания структурной проекции (μ) в хэш-структуре (*Mapping*) с целью ускорения доступа к правилам структурной проекции. При этом левые части правил проекции становятся ключами в хэш-структуре. Правая часть каждого правила проекции сохраняется в специальном абстрактном типе данных (АТД), после чего для доступа к нужным значениям правой части правила используются операции этого АТД. Экземпляры АТД сопоставляются ключам в хэш-структуре.

Этап 2. Выполняется рекурсивный обход исходной семантической сети (SS_{inst}) в глубину, на каждом шаге которого:

для вершины-прототипа текущей вершины из описания проекции (*Mapping*) выбирается применимое правило (если такое есть). В соответствии с описанием правой части выбранного правила в целевой семантической сети (TS_{inst}) достраивается подсеть.

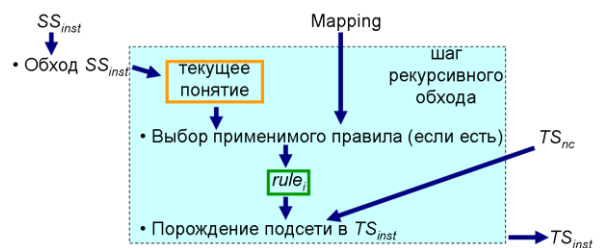


Рисунок 1 – Общая схема метода преобразования “семантическая сеть – семантическая сеть”

При этом создается новая подсеть, являющаяся экземпляром некоторой подсети из описания класса сетей понятий целевой информации (TS_{nc}), и связывается с уже сформированной на данном этапе (если такая есть) семантической сетью. После применения очередного правила формируемая целевая сеть всегда соответствует своей

метаинформации (рис.1). Порождение подсети TS_{inst} определяется через порождение множеств понятий и отношений, в нее входящих, по их прототипам из TS_{nc} .

Понятие в TS_{inst} порождается следующим образом: создается нетерминальное или терминальное понятие соответствующего типа (эта информация берется на основе информации о понятии-прототипе), а в качестве значения понятию может назначаться (в соответствии с правой частью применяемого правила) (ниже приведены альтернативные варианты):

- имя/значение понятия-прототипа;
- имя/значение понятия из SS_{inst} или его терминального понятия-потомка;
- значение из хеш-таблицы по заданному псевдониму (которое было туда предварительно помещено);
- явно заданное в правой части правила значение, либо это значение вычисляется по заданному в правиле выражению (в которое может входить произвольное количество переменных, инициализируемых значениями из SS_{inst} , либо из хеш-таблицы).

Направленное отношение между двумя понятиями в TS_{inst} порождается следующим образом:

- начальное и конечное понятия берутся из хеш-таблицы по заданному псевдониму, либо ищутся в уже сформированной на данном этапе TS_{inst} (в этом случае в правиле явно указано имя/значение понятия (или его понятия-прототипа, по которому оно порождено));
- отношение добавляется к множеству отношений, выходящих из понятия-начала, либо в соответствие с заданным порядковым номером, либо последним, если порядковый номер в правиле не задан.

ЗАКЛЮЧЕНИЕ

Разработаны методы преобразований “текст – семантическая сеть”, “семантическая сеть – текст” и “семантическая сеть – семантическая сеть”, определяющие, как по соответствующей спецификации преобразования на основе входной информации получить выходную, удовлетворяющую этой спецификации. Метод преобразования “текст – семантическая сеть” позволяет анализировать текст на языке, задаваемом КС-грамматикой без ограничений, в то время как большинство широко распространенных средств реализуют метод разбора, требующий, чтобы входная грамматика удовлетворяла определенным свойствам.

Следует отметить, что в зависимости от решаемой задачи преобразования, эти методы можно комбинировать между собой в разных

сочетаниях. Всего допустимо три комбинации, в зависимости от представления входной информации (текстовое или в виде семантической сети) и требуемого представления выходной информации (текстовое или в виде семантической сети). Возможность варьировать порядок применения методов позволяет существенно расширить спектр прикладного применения результатов работы.

Работа выполнена при финансовой поддержке РФФИ, проект 12-07-00179-а и ДВО РАН по Программе ОНИТ, проект 12-И-ОНИТ-04.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Agrawal et al., 2005] Agrawal, A. Reusable Idioms and Patterns in Graph Transformation Languages / A. Agrawal, A. Vizhanyo, Z. Kalmar, F. Shi, A. Narayanan, G. Karsai // Journal Electronic Notes in Theoretical Computer Science (ENTCS). – Vol. 127. – Issue 1. – 2005. – Pp. 181 – 192.
- [Ehrig et al., 2005] Ehrig, K. Model Transformation by Graph Transformation: A Comparative Study / K. Ehrig, E. Guerra, J. de Lara, L. Lengyel, T. Levendovszky, U. Prange, G. Taentzer, D. Varro, S. Varro-Gyapay // Proceedings of Model Transformations in Practice Workshop, MoDELS Conference. – Jamaica. – 2005. [Electronic resource]. URL: <http://www.inf.mit.bme.hu/FTSRG/Publications/varro/2005/mtip05.pdf> (дата обращения: 26.11.2012).
- [Kurtev, 2002] Kurtev, I. Technological Spaces: An Initial Appraisal / I. Kurtev, J. Bezivin, M. Aksit // Int. Federated Conf. (DOA, ODBASE, CoopIS), Industrial track, Irvine. – 2002.
- [Bezivin, 2005] Bezivin, J. Model-based Technology Integration with the Technical Space Concept / J. Bezivin, I. Kurtev // In Proceedings of the Metainformatics Symposium, Springer-Verlag. – 2005.
- [Czarnecki, 2006] Czarnecki, K. Feature-based survey of model transformation approaches / K. Czarnecki, S. Helsen // IBM Systems Journal, special issue on Model-Driven Software Development. 45(3). – 2006. – Pp. 621 – 645.
- [Тимченко, 2012] Тимченко, В. А. Модель классов семантических сетей и их преобразований / В. А. Тимченко // Материалы международной научно-технической конференции OSTIS-2012. – Минск БГУИР. – 2012. – С. 63 – 70.
- [Хомский, 1966] Хомский, Н. Формальные свойства грамматик / Н. Хомский // Кибернетический сборник, новая серия, вып. 2, изд-во “Мир”, Москва. – 1966. – С. 121 – 230.
- [Ахо, 1978] Ахо, А. Теория синтаксического анализа, перевода и компиляции / А. Ахо, Дж. Ульман // Москва: Мир. 1978. В двух томах.

THE METHOD OF SEMANTIC NETWORK CLASSES TRANSFORMATION

Timchenko V.A.

*Federal State Budget Institution of Science
Institute of Automation and Control Processes Far
Eastern Branch of the Russian Academy of
Sciences, Vladivostok, Russia*

rakot2k@mail.ru

The paper presents a method for semantic network classes transformation. This method defines how to generate the target textual or structural representation of semantic information from source textual or structural representation of semantic information in concordance with transformation specification. This method can be represented as a combination of three basic methods of transformation. Specifications of transformations are described in terms of the models of semantic network

classes and their transformations. Both models are invariant to the different technological spaces.

INTRODUCTION

Currently, semantic networks have been successfully used as a visual and universal structure for representation of common information (mostly declarative) in different domains. Different types of information, regardless of the concrete syntax of their representation, in the abstract syntax are, in general, the (multi-) typed graphs [Agrawal A. et al., 2005] [Ehrig K. et al., 2005]. Therefore many problems of information transformation appearing in different areas of professional activity, technological spaces [Kurtev, 2002] [Bezivin, 2005], as well as at their joint, can be formulated in terms of the semantic networks transformation. These problems include: the problem of ontology mapping, knowledge bases transformation, programs transformation, models transformations in model-driven engineering (MDE). The problem of semantic networks transformation generally can be formulated as follows: on basis of initial semantic network and the transformation specification to generate the new semantic network that meets this specification.

At the same time, in different technological spaces the efficient models and methods as well as software systems for information transformation were developed independently. However, as practice and experience find out, the methods and formalisms applied in a technological space often cannot be directly "mapped" to another one [Czarnecki, 2006].

The main drawback of the existing methods for different types of semantic information transformation is their orientation on a special technological space or one type of transformation – endogenous or exogenous, or even on a specific class of problems. The other lack of the existing methods is that they are designed for group of professionals who have special knowledge and skills in a particular domain or technological space.

This paper presents a method for semantic network classes transformation, based on the models of semantic network classes and their transformations which are invariant to the different technological spaces [Timchenko, 2012]. This method can be represented as a set of three basic methods of transformation: "text-to-semantic network", "semantic network-to-text" and "semantic network-to-semantic network".

The reason that it should be possible to perform the "text-to-semantic network" and "semantic network-to-text" transformations is that in many domains the textual representation of the information (representation in a concrete syntax) are often more familiar to people than structural representation, while the latter (the representation in the abstract syntax) is usually much more convenient for software systems processing. Thus, the source information can have both textual and structural representation (in the form of a semantic network). Target information can also be required to present both in textual and in the structural form.

Another argument is that it is often faster and easier to create a representation of information in a network of concepts automatically based on textual representation than manually. The latter means using some editing tools to generate such a network of concepts using some semantic network class as meta information. An example is the creation of the program in a programming language in the form of an abstract syntax tree in concordance with the grammar of this language.

The specification for the "semantic network-to-semantic network" transformation method is a description of the structure mapping in the terms of language for structure mappings description. The specification for the "text-to-semantic network" and "semantic network-to-text" transformation methods is the description of syntax restrictions (in terms of the syntax restrictions model) [Timchenko, 2012].

MAIN PART

The first section presents the description of the main ideas and the key steps of the "text-to-semantic network" transformation method.

The second section presents the description of the main ideas and the key steps of the "semantic network-to-text" transformation method.

The third section presents the description of the main ideas and the key steps of the "semantic network - to-semantic network" transformation method.

CONCLUSION

There have been developed "text-to-semantic network", "semantic network-to-text" and "semantic network-to-semantic network" methods of transformations defining how on the basis of the appropriate transformation specification and the input information to generate the output one, that meets this specification. "Text-to-semantic network" transformation method allows you to analyze the text in the language specified by a CF-grammar without restrictions, while most common implemented methods of analysis require that the input grammar satisfies certain properties.

It should be noted that, depending on the problem of transformation, these methods can be combined together in different ways. In total three combinations are allowed, depending on the representation of input information (text or in the form of a semantic network) and the required representation of output information (text or in the form of a semantic network). The possibility to vary the order of the methods application can significantly expand the range of applications of the results of the work.

The paper was written with financial support from RFBR, project 12-07-00179-a and from FEB RAS under Program of the RAS branch "Branch of nanotechnologies and information technologies", project 12-I-OHIT-04.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 519.711.74

ПОСТРОЕНИЕ АССОЦИАТИВНОЙ МОДЕЛИ ПАМЯТИ. КОГНИТИВНЫЙ ПОДХОД

Жилякова Л.Ю.

*Институт проблем управления им. В.А. Трапезникова РАН,
г. Москва, Россия*

zhilyakova.ludmila@gmail.com

В работе излагаются принципы построения модели ассоциативной памяти, основанной на динамической ресурсной сети. Вершины сети соответствуют понятиям предметной области, ребра – ассоциативным связям между ними. Топология сети динамически изменяется в процессе ее наполнения. Задан алгоритм структурирования сети и изменения ее топологии, с помощью которого сущности, соответствующие базовым когнитивным категориям, в процессе самоорганизации становятся центрами соответствующих образов.

Ключевые слова: ассоциативная память, ресурсная сеть, категоризация, когнитивная категоризация.

ВВЕДЕНИЕ

Работа посвящена описанию структуры модели ассоциативной памяти и принципов ее построения. Основной особенностью модели является непрерывное изменение топологии, которое состоит в создании новых связей и изменении весов ребер, что влечет за собой усиление одних ассоциаций и ослабление других. Эти изменения происходят как в процессе функционирования сети, так и в процессе ее построения.

Математический аппарат, используемый при моделировании, – динамическая потоковая модель, функционирующая в дискретном времени, названная ресурсной сетью [Кузнецов, 2009]. Вершины сети могут содержать произвольное количество ресурса; веса ребер соответствуют их пропускным способностям. Модель нелинейна: в каждый момент времени вершина отдает ресурс во все смежные с ней вершины в соответствии с одним из двух правил, в зависимости от количества ресурса, которым она обладает.

Существующие графовые модели со сходными принципами функционирования можно условно разделить на два больших класса. Первый из них – модели рассеяния на графах, описываемые конечными однородными цепями Маркова [Blanchard et al., 2011], [Lovasz et al., 1995]. Второй – пороговые целочисленные модели, такие, например, как chip-firing games [Bjorner et al., 1991, 1992]. Ресурсная сеть обладает пороговым значением ресурса, зависящим только от пропускных способностей всех ребер: при

суммарном ресурсе, меньшем порогового значения, она ведет себя как марковская модель, при ресурсе, превосходящем пороговое значение, в сети появляются вершины, функционирующие по правилам, задающимся в целочисленных пороговых моделях. Эта особенность, с одной стороны, влечет за собой большое разнообразие переходных процессов, с другой же, – бесконечная делимость ресурса обеспечивает сходимость в тех случаях, когда целочисленные модели не сходятся.

В данной работе при построении ассоциативной модели используются принципы когнитивной категоризации, придающие модели дополнительное сходство с человеческой памятью. Эти особенности могут в дальнейшем стать основой для построения «быстрых рассуждений», присущих человеческому мышлению.

1. НЕКОТОРЫЕ ПОДХОДЫ К МОДЕЛИРОВАНИЮ АССОЦИАТИВНОЙ ПАМЯТИ

Понятие ассоциативности применяется в равной степени к человеческой и к компьютерной памяти. Ассоциативность человеческой памяти означает, что мысль о некотором предмете способна породить последовательность образов, связанных с начальным предметом не только семантически, но некоторыми ассоциациями. О том, что память человека ассоциативна, писал еще Аристотель. Его интерпретация перехода по ассоциативным связям такова: поиск воспоминания возможен только по некоторой цепочке, начало которой доступно нам в настоящий момент времени, а в конце ее находится

нужное воспоминание [Аристотель]. Эти же идеи высказываются и современными авторами: «Вспоминать – значит иметь воспоминание или приступить к поиску воспоминания» [Рикёр, 2004]. Таким образом, чтобы вспомнить нечто, необходимо иметь либо непосредственный доступ к нужной информации, либо доступ к месту в памяти, из которого эта информация достижима.

Понятие компьютерной ассоциативной памяти связано с рядом серьезных упрощений. Память в компьютере является локально адресуемой, поэтому моделирование ассоциативной памяти связано, в первую очередь, с преодолением необходимости обращения к данным по их физическому адресу. Чаще всего под компьютерной ассоциативной памятью в широком смысле понимается память, в которой поиск информации производится по ее содержанию.

Математические модели долговременной памяти имитируют некоторые процессы, происходящие в естественной памяти. В модели долговременной памяти (модель памяти со случайной выборкой) предложенной В.Л. Стефанюком [Стефанюк, 2004], [Стефанюк, 2011], повышение скорости поиска по образцу происходит за счет дублирования информации и оптимизации количества копий, созданных для каждой сущности. Тем самым, повышается вероятность извлечь из памяти сущности, поиск которых происходит чаще.

Идея дублирования информации в ином контексте была предложена автором в работе [Жилиякова, 2008]. В этой модели использовался теоретико-множественный подход. Ассоциативные связи между сущностями определяются через наличие у них общих свойств. Число копий новой сущности, добавляемой в память, равно числу свойств, которыми она обладает. Каждому свойству соответствует множество копий сущностей, обладающих этим свойством. Например, чтобы выделить все зеленые или круглые предметы, достаточно найти множество с соответствующим именем. Предметы, обладающие некоторым набором свойств, ищутся как пересечения соответствующих множеств. Поскольку копируются не сами сущности, а ссылки на них, увеличение объема памяти незначительно.

В настоящей работе предложена сетевая модель, позволяющая структурировать память таким образом, что поиск информации можно осуществлять, следуя по ассоциативным цепочкам, создающимся и изменяющимся автоматически на уровне топологии сети – в процессе поступления и обработки запросов. Причем, часто используемая информация оказывается более доступной, и сила ассоциативных связей тем больше, чем чаще сущности упоминаются вместе. Такое хранение и поиск информации в памяти отчасти имитирует клеточные ансамбли Хебба [Hebb, 1959].

Ассоциативная связь между сущностями в этой модели тем сильнее, чем чаще они попадают вместе

во входное или выходное множество вершин запроса. Под «выполнением запроса» здесь понимается процесс, который описан у Аристотеля и Рикёра – движение по цепочкам ассоциаций от некоторого начального набора сущностей к сущностям, ассоциированным с ними. Он осуществляется с помощью перераспределения ресурса, соответствующего «яркости образа» в памяти. После выполнения каждого запроса происходит изменение топологии сети – образуются новые ребра между вершинами, принадлежащими одному входному или выходному множеству (если таковых не было); каждое ребро, по которому проходил ресурс, увеличивает свою пропускную способность.

Добавление новой информации в модель предполагает, что каждая новая сущность будет связана хотя бы с одной уже имеющейся в модели. При этом у вершины, уже существующей в сети, увеличивается пропускная способность петли, отвечающая за автоассоциацию [Кохонен, 1980].

2. АССОЦИАТИВНАЯ РЕСУРСНАЯ СЕТЬ

В работе [Жилиякова, 2009] описана модель памяти, названная *ассоциативной ресурсной сетью*. В ее основе лежит неоднородная ресурсная сеть.

Ресурсная сеть – динамическая потоковая модель, предложенная в [Кузнецов, 2009]. Она представляет собой ориентированный граф с множеством вершин $V = \{v_i\}$, $|V| = n$. Каждая вершина v_i в момент t содержит ресурс $q_i(t) \geq 0$. Емкости вершин не ограничены. Вершины в каждый такт дискретного времени t обмениваются ресурсами, следуя заданным правилам. В сети выполняется закон сохранения: при ее функционировании ресурс не поступает извне и не расходуется. Ребра (v_i, v_j) имеют ограниченные пропускные способности r_{ij} . Вершины могут иметь петли (v_i, v_i) с пропускной способностью, равной r_{ii} . Ресурс, попавший в петлю на такте t , вернется в вершину на следующем такте.

Ресурсная сеть является пороговой моделью: в каждый момент каждая вершина отдает ресурс во все исходящие ребра по одному из двух правил: если величина ресурса в ней больше суммарной выходной пропускной способности r_i^{out} , она отдает по полной пропускной способности в каждое ребро, оставляя себе излишки; если ресурс в вершине меньше r_i^{out} , он распределяется пропорционально пропускным способностям во все исходящие ребра. Формально правила описаны в ряде работ, например, [Кузнецов и др., 2010].

Ассоциативная ресурсная сеть отличается от обычной ресурсной сети, в первую очередь, тем, что имеет переменную топологию. Вершины в ней соответствуют сущностям предметной области, ребра – ассоциативным связям между ними.

Количество ресурса в вершине в данной модели

соответствует яркости вершины при поиске. Ребра обозначают ассоциативные связи. Чем больше пропускная способность ребра между двумя вершинами, тем больше сила ассоциации между соответствующими сущностями. Пропускная способность петли соответствует силе автоассоциации. Отношение ассоциации всегда обоюдно, и поэтому каждая пара смежных вершин связана двумя противоположно ориентированными ребрами.

В ассоциативной сети существует быстрое и медленное время. Одному такту медленного времени соответствует исполнение одного запроса. Каждый запрос выполняется в быстром времени, которое соответствует времени *ресурсной сети*.

Запрос – это помещение ресурса в одну или несколько вершин сети. Ответ на запрос – распределение ресурса после его стабилизации. После выполнения запроса пропускные способности всех ребер, по которым тек ресурс, увеличиваются пропорционально суммарному пропущенному ресурсу. Если в начальном множестве запроса существуют несвязанные вершины, в сети создаются новые двусторонние пары, задающие новую ассоциацию.

Для того чтобы ресурс не растекался из начального множества вершин одинаково по всем направлениям, используются рекурсивные запросы.

Под рекурсивным запросом будем понимать многократный запрос в один такт медленного времени, входное множество вершин которого изменяется в зависимости от выходного множества на предыдущем шаге по одному из наперед заданных правил.

Для предотвращения неограниченного роста суммарной пропускной способности сети, вводится процедура нормировки. Указывается диапазон, в котором может варьироваться суммарная пропускная способность; когда достигается верхняя граница, вся сеть нормируется к нижней границе диапазона. Эта процедура реализует естественное забывание: петли редко используемых в запросах вершин и редко проявляемые ассоциации «истончаются». Если вычислительные ресурсы позволяют, перенормировку сети желательно делать после каждого нового запроса.

3. ИСПОЛЬЗОВАНИЕ ПРИНЦИПОВ КОГНИТИВНОЙ КАТЕГОРИЗАЦИИ ПРИ ПОСТРОЕНИИ АССОЦИАТИВНОЙ СЕТИ

Для того чтобы все ребра сети имели веса, адекватно описывающие силы ассоциативных связей, необходимо изменять пропускные способности не только на этапе обращения к сети с запросами, но и на этапе построения сети, т.е. на этапе *создания* памяти. Здесь большую роль играет последовательность наполнения сети. Наибольшую яркость при поиске всегда должны иметь сущности, принадлежащие *базовому уровню категоризации*, –

они являются центрами соответствующих им образов [Голицын, 1997]. Поэтому именно эти сущности должны быть помещены в сеть первыми.

В когнитивной категоризации центральным является понятие базового уровня. Для членов категорий базового уровня характерно следующее [Lakoff, 1987] [Кузнецов, 2012]):

- они имеют единый, целостно воспринимаемый ментальный образ (гештальт);
- они быстро узнаются;
- в качестве их имен используются наиболее короткие и общеупотребительные слова, первичные с точки зрения вхождения в словарный запас языка;
- большинство признаков членов категории хранится на этом уровне;
- формирование категорий у детей начинается с категорий базового уровня.

«Базовое» не значит «элементарное», поскольку концепты базового уровня не являются атомарными блоками, лишенными внутренней структуры. Базовые категории с когнитивной точки зрения находятся «в середине» иерархии общего-конкретного. Обобщение происходит вверх от базового уровня, специализация – вниз. Например: *собака* – базовая категория, *хищник* – обобщение, *овчарка* – специализация.

Из сказанного следует, что при любом запросе к ассоциативной памяти в выходное множество уже при малом числе шагов быстрого времени должны попасть соответствующие вершины базового уровня, определяющие гештальты. К ним должны вести ребра с наибольшей пропускной способностью. Поэтому наполнение сети информацией должно происходить по правилам, учитывающим эти свойства. Для реализации наполнения сети, начиная с категорий базового уровня, не нужны специальные дополнительные построения. При заданном алгоритме наполнения сети это происходит автоматически. Алгоритм занесения новой вершины:

1. Если новая сущность принадлежит базовой категории, создается соответствующая ей вершина и ребра, связывающие ее с другими вершинами (в зависимости от контекста). Выход.
2. Если новая сущность принадлежит конкретизации/обобщению базовой категории, то
 - а. если вершина, соответствующая базовой категории, уже есть в сети, с ней создается связь, независимо от остальных связей новой вершины; пропускная способность ее петли увеличивается,
 - б. если вершины, соответствующей базовой категории нет, сначала создается эта вершина, и затем вершина, соответствующая вносимой сущности.

Сеть, построенная по такому алгоритму, и изменяющая топологию после каждого запроса, будет иметь вершины базового уровня в качестве самых доступных при поиске. Таким образом, описанные алгоритмы в некотором смысле можно рассматривать как алгоритмы создания гештальтов.

ЗАКЛЮЧЕНИЕ

В работе описаны основные правила построения ассоциативной ресурсной сети, правила изменения ее топологии, основные виды рекурсивных запросов и алгоритмы их реализации. Основное внимание уделено формированию базовых категорий, каждая из которых является центром соответствующего образа.

Работа выполнена при поддержке РФФИ (грант № 11-01-00771).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Аристотель] Аристотель. О памяти и припоминании. [Электронный ресурс]. URL: http://www.ec-dejavu.net/m/Memory_Greek.html#aristotle
- [Голицын, 1997] Голицын Г.А. Образ как концентратор информации. // Математика и искусство. Труды международной конференции. Москва, 1997, с.96-99.
- [Жилякова, 2008] Жилякова Л.Ю. Структурирование знаний в ассоциативной модели // Одиннадцатая национальная конференция по искусственному интеллекту с международным участием КИИ' 2008. Труды конференции, том 1, М., URSS, 2008. с. 104-111.
- [Жилякова, 2009] Жилякова Л.Ю. Поиск в ассоциативной модели памяти. // IX международная конференция имени Т.А. Таран ИАИ-2009. Киев, «Просвіта», 2009. с. 124-130.
- [Кохонен, 1980] Кохонен Т. Ассоциативная память. – М.: Мир, 1980.
- [Кузнецов, 2009] Кузнецов О.П. Однородные ресурсные сети. I. Полные графы. // Автоматика и телемеханика, 2009, № 11, с.136-147.
- [Кузнецов и др., 2010] Кузнецов О.П., Жилякова Л.Ю. Двусторонние ресурсные сети – новая потоковая модель // Доклады АН, 2010, том 433, №5, с.609-612.
- [Кузнецов, 2012] Кузнецов О.П. Когнитивная семантика и искусственный интеллект // Искусственный интеллект и принятие решений, 2012, №4, с.32-42.
- [Рикёр, 2004] Рикёр П. Память, история, забвение. М., 2004.
- [Стефанюк, 2004] Стефанюк В.Л. Локальная организация интеллектуальных систем: модели и приложения. – М. Физматлит, 2004.
- [Стефанюк, 2011] Стефанюк В.Л. Феноменологические модели биологической памяти. // Сборник научных трудов VI-й Международной научно-технической конференции Интегрированные модели и мягкие вычисления в искусственном интеллекте. Т.1. – М. Физматлит, 2011. с. 89-100.
- [Björner et al., 1991] A. Björner, L. Lovasz and P. Shor, Chip-firing games on graphs, *Europ. J. Comb.* 12 (1991), 283–291.
- [Björner et al., 1992] A. Björner and L. Lovasz, Chip-firing games on directed graphs, *J. Algebraic Combinatorics* 1 (1992), 305–328.
- [Blanchard et al., 2011] Blanchard, Ph., Volchenkov, D. *Random Walks and Diffusions on Graphs and Databases: An Introduction* (Springer Series in Synergetics). Springer-Verlag – Berlin–Heidelberg, 2011.
- [Hebb, 1959] Hebb D., *Intelligence, brain and the theory of mind* // *Brain*. 1959, v. 82.
- [Lakoff, 1987] Lakoff J. 1987. *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*. University of Chicago Press. (Русский перевод: Лакофф Д. Женщины, огонь и опасные вещи: Что категории языка говорят нам о мышлении, М. 2004).
- [Lovasz et al., 1995] L. Lovasz and P. Winkler. *Mixing of Random Walks and Other Diffusions on a Graph* // *Surveys in*

Combinatorics, 1995 (ed. P. Rowlinson), London Math. Soc. Lecture Notes Series 218, Cambridge Univ. Press, 119–154.

MODELLING OF THE ASSOCIATIVE MEMORY. COGNITIVE APPROACH

Zhilyakova L. Yu.

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

zhilyakova.ludmila@gmail.com

olkuznes@ipu.rssi.ru

In the paper the principles of the modelling of associative memory based on a dynamic resource network are outlined. The vertices of a network correspond to the concepts (entities) of domain; the edges denote associations between the two entities. The network topology changes dynamically in the process of filling the network and execution of queries. An algorithm for structuring the network and changing its topology is specified. According to it, the entities that meet the basic cognitive categories, become in the process of self-organization the centers of the corresponding images.

INTRODUCTION

This paper describes the structure of an associative memory model and the principles of its construction. The main feature of the model is a continuous change in the topology, such as creation of new connections and changing the weights of edges, resulting in a strengthening of some associations and weakening others. These changes occur both in the operation of the network, and in the process of its construction.

MAIN PART

The associative model is based on the non-homogeneous resource network with a variable topology. The resource network is a dynamical flow model: resource is contained at vertices which exchange it at every step of discrete time. Vertices are able to store unlimited amount of resource.

The *associativity* of a network lies in the property that the most frequently used information is the most accessible when searching from any place of memory. Moreover, the entities of a base cognitive level are always the most accessible by construction.

CONCLUSION

The construction of an associative model uses the principles of cognitive categorization, giving to the model an additional similarity to the human memory. These features could then be the basis for the construction of the "fast reasoning" inherent in human thinking.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.9:510

ТЕХНОЛОГИИ СВЯЗЫВАНИЯ ДАННЫХ В ПРОСТРАНСТВЕ ОТКРЫТЫХ ДАННЫХ НА ПРИМЕРЕ МАТЕМАТИЧЕСКОЙ КОЛЛЕКЦИИ

Невзорова О.А., ** Кириллович А.В.*

* *Казанский федеральный университет,
г. Казань, Россия*

alikh.kirillovich@gmail.com

** *Научно-исследовательский институт «Прикладная семиотика» АН Республики Татарстан,
Казанский федеральный университет,
г. Казань, Россия*

onevzoro@gmail.com

В работе рассматриваются программные технологии связывания данных в пространстве открытых данных с помощью программных систем LIMES и SILK, и эксперименты по связыванию математического RDF-набора данных.

Ключевые слова: связывание данных; пространство открытых связанных данных.

Введение

Проект Linked Open Data (LOD)¹ является наиболее значимым по результатам примером принятия и применения принципов Linked Data (Связанные данные). На сегодняшний день технологии Linked Data все более широко используются производителями первичных данных. Главное преимущество заключается в стандартизованном подходе к структурированию и хранению интегрированных данных. Как правило, данные загружаются и представляются в виде RDF (Resource Description Framework²), т.е. триплетов вида «субъект — предикат — объект», из таких традиционных хранилищ как реляционные базы данных или, реже, из веб-страниц или полуструктурированных текстовых документов, используя принципы связанных данных, предложенные Т. Бернерсом-Ли³. Ссылки RDF соединяют данные из различных источников в единый глобальный граф RDF и позволяют браузерам и поисковым роботам Связанных данных перемещаться между источниками данных.

В сентябре 2011 г. объем данных в LOD составлял более 30 млрд. триплетов, хранящихся в

примерно 300 наборах данных. Среди доминирующих предметных областей — данные из правительственных источников (43% по числу триплетов), географические данные (22%) и науки о жизни (биология, биохимия, генетика и др. - 9%).

Технологии Linked Data применяются для совместного использования данных, относящихся к широкому спектру различных тематических доменов, а также данных, относящихся к различным доменам. Характерным примером и прототипом междоменных Связанных данных является DBpedia – набор данных, автоматически извлекаемый из общедоступного ресурса Википедия. Одним из перспективных направлений использования Связанных данных являются приложения в области образования, науки и культуры. Существенное продвижение имеется, например, в задачах интеграции библиотечных каталогов на глобальной основе, установления взаимных связей между содержанием библиотечных каталогов по теме, географическому расположению или историческому периоду, установление взаимных связей между библиотечными каталогами и данными третьих сторон (архивы изображений и видео, базы знаний типа DBpedia). Научные коллекции представлены в облаке LOD пока достаточно фрагментарно и поддерживаются, например, в наборах данных

¹ <http://lod2.eu>

² <http://www.w3.org/RDF/>

³ <http://www.w3.org/DesignIssues/LinkedData.html>

ACM⁴, DBLP⁵ и CiteSeer⁶. Основным содержимым этих наборов данных являются стандартные метаданные статей (название, год публикации, информация об авторах и др.).

В статье рассматриваются программные технологии связывания данных из математического RDF-набора, подготовленного на основе статей математического журнала «Известия ВУЗов. Математика», издаваемого в Казанском (Приволжском) федеральном университете. Можно ожидать, что публикация в облаке LOD данных математической коллекции позволит распространить концепцию открытых данных в математическом сообществе, поможет профессиональным исследователям и студентам получить удобный доступ к специальным знаниям, интегрированным в глобальную систему знаний.

1. Стек программных технологий подготовки математического RDF-набора данных

Программные технологии подготовки математического RDF-набора данных подробно описаны в [Невзорова и др., 2012].

В основе подхода лежит представление коллекции математических документов в виде единого семантического графа, в котором как вершины – объекты математического знания, так и ребра – связи между ними, определяются множеством специализированных словарей (онтологий), которые специфицируют как термины из области математики, так и элементы логической структуры математического документа.

Стек программных технологий, разработанный для публикации математических данных в LOD, включает набор программных систем, предназначенных для выполнения следующих задач:

- преобразование формата данных;
- аннотирование текста;
- семантическое аннотирование;
- аннотирование метаданных;
- генерация RDF-набора;
- связывание данных.

В статье рассматриваются основные результаты по связыванию данных существующими инструментами и проблемы связывания с наиболее популярными Интернет-ресурсами.

2. Программные инструменты связывания данных

В Linked Data основополагающим принципом связывания идентичных ресурсов является

использование ссылок на другие URI для поиска фактов относительно интересующего объекта. Типичным решением является формирование ссылочного утверждения `owl:sameAs` между различными семантически похожими URI, обозначающими эти ресурсы.

Целью процесса сопоставления является поиск семантически связанных сущностей и установление типовых ссылок между ними, большинство из которых имеют формальные свойства (такие как транзитивность, симметрия и т.д.), которые могут использоваться в механизмах рассуждений и других приложениях для выведения новых знаний.

Можно выделить две категории систем и методов выявления связей между сущностями (классами, свойствами или экземплярами) баз знаний:

- первая категория использует методики сопоставления онтологий и имеет целью установление ссылок между онтологиями, лежащими в основе двух источников данных;
- вторая, более сложная категория занимается попарным сопоставлением семантически сходных экземпляров и выявляет наличие связей между экземплярами в различных источниках данных.

В статье рассматривается задача выявления связей между экземплярами объектов, заданных различными URI.

Для указанных целей предлагается применять подходы, реализованных в программных системах SILK⁷ и LIMES⁸. Далее кратко рассмотрим решения по связыванию данных в системах LIMES и SILK.

2.1. Система LIMES

Система LIMES (Link Discovery Framework for MEtric Spaces) - система выявления связей для метрических пространств – ориентирована на решение задачи сопоставления экземпляров на основе эффективных (по времени) методов выявления связей между источниками Связанных данных. Проблема масштабируемости процесса выявления связей решается с использованием аксиомы треугольника в метрических пространствах для определения оценок сходства экземпляров. Официальная страница системы LIMES находится на сайте исследовательской группы Agile Knowledge Engineering and Semantic Web (AKSW) по адресу¹¹.

Общая последовательность выполняемых функций по решению задачи связывания в системе LIMES представляется в виде схемы (рис.1). В систему поступает информация об исходном источнике данных (Source), целевом источнике данных (Target) и спецификации устанавливаемой связи (Link Specification) в виде конфигурационного файла.

⁴ <http://acm.rkbexplorer.com/>

⁵ <http://dblp.rkbexplorer.com/>

⁶ <http://citeseer.rkbexplorer.com/>

⁷ <http://www4.wiwiw.fu-berlin.de/bizer/silk/>

⁸ <http://aksw.org/Projects/limes>

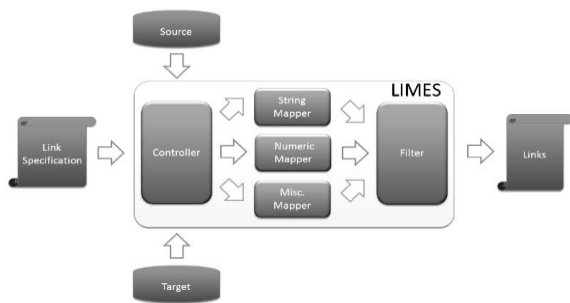


Рисунок 1 – Функции системы LIMES

Информация поступает в контроллер (Controller), где происходит разделение данных из источников по разным типам. Далее строки обрабатываются в строковом преобразователе (String Mapper). Числовые значения (а также все значения, которые могут быть эффективно преобразованы в векторное пространство) отображаются в метрическое пространство и обрабатываются с помощью алгоритма HYPO (Numeric Mapper). Все прочие значения преобразуются в смешанном преобразователе (Miscellaneous Mapper). Результаты работы всех преобразователей фильтруются и объединяются с помощью эффективных по времени алгоритмов (Filter). Затем генерируются связи (Links).

Для связывания данных пользователю системы LIMES необходимо предоставить следующую информацию: источники данных; объекты, подлежащие связыванию, типы объектов для связывания; условия связывания объектов. Вся эта необходимая информация должна быть представлена в виде конфигурационного файла, написанного на конфигурационном языке LIMES (LIMES Configuration Language). На рис. 2 представлен графический интерфейс конфигурационного файла. В версии LIMES 0.5RS1 пользователь заполняет соответствующие поля формы, после чего конфигурационный файл

формируется автоматически.

Назначение основных полей графического интерфейса:

- **Endpoint** – точка доступа SPARQL к источнику данных. Версия LIMES 0.5RS1 поддерживает файлы формата CSV, N3 (для файла-источника данных в поле Endpoint указывается абсолютный путь);
- **Var** – переменная, обозначающая источник данных;
- **Pagesize** – максимальное количество триплетов, возвращаемое точкой доступа SPARQL на каждый запрос. Если это количество не ограничено или источником данных является файл, значение поля Pagesize устанавливается равным -1;
- **Restriction** – ограничения на входные данные, позволяющие уменьшить количество сравнений, например, за счет указания принадлежности экземпляров данных к определенному классу, а также выполнения стандартной предобработки данных;
- **Property** – указание классов (типов объектов), которые будут участвовать в связывании.

Выбор типов объектов, которые могут участвовать в связывании, осуществляется на этапе описания источников данных. Правила идентификации объектов являются важной составляющей конфигурационного файла.

В версии LIMES 0.5RS1 для составления условия связывания используются поля, показанные на рис. 3. В поле Metric описывается метрика, которая вычисляется для пары объектов. Система LIMES поддерживает набор встроенных метрик:

- для строк – расстояние Левенштейна (Levenshtein), триграммы (Trigrams), коэффициент перекрытия (Overlap), коэффициент Жаккарда (Jaccard), косинусный коэффициент (Cosine);

LIMES

Source:	Target:
Endpoint: <input type="text" value="http://dbpedia.org/sparql"/>	Endpoint: <input type="text" value="http://www4.wiwiss.fu-berlin.de/drugbank/sparql"/>
Graph: <input type="text" value="-1"/>	Graph: <input type="text" value="-1"/>
Var: <input type="text" value="?x"/>	Var: <input type="text" value="?y"/>
Pagesize: <input type="text" value="1000"/>	Pagesize: <input type="text" value="-1"/>
Restriction: <input type="text" value="?x rdf:type dbpedia:Drug"/>	Restriction: <input type="text" value="?y rdf:type drugbank:drugs"/>
Property: <input type="text" value="rdfs:label"/>	Property: <input type="text" value="drugbank:genericName"/>

Metric:

Output:

Execution:

Acceptance:	Review:
Threshold: <input type="text" value="0.9"/>	Threshold: <input type="text" value="0.8"/>
Relation: <input type="text" value="owl:sameAs"/>	Relation: <input type="text" value="owl:sameAs"/>

Detected prefixes:

rdf: (-)

rdfs: (-)

owl: (-)

dbpedia: (-)

drugbank: (-)

rdfs: (-)

drugbank: (-)

Рисунок 2 – Графический интерфейс создания конфигурационного файла

Рисунок 3 – Поля для составления условия связывания данных

- для числовых векторов – Евклидово расстояние (Euclidean), wgs84.

Система LIMES оперирует формальными выражениями для построения сложных метрик путем комбинирования метрических и логических бинарных операций. К метрическим операциям относятся встроенные метрики, а также операторы нахождения минимума (MIN), максимума (MAX), суммы (ADD), произведения (MULT) и др. Логические операции (AND, OR, DIFF и др.) позволяют объединять или фильтровать результаты метрических операций.

В поле Output назначается формат результирующих данных (поддерживаются форматы N3 и TAB). В разделе Acceptance указывается значение порога метрики (поле Threshold), при достижении которого устанавливается связь для данной пары объектов и результат записывается в выходной файл. В разделе Review указывается порог, значение которого меньше, чем в разделе Acceptance. Соответственно, связи, для которых значение метрики попадет в промежуток между этими двумя порогами, будут записаны в другой файл для последующей ручной обработки. В поле Relation указывается тип связи объектов, например, owl:sameAs.

Результатом работы системы LIMES является файл, содержащий данные об установленных связях.

В разделе 3 подробно рассматриваются эксперименты по связыванию данных математического RDF-набора данных в системе LIMES.

2.2. Система SILK

Официальная страница проекта находится на сайте Свободного Университета Берлина: <http://www4.wiwiiss.fu-berlin.de/bizer/silk/>.

Система SILK предназначена для нахождения отношений между элементами из разных наборов Связных Данных. Система принимает на входе схему связывания, на выходе возвращает набор найденных связей. Для связывания данных пользователю системы SILK необходимо предоставить следующую информацию: источники данных; объекты, подлежащие связыванию, типы объектов для связывания; условия связывания объектов.

Взаимодействие с системой SILK осуществляется посредством командной строки

(Silk Single Machine), либо с помощью графического интерфейса (SILK Workbench).

Источники данных задаются указанием SPARQL-точки доступа, либо как локальный RDF-файл в разных форматах (RDF/XML, N3, Turtle и др.).

Схема связывания задается как XML-файл в формате Silk-LSL (Silk Link Specification Language), либо конструируется при помощи графического интерфейса.

3. Эксперименты по связыванию математического RDF-набора данных в системах LIMES и SILK

В подготовленном наборе данных IVM [Невзорова и др., 2012] представлены метаданные 1456 статей журнала «Известия ВУЗов. Математика», издаваемого в Казанском федеральном университете, за 1997-2009 годы, которые включают:

- название публикации (на русском и английском языках), выпуск журнала и его год, страницы в журнале, список литературы (сокращенный), авторы;
- автор публикации: имя (на русском и английском языках, сокращенная и полная формы), адрес электронной почты, адрес страницы на ресурсе mathnet.ru, организация;
- организация (как место работы): название (несколько вариантов написания, на русском и английском);
- адрес публикации на ресурсе mathnet.ru.

Для описания метаданных статей используется популярная схема AKT Reference Ontology (AKT). В наборе IVM представлено около 24000 триплетов метаданных. Доступ к данным IVM осуществляется через локальный RDF-файл.

Для представления математических знаний в ходе разработки проекта по математическому RDF-набору IVM с помощью экспертов-математиков Казанского федерального университета была подготовлена онтология профессиональной математики OntoMath^{pro}. Главная цель OntoMath^{pro} – предоставить информационный терминологический ресурс для автоматизированной обработки электронных профессиональных математических публикаций на русском языке. Онтология OntoMath^{pro} содержит определения как общепринятых математических понятий, так и развивающуюся терминологию, в основном, из следующих разделов математики: теория чисел, теория множеств, алгебра, геометрия, математическая логика, дискретная математика, теория алгоритмов, математический анализ, дифференциальные уравнения, численные методы, теория вероятностей и математическая статистика. Источниками для определения семантики концептов OntoMath^{pro} служили: классические учебники соответствующих разделов математики,

электронные ресурсы – Wikipedia и Cambridge Mathematical Thesaurus, научные статьи журнала «Известия вузов. Математика», а также профессиональные знания и опыт экспертов-математиков.

В качестве языков представления OntoMath^{PRO} выбраны языки OWL-DL/RDFS, которые предоставляют высокие выразительные логические средства и алгоритмически разрешимы не только теоретически, но и практически, с помощью возможностей таких инструментов логического вывода, как Pellet и Fact++. В частности, разделы математики и элементы математического знания выражались с помощью концепта owl:Class. Уникальный идентификатор ресурса (URI) каждого класса представляет собой суррогатный ключ, который составлен из пространства имен онтологии и кода, однозначно идентифицирующего класс внутри онтологии.

Опишем ряд экспериментов по связыванию, выполненных с помощью инструментальных средств SILK и LIMES для RDF-набора IVM и онтологии OntoMath^{PRO}.

3.1 Связывание научно-исследовательских организаций в системе LIMES

Связывание данных о научно-исследовательских организациях выполнено для RDF-набора IVM и набора связанных данных CORDIS (Community R&D Information Service: RKBExplorer).

Доступ к данным CORDIS может осуществляться как через локальные RDF-файлы, так и через SPARQL-точку доступа. Список ссылок на локальные RDF-файлы содержится по адресу: <http://dblp.rkbexplorer.com/sitemap.xml>. Каждый из файлов содержит информацию о научных организациях, сотрудники которых имеют в соответствующий год представленную в CORDIS публикацию. SPARQL-точка доступа имеет адрес: <http://cordis.rkbexplorer.com/sparql/>.

Многие элементы из IVM имеют эквивалентные им элементы из CORDIS, т.е. соответствуют одной и той же реальной организации. Необходимо найти пары эквивалентных элементов и соединить их ссылками owl:sameAs. Эквивалентность организаций определяется по схожести только одного признака — названия организации.

Оба набора данных используют онтологию АКТ, организации в обоих наборах имеют тип akt:Organization. В наборе IVM название организации задается свойством akt:name, в наборе CORDIS — свойством akt:has-pretty-name.

На рис. 4 приведено решение по связыванию набора IVM и локального файла cordis-projects-2004.rdf набора CORDIS, подготовленного с помощью графического интерфейса системы LIMES.

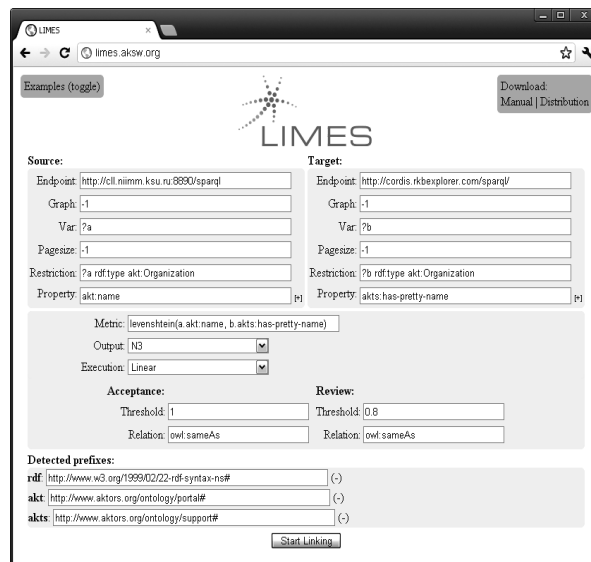


Рисунок 4 - Решение по связыванию наборов IVM и Cordis

Построенный список содержит 87 найденных гарантированно правильных связей при ограничениях на размер выборки из SPARQL-точки доступа Cordis.

3.2 Связывание классов онтологии OntoMath^{PRO} с ресурсами DBPedia в системе SILK

Связывание классов онтологии OntoMath^{PRO} с данными из ресурсов DBPedia осуществлялось на основе следующих признаков:

- название класса (rdfs:label). Название класса из OntoMath^{PRO} сравнивается с аналогичным названием ресурса из DBPedia;
- ссылка на статью английской Википедии в поле комментария (rdfs:comment). Адрес ссылки сравнивается со значением входящего свойства foaf:primaryTopic ресурса из DBPedia;
- ссылка на статью русской Википедии в поле комментария (rdfs:comment). Название статьи (т.е. та часть адреса ссылки, которая идет после «<http://ru.wikipedia.org/wiki/>») сравнивается с названием ресурса (rdfs:label) из DBPedia.

При этом из всего множества ресурсов DBPedia отбираются для связывания только ресурсы из категории *category:Mathematics* и категорий разделов математики до 4-го уровня вложенности. Введенное ограничение позволяет обеспечить необходимые требования по точности и полноте связывания. Например, точность можно повысить за счет исключения из рассмотрения связей с классами-омонимами из других (нематематических) областей знаний.

Например, #E64 (Якорь из области теории групп) ≠ dbpedia:Anchor (Морской якорь).

Ограничение на уровень вложенности ресурсов вызвано рядом причин. По техническим причинам в системе SILK невозможно задать транзитивный запрос на выборку ресурсов с произвольным уровнем вложенности. Поэтому используется

запрос, в котором уровень вложенности фиксирован (ограничен 4 уровнями) и в котором явно перечисляются условия на выборку ресурсов из категории *category:Mathematics*. Можно также отметить ряд проблем, связанных с транзитивностью системы категорий DBpedia.

Иерархия категорий DBpedia является несовершенной и в ней часто нарушается принцип транзитивности, то есть ресурс, относящийся к достаточно вложенной подкатегории некоторой категории, к самой категории может и не относиться.

Например, категория *category:Algebra* содержит вложенную подкатегорию *category:Museu_Picasso* в иерархии Algebra → Convex_geometry → ... → Cubes → Cubism → Pablo_Picasso → Museu_Picasso.

Более того, некоторые категории содержат очень общие подкатегории (Space, Time, Structure), и поэтому, на очень глубоком уровне вложенности содержат почти все ресурсы DBpedia.

Онтология OntoMath^{pro} содержит 3450 классов. В результате процесса связывания классов онтологии с ресурсами DBpedia были найдены 842 связи. Число связанных классов составило 828 (некоторые классы были связаны сразу с несколькими ресурсами DBpedia).

Для оценки полноты связывания (при ограничении до 4 уровней иерархии) была выполнена оценка количества ресурсов, участвующих в связывании при увеличении уровней иерархии до 5-8. В результате не была найдена ни одна дополнительная связь, что обосновывает выбранное ограничение глубины, которое существенно не влияет на полноту связывания.

Оценки точности связывания составляют 95% (примерно 43 связи (5%) были оценены как некорректные или не совсем корректные). Эти некорректные и не совсем корректные связи возникли по следующим причинам:

- неточные ссылки на онтологии (ссылка на статью Википедии для связанных или более общих понятий). Например, #E203 (Компактный слой) ≠ dbpedia:Novikov's_compact_leaf_theorem (Теорема о компактном слое), или пример связи с более общим понятием: #E3004 (Признак Лейбница сходимости знакопередающегося ряда) ≠ dbpedia:Alternating_series (Знакопередающийся ряд);

- некорректные интервики (ссылки между языковыми разделами) в Википедии. Например, #E1317 (Сравнение по модулю натурального числа) ≠ dbpedia:Modular_arithmetic (Модульная арифметика);

- наличие омонимичных статей в пределах четвертого уровня вложенности, например, #E1408 (Отображение спаривания) ≠ dbpedia:Mating (Сексуальное спаривание), которое попало в категорию *category:Statistics* через следующую цепочку подкатегорий: *category:Fertility* → *category:Demography* →

category:Fields_of_application_of_statistics → *category:Statistics*.

Заключение

Публикация данных в пространстве Linked Open Data позволяет каталогизировать данные, повысить ценность данных за счет связывания их с другими данными, понизить степень дублирования данных, облегчить доступ к данным заинтересованным сторонам. Процесс связывания данных выполняется специальными инструментами связывания, которые в интерактивном режиме позволяют формировать различные схемы связывания данных, осуществлять настройку методов связывания, в том числе управляя ограничениями на условия связывания данных.

В результате проделанной работы были выявлены ряд важных ограничений на связывание данных с ресурсами DBpedia, получены оценки точности и полноты связывания данных с данным ресурсом, изучены основные технологические решения систем связывания SILK и LIMES. Проведенные эксперименты выявили также ряд особенностей математического RDF-набора, затрудняющего эффективное связывание, что позволило наметить пути улучшения качества построенных решений.

Благодарности

Работа выполнена при финансовой поддержке РФФИ, грант № 11-07-00507а.

Библиографический список

[Невзорова и др., 2012] Невзорова О.А., Жильцов Н.Г., Заикин Д.А., Жибрик О.Н., Кириллович А.В., Невзоров В.Н., Бирыльцев Е.В. Прототип программной платформы для публикации семантических данных из математических научных коллекций в облаке LOD // Ученые записки КГУ. Серия Физико-математические науки. 2012. В печати.

DATA BINDING TECHNOLOGY IN THE SPACE OF OPEN DATA ON THE EXAMPLE OF MATHEMATICAL COLLECTIONS

Nevezorova O.A., ** Kirillovich A.V. *

* Kazan Federal University, Kazan, Russia
al.kirillovich@gmail.com

** Research Institute of Applied Semiotics of the Academy of Sciences of Tatarstan Republic, Kazan Federal University, Kazan, Russia
onevzoro@gmail.com

The paper is presented the software technologies of data binding in the space of Open data with use of LIMES and SILK systems, and experiments on the binding of mathematical RDF-dataset.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ИССЛЕДОВАТЕЛЬСКИЙ ПОРТАЛ ПО МОДЕЛИРОВАНИЮ ИНФОРМАЦИОННЫХ СИСТЕМ

Ланин В.В., Лядова Л.Н.

Пермский филиал федерального государственного автономного образовательного учреждения высшего профессионального образования "Национальный исследовательский университет "Высшая школа экономики", г. Пермь, Россия

lanin@perm.ru

lnlyadova@mail.ru

Статья посвящена описанию разработки портала, ориентированного на поддержку работы исследователей (преподавателей, аспирантов, студентов), занимающихся вопросами моделирования информационных систем в различных предметных областях, создания и использования инструментальных средств разработки, основанных на (мета)моделировании (применении DSM, DSL и др.). В данной работе представлено описание архитектуры портала, средств информационного поиска и управления документами, создания единой системы документов, относящихся к данной области исследований.

Ключевые слова: портал; онтологии, моделирование информационных систем.

ВВЕДЕНИЕ

В настоящее время не существует единого информационного ресурса, ориентированного на поддержку работы исследователей, занимающихся вопросами моделирования информационных систем в различных предметных областях, хотя проекты по данному направлению выполняются в Санкт-Петербургском государственном университете (под руководством профессора Терехова А.Н.), Южном федеральном университете (руководитель – профессор Рогозов Ю.И.) и др. Информация об исследованиях размещается на сайтах отдельных организаций (кафедр университетов и пр.), фирм-разработчиков, имеются порталы, посвященные отдельным проектам, проводимым по данной тематике конференциям (<http://www.metacase.com/>, <http://www.eclipse.org/>, <http://www.omg.org/>, <http://www.dsm-conference.org/> и др.). Поиск нужной информации (особенно для молодых исследователей) затруднен: с одной стороны, имеется множество ресурсов, с другой – многие из них оказываются бесполезными, не отвечающими поисковым запросам пользователей. Например, при поиске по слову «метамоделирование» этот термин появляется в сочетании с такими понятиями как «нейролингвистическое программирование», «психология», «гипноз» и т.д.

Цель создания портала в НИУ ВШЭ – создание «саморазвивающегося» ресурса, предоставляющего

в распоряжение пользователей средства интеллектуального поиска и автоматизированной обработки полученных результатов (документов, источников), удобные средства навигации по найденным ресурсам. Реализация основана на использовании онтологий, описывающих как предметную область, так и обрабатываемые документы.

Пользователи портала должны получить возможность настройки этих средств в соответствии со своими потребностями, а также возможность оперативного взаимодействия, обмена информацией и публикации результатов своих исследований для обсуждения, коллективной работы над проектами.

Для работы с моделями должен быть разработан специализированный язык (DSL) с использованием созданных при выполнении проекта средств. При реализации проекта по созданию портала решаются задачи, связанные с организацией коллективной работы, поиском, сбором и анализом материалов и их публикацией.

Задачи подобного характера решались и ранее другими исследователями. Особенно хотелось бы отметить работы [Загорулько, 2004], [Загорулько, 2008], [Мальцева, 2008]. Новизна представленной работы заключается в комплексном подходе к разработке портала, интегрирующем возможности информационных технологий и систем различного назначения на основе знаний о предметной области системы.

При работе с порталом пользователи получают эффективные интеллектуальные средства поиска информации на основе семантической индексации, автоматической классификации и каталогизации найденных документов с построением семантических связей между ними и автоматического реферирования документов с использованием знаний. Эффективность работы с электронными документами предполагается значительно увеличить за счет их интеллектуального анализа, для которого применяются агентный и онтологический подходы [Ланин, 2009а].

1. Концепция портала

Ключевой идеей функционирования разрабатываемого портала является адаптируемость к потребностям пользователя. При создании портала необходимо учитывать не только потребности пользователей, но и особенности представления знаний в Интернет. Портал должен не только обеспечить более быстрый доступ к информационным ресурсам, но и предоставить в распоряжение пользователей дополнительные возможности по организации научных исследований и совместной работы над проектами. Таким образом, необходимо реализовать инструментарий, обеспечивающий автоматизацию трудоемких операций по поиску и анализу данных, разработке моделей инновационного развития и их апробации и пр. Многомерная классификация и удобная каталогизация ресурсов, наличие средств навигации, настраиваемых в соответствии с запросами пользователей, – еще одно требование к portalу.

В качестве основы исследовательского портала должно быть создано информационное ядро дисциплины, которую он представляет. На основе этого ядра, предоставляющего исследователям базовую информацию и набор сервисов, должны развиваться информационные ресурсы, создаваемые пользователями портала. Один из фундаментальных принципов – открытость в сочетании с защищенностью. Портал должен быть построен как «самоподдерживающийся» ресурс: развитие портала, расширение его ядра и наполнение новыми ресурсами, как и управление, ложится на самих пользователей.

При создании портала учитываются возможности современных информационных технологий, полезные с точки зрения решаемых задач. Перспективными тенденциями развития Интернет считаются концепция Web 2.0 и технология Semantic Web [O'Reilly, 2005], [Berners-Lee, 2001]. Вытесняющий традиционные Web-сервисы Web 2.0 не является технологией или специальным стилем Web-дизайна – его следует рассматривать как комплексный подход к организации, реализации и поддержке Web-ресурсов. Наиболее характерными технологиями, реализующими подход Web 2.0, являются wiki-

ресурсы, блоги, технология FOAF, технологии синдикации новостей и пр.

Блоги (сетевые дневники) представляют собой один из самых ярких примеров использования принципов Web 2.0. Значительная часть Web-контента создается пользователями, а не владельцами ресурса. Для этого активно используют технологии RSS и FOAF, характерные для Web 2.0. Используются также тэги (tags, метки) для тематического структурирования контента. Технология FOAF (Friend Of A Friend) является одной из важнейших составляющих социальных Интернет-сетей. Пользователю предоставляется возможность подписаться на новости и материалы тех пользователей, которые находятся в так называемом «списке друзей». Этим самым поощряется общение пользователей Сети. Технология RSS (Really Simple Syndication) – это простая и эффективная технология экспорта гипертекста, используемая для создания новостных лент. Эта технология, как и другие технологии Web 2.0, основана на использовании языка разметки XML.

Все эти возможности используются при создании портала для организации эффективной работы пользователей, их оперативного взаимодействия в ходе исследований.

2. Архитектура портала

Архитектура портала основана на создании многоуровневых моделей, управляющих работой системы. Фактически программное обеспечение работает в режиме интерпретации этих моделей, что обеспечивает максимальные возможности адаптации системы через внесение изменений в модели в процессе эксплуатации системы [Лядова, 2008]. Упрощенная структура моделей портала показана на рис. 1. Предметно-зависимые модели – это модели онтологического уровня.

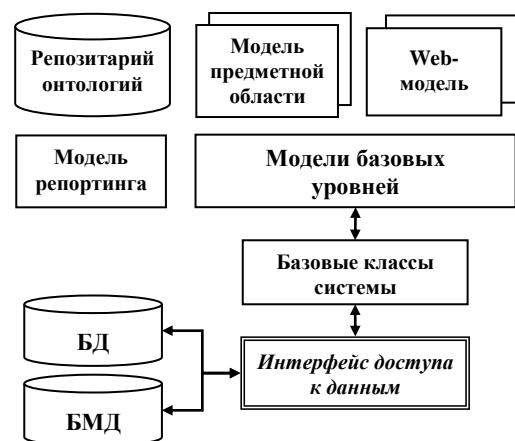


Рисунок 1 – Общая схема работы поисковой системы

Модели базовых уровней позволяют описать «ядро» портала, базовые понятия его предметной области, а также сгенерировать и настроить интерфейс пользователя и сформировать базу

данных портала. Портал функционирует в режиме интерпретации этих моделей. Базовые модели являются расширяемыми, на их основе могут быть созданы новые предметно-ориентированные (онтологические) модели. Для их создания разрабатываются специальные предметно-ориентированные языки. Пользователи могут вносить изменения в модели в процессе функционирования системы, настраивая ее на свои потребности.

Программные компоненты системы работают с моделями своих уровней (языковой инструментарий, средства реструктуризации данных, средства интеллектуального поиска, анализа и каталогизации документов, средства репортинга, средства анализа данных и моделей, подсистема безопасности и пр.). Система допускает расширение функциональности через подключение новых программных компонентов.

3. Описание документа с помощью онтологии

Процесс поиска информации с помощью поисковой системы может быть описан следующим образом [Ланин, 2009b]. У пользователя возникает информационная потребность (необходимость найти сведения по какому-либо вопросу). Затем пользователь некоторым образом формализует свою информационную потребность в виде запроса (в традиционных системах это выделенное множество ключевых слов с зафиксированными отношениями между ними). На следующем этапе через интерфейс поисковой системы вводится запрос. Система на множестве документов, являющемся информационно-поисковым пространством, осуществляет выборку документов, которые по внесенным в систему критериям соответствуют запросу пользователя, и формирует результат (отклик). Найденные документы по своему содержанию делятся на две группы: документы, соответствующие информационной потребности пользователя (релевантные), и документы, не соответствующие его информационной потребности, но соответствующие запросу пользователя с точки зрения информационно-поисковой системы (информационный шум).

Учитывая специфику решаемой задачи, процесс поиска информации может быть улучшен по двум направлениям: релевантности результата и представлению отклика. Обе задачи предлагается решать с помощью онтологического подхода, завоевывающего все большую популярность.

Модифицированная схема поиска, реализуемая при создании портала, представлена на рис. 2.

Основная особенность предлагаемого подхода – использование репозитория онтологий на этапах преобразования запроса и документа. Откликом является структурированный документ, т.е. документ, в котором выделены понятия онтологий.



Рисунок 2 – Общая схема работы поисковой системы

О структуре репозитория онтологий рассказано в следующем разделе.

4. Подсистема управления документами портала

Методы искусственного интеллекта, как правило, используются для решения трудно формализуемых задач, постановка которых проста и понятна для человека, но при разработке алгоритмов их решения возникают трудности. Одна из таких задач – работа с документами в информационных системах: их поиск и каталогизация, анализ и извлечение информации.

В настоящее время существуют различные подходы, модели и языки, ориентированные на интегрированное описание данных и знаний. Наиболее перспективным и универсальным, по мнению авторов, представляется онтологический подход.

Согласно общепринятому определению, под онтологией (в широком смысле) понимается база знаний специального типа, которая может «читаться» и пониматься, отчуждаться от разработчика и/или физически разделяться ее пользователями. Учитывая специфику решаемых в данной работе задач, можно конкретизировать понятие онтологии: онтология – это спецификация некоторой предметной области, которая включает в себя словарь терминов (понятий) предметной области и множество связей между ними, которые описывают, как эти термины соотносятся между собой [Ланин, 2009b].

Для построения иерархии понятий онтологии используются следующие базовые типы отношений: “is_a” («класс – подкласс», гипонимия); “part_of” («часть – целое», меронимия); “synonym_of” (синонимия). Следует учесть, что данные типы отношений являются базовыми и не зависят от онтологии, но необходимо предоставить

пользователю возможность добавления новых отношений, которые бы учитывали специфику описываемой предметной области.

В представленном подходе выделяются три типа онтологий:

- онтология предметной области конкретной информационной системы (ИС);
- онтология как база знаний (БЗ) интеллектуального агента;
- онтология как описание документа.

Рассмотрим назначение каждого из перечисленных типов онтологий.

Онтологии предметной области имеют наиболее типичное применение, они используются для описания понятий предметной области ИС. Например, школьное образование, социальная помощь гражданам или инновационное развитие регионов. В онтологии этого типа описывается связь понятий, языковые единицы для их выражения, аксиомы предметной области. Онтология предметной области используется для семантического индексирования и анализа всех документов системы.

Для анализа документов используется мультиагентный подход. Интеллектуальные агенты, руководствуясь онтологией как базой знаний (второй тип онтологий), производят поиск и анализ конкретных понятий документа. Каждая из вершин такой онтологии имеет определенный прототип, интерпретация которого известна агенту. Таким образом, агент использует онтологию как определенную программу своих действий. Вершинами онтологии данного типа могут являться понятия из онтологии предметной области.

Третий тип онтологий используется для описания структуры и содержания документов. Этот тип онтологий включает в себя два класса (плоскости) вершин. К первому классу относятся вершины, описывающие структуру документа. Например: таблица, дата, должность и т.д. (они представляют собой общие понятия, не зависящие от конкретной предметной области). Другим типом будут являться вершины, содержащие понятия документа. Первый тип вершин будем называть структурные вершины, второй тип – семантические вершины. Благодаря такому подходу из документа можно получить требуемые данные: известно, где искать данные и как они могут быть интерпретированы.

Если представлять документ с использованием онтологий, то задача сопоставления онтологии и анализируемого документа сводится к задаче поиска понятий онтологии в документе. Как следствие, системе необходимо ответить на вопрос: описывает ли данная онтология документ или нет. На последний вопрос можно ответить утвердительно, если в процессе сопоставления в документе были найдены все понятия, включенные в онтологию.

Таким образом, исходная задача сводится к задаче поиска в тексте документа общих понятий на основе формальных описаний. На основе онтологии может быть получен фрейм, слоты которого заполняются в процессе анализа документа. В качестве слотов фрейма выступают понятия онтологии, а значения этих фреймов заполняются данными анализируемого документа. Таким образом из найденного неструктурированного документа может быть получен структурированный документ-фрейм.

Онтологии располагаются на трех уровнях репозитория. На первом уровне расположены онтологии, описывающие объекты, используемые в конкретной системе и учитывающие ее особенности. На втором уровне описываются объекты, инвариантные к предметной области. Объекты третьего уровня описывают наиболее общие понятия и аксиомы, с помощью которых описываются объекты нижележащих уровней.

Помимо описанных выше онтологий в процессе работы портала используются дополнительные онтологии: онтология источников информации и онтология форматов электронных документов. На данный момент в онтологии источников детально представлены ресурсы сайта <http://www.dsmforum.org>. Также в онтологии представлены конференции и другие мероприятия, проводимые по тематике моделирования информационных систем, блоги разработчиков инструментальных средств MetaCase и Microsoft Visual Studio. Онтология ресурсов портала базируется на понятиях Дублинского ядра метаданных. Онтология форматов документов используется для унификации обработки документов в различных форматах. Онтологии описаны на языке OWL 2.0 с помощью редактора Protégé 4.2.

5. Реализация подсистемы управления документами портала

Основными обрабатываемым на портале объектами будут являться электронные документы, поэтому подсистема управления документами является крайне важным компонентом. При проектировании портала принято решение использовать одну из существующих ECM-систем.

Согласно последнему опубликованному отчету агентства Gartner [Weintraub, 2011] по ECM системам в знаменитый магический квадрант попала единственная Open Source из всех представленных система Alfresco. Gartner отметили инновационный подход [Gilbert, 2012] – именно поэтому Alfresco располагается в правом нижнем углу. Также Alfresco Software попала и в отчет агентства Forrester. Согласно Forrester, Alfresco может составить реальную альтернативу таким крупным игрокам рынка ECM систем, как IBM, Oracle, Open Text и EMC [Weintraub, 2011].

Пользователи могут начать использовать Alfresco в их повседневной работе сразу после установки,

однако это не позволит использовать преимущества, предоставляемые данной системой – одним из основных является гибкая система управления контентом.

Модель содержимого в Alfresco включает в себя 5 основных составляющих перечисленных ниже [Potts, 2008].

Пользовательские типы (custom types) предназначены для создания таксономии содержимого репозитория. С их помощью пользователь определяет типы документов, которые будут содержаться во внутреннем хранилище Alfresco. При этом пользовательский тип может обозначать не только конкретный тип документа, но и обозначать некоторую абстрактную категорию документов (например, «финансовый документ»), что позволяет гибко настраивать модель содержимого под нужды конкретной организации. Кроме того, для пользовательских типов определен механизм наследования, с помощью которого конкретные типы документов могут наследовать свои свойства от более абстрактного.

Однако обычно для создания полноценной модели содержимого конкретной организации зачастую бывает недостаточно использования только одних пользовательских типов. Часто пользователю требуется сопоставить с каждым из пользовательских типов некоторый набор метаданных, качественно характеризующий данный конкретный тип. Для этого в Alfresco используются *свойства (properties)*. При определении нового свойства пользователь должен указать его имя и используемый тип данных. Работа со свойствами в Alfresco напоминает работу с полями классов в ООП. Однако в последнем случае пользователю необходимо программировать всю бизнес-логику самостоятельно, в то время как Alfresco позволяет декларативно описывать ограничения, накладываемые на значения свойств.

Alfresco предоставляет возможность задания следующих *ограничений (constraints)*: ограничение на минимальное и максимальное значение свойства, ограничение на длину значения свойства, ограничение, контролирующее количество и вид элементов в случае использования типа данных «список», ограничение, задаваемое с помощью регулярных выражений.

Помимо определения таксономии документов организации и набора их метаданных, зачастую бывает необходимо определять связи между различными типами документов. Для этого в Alfresco используются *ассоциации (association)*, которые можно разделить на два класса.

1) Простые ассоциации, с помощью которых пользователь указывает, что два конкретных пользовательских типа связаны между собой.

2) Ассоциации, определяющие «родственные» связи между двумя пользовательскими типами (child associations). Ассоциации данного типа налагают

дополнительное ограничение, предполагающее, что связанный документ зависит от документа, с которым он связан. Т.е., например, при удалении родительского документа, дочерний также должен быть удален. В Alfresco есть встроенная ассоциация данного типа – Contains, которая используется для определения связи между пользовательскими типами, выступающим в качестве контейнера, например, «Каталог», и его содержимым.

С помощью использования *аспектов*, пользователь может расширить набор свойств отдельных пользовательских типов. Зачастую возникает необходимость, сделать доступными извне только некоторые документы. Для этого необходимо определить набор дополнительных свойств, с помощью которых Alfresco могла бы определять, должен ли быть доступен данный документ извне, или нет. Использование аспектов напоминает использование интерфейсов в объектно-ориентированном программировании. С их помощью пользователь может добавить ряд дополнительных свойств к отдельным типам, не меняя иерархию в целом.

Microsoft SharePoint – инструмент для создания сайтов, предоставляющих пользователям возможность для совместной работы. Создаваемые на платформе SharePoint сайты могут быть использованы в качестве хранилища информации, знаний и документов, а также использоваться для исполнения облегчающих взаимодействие веб-приложений, таких как вики и блоги.

Модель данных SharePoint 2010 состоит из перечисленных ниже понятий [Perran, 2010].

Столбцы (columns) представляет собой набор сведений, предназначенный для совместного использования различными пользователями.

Библиотеки (libraries) предназначены для хранения контента организации. Каждая библиотека отображает список документов и некоторый набор сведений о них, которые помогают пользователям в работе. Пользователи могут управлять отображением документов, а также контролировать доступ к ним, ограничивая круг пользователей, которым разрешается просматривать документы до их утверждения. Кроме того, библиотеки являются средством контроля версий: с их помощью возможно одновременно работать с различными версиями одного и того же документа. Содержимое библиотеки, также как и содержимое списка, определяется ее типом.

Столбец (lists) представляет собой атрибут метаданных, используемый для описания отдельных элементов в списке или библиотеке.

Типы контента (content types) используются для описания свойств элементов списков и библиотек. С помощью типов контента пользователь может, например, определить таксономию документов в библиотеке документов. Поддерживаются следующие типы контента: метаданные или

свойства, пользовательские формы, рабочие процессы, шаблоны документов.

В результате сравнения принято решение при разработке портала использовать Alfresco. Немаловажными факторами при выборе системы стали открытость исходного кода и использование открытых стандартов. Кроме того, система Alfresco реализована на языке Java, что значительно упрощает интеграцию с компонентами обработки онтологий и инструментами Semantic Web, так же реализованными на данном языке. SharePoint в свою очередь ориентирован на корпоративный сегмент и решения Microsoft, обладает ограниченными возможностями интеграции семантических технологий.

Заключение

Применение описанных подходов при построении портала существенно снижает трудоемкость поиска необходимой информации, ее анализа и возможности использования в исследованиях. Полученная в результате анализа документов информация может использоваться исследователями для усовершенствования моделей предметной области, построенных ими. Таким образом, появляется основа для создания интеллектуальной системы с высокой степенью обратной связи. Ориентация на знания является базовым механизмом функционирования портала, что позволяет комплексно решать поставленные задачи.

Работа выполнена при поддержке Программы «Научный фонд НИУ ВШЭ» софинансирования грантов РФФИ и РГНФ (проект № 12-09-0102).

Библиографический список

[Загорюлько, 2004] Загорюлько Ю.А., Булгаков С.В. Использование онтологий для построения инновационных цепочек в системе поддержки инновационной деятельности в регионе // Труды VI-й международной конференции «Проблемы управления и моделирования в сложных системах». Самара: Самарский Научный Центр РАН, 2004. С. 328–333.

[Загорюлько, 2008] Загорюлько Ю.А. Автоматизация сбора онтологической информации об Интернет-ресурсах для портала научных знаний // Известия Томского политехнического университета / Томск: Томский политехнический университет, 2008. Т. 312. № 5. С. 114–119.

[Ланин, 2009а] Ланин В.В. Методы и средства решения задач информационного поиска для системы поддержки научных исследований // Инновационное развитие регионов: методы оценки и поддержка исследований: межвуз. сб. науч. статей / Перм. гос. ун. т. – Пермь, 2009. С. 80–88.

[Ланин, 2009б] Ланин В.В. Решение задач информационного поиска для исследовательского портала на основе агентного и онтологического подходов // Инновационное развитие регионов: методы оценки и поддержка исследований: межвуз. сб. науч. статей / Перм. гос. ун. т. – Пермь, 2009. С. 89–96.

[Лядова, 2008] Лядова Л.Н. Метамоделирование и многоуровневые метаданные как основа технологии создания адаптируемых информационных систем // Advanced Studies in Software and Knowledge Engineering / International Book Series “Information Science & Computing”, Number 4. Volume 2, 2008. Institute of Information Theories and Applications FOI ITHEA, Sofia, 2008. P. 125–132.

[Лядова, 2009] Лядова Л.Н. О подходе к построению исследовательского портала на основе метамоделирования //

Инновационное развитие регионов: методы оценки и поддержка исследований: межвуз. сб. науч. статей / Перм. гос. ун. т. – Пермь, 2009. С. 74–79.

[Мальцева, 2008] Мальцева С.В., Проценко Д.С. Серверы отношений сетевых сообществ практики на основе онтологических моделей // Автоматизация и современные технологии. №3, 2008. Научно-техническое издательство «Машиностроение». С. 26–29.

[Мальцева, 2008] Мальцева С.В. Применение онтологических моделей для решения задач идентификации и мониторинга предметных областей // Бизнес-информатика, №3(05), 2008. С. 18–24.

[Berners-Lee, 2001] Berners-Lee T., Hendler J., Lassila O. The Semantic Web // Scientific American. Vol. 284, No. 5, 2001. P. 35–43.

[Potts, 2008] Potts J. Alfresco Developer Book. Customizing Alfresco with actions, web scripts, web forms, workflows, and more. Packt publishing, 2008

[Gilbert, 2012] Gilbert M. R., Shegda K. M., Chin K., Tay G., Koehler-Kruener Hanns Gartner’s Magic Quadrant for Enterprise Content Management. 18 October 2012

[Weintraub, 2011] Weintraub A. The Forrester Wave™: Enterprise Content Management, Q4 2011 Alan Weintraub November 1, 2011

[Perran, 2010] Perran A., Perran S., Mason J., Rogers L.- Beginning SharePoint 2010 - Building Business Solutions with SharePoint - 2010

[O’Reilly, 2005] O’Reilly T. What Is Web 2.0 [Электронный ресурс] [<http://oreilly.com/web2/archive/what-is-web-20.html>].

RESEARCH PORTAL OF MODELING INFORMATION SYSTEMS

Lanin V.V., Lyadova L.N.

*National Research University Higher School of
Economics, City of Perm*

lanin@perm.ru

lnlyadova@mail.ru

The paper describes the development of a portal about development and use of tools based on the (meta) modeling (using DSM, DSL, etc.). The architecture of a portal, information retrieval subsystem and document management are described.

The purpose of the portal is the creation of "self-developing" resource, which provides intelligent search and automatic processing of the results (documents and sources), easy navigation on the found resources. Implementation is based on the ontologies approach.

The main feature of suggested methods is an integrated approach to development. The approach bases on a multi-level ontology repository. The portal allows searching and analyzing information, creating and researching model, publishing research results. Software gives an opportunity of a flexible customizing. The main topic of this paper is an intelligent information search means based on semantic indexation, automatic document classification, tracking of semantic links between documents and automatic summarization.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПРАКТИЧЕСКИЕ АСПЕКТЫ РЕАЛИЗАЦИИ ПОИСКОВЫХ ИНТЕРФЕЙСОВ СЕМАНТИЧЕСКИХ ПРИЛОЖЕНИЙ LINKED DATA

Галушка И.Н.* , Завгородний В.В.** , Солошич С.Н.* , Щербак С.С.*

** Кременчугский национальный университет имени Михаила Остроградского,
г. Кременчуг, Украина*

sergey.shcherbak@gmail.com

***Днепропетровский государственный технический университет
г. Днепропетровск, Украина*

Рост популярности концепции связанных данных наряду с описанием информации в виде триплетов RDF обусловил необходимость исследования процедур взаимодействия с хранилищами триплетов. В результате разработан универсальный поисковый интерфейс, обеспечивающий визуальное построение запросов к хранилищам триплетов. Визуальные запросы автоматически преобразуются в язык запросов SPARQL, который используется для доступа к хранилищам триплетов. После выполнения запроса пользователь получает контекст с триплетами, соответствующими искомым с учетом заданных ограничений на предикаты и объекты.

Ключевые слова: поиск; связанные данные; шаблон реализации; триплет

ВВЕДЕНИЕ

Широкое развитие стандартов и поддержка концепции Linked Data крупными ИТ компаниями определяет тенденции развития будущего WWW как глобальной базы данных, в которой можно через специализированные поисковые интерфейсы получать доступ к структурированным представлениям данных, документов, распределенных по Web, для решения задач поиска и подобных.

В рамках Linked Data информация описывается в терминах языка RDF (англ. Resource Description Framework), а именно, в виде триплетов, троек вида «субъект-предикат-объект» или квадов (quad) – поименованных графов вида «граф-субъект-предикат-объект». Далее, если это не будет приводить к противоречию, будем использовать понятие «триплет» как для понятия «триплет», так и для понятия «квад».

Модель данных RDF предполагает распределенное хранение объектов и их схем, при их наличии, на различных web-серверах с интегрированным или внешним специализированным хранилищем триплетов (Triplestore). Для доступа к хранилищам триплетов используется протокол и язык запросов SPARQL (англ. SPARQL Protocol and RDF Query Language), который является в некотором смысле аналогом

языка SQL, применяемого для обработки данных реляционных баз. Использование подобных языков запросов подразумевает наличие специализированных знаний для эффективного их применения, что не способствует ни популяризации языка SPARQL, как средства извлечения данных, ни увеличению распространенности хранилищ триплетов.

В данной работе предлагается подход к увеличению эффективности поиска на основе хранилищ триплетов путем уменьшения сложности процедур динамического формирования запросов на языке SPARQL и применения шаблонов запросов для генерации пользовательского поискового интерфейса.

1. Поисковые пользовательские интерфейсы к хранилищам триплетов

Поиск информации в рамках Linked Data основывается на использовании данных, представленных в виде объектов, принадлежащих к некоторой предметной области. В рамках такого подхода содержимое документа представляется как совокупность объектов, объединенных некоторым контекстом. Для определения контекста объектов будем использовать понятие «граф» квада. Тогда поиск информации сводится к идентификации по заданным критериям (пользовательским ограничениям) триплетов графа и выводу

контекстов с триплетами, соответствующих идентифицированным.

Для создания эффективных поисковых интерфейсов, «дружественных» к пользователю, необходима разработка типовых решений – паттернов реализаций, которые могли бы позволить визуально конструировать запросы к поименованным графам хранилища триплетов и не требовала бы от разработчика или пользователя программных систем специализированных знаний о языке SPARQL.

На примере веб-приложения рассмотрим архитектуру классического решения на основе хранилища триплетов и языка запросов SPARQL (рис.1).

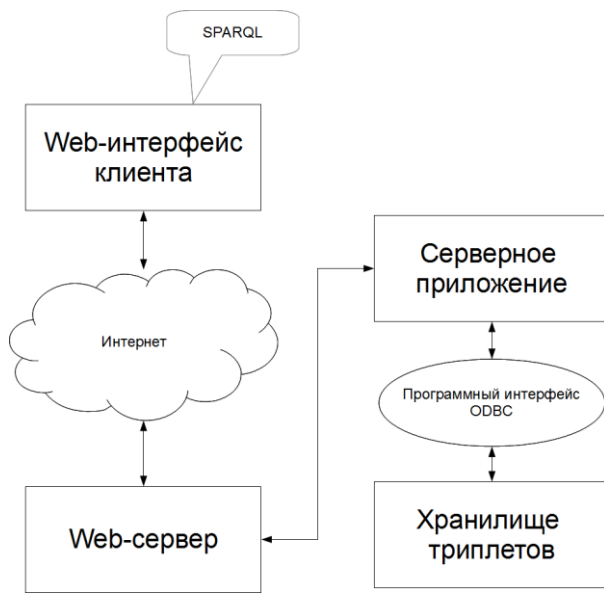


Рисунок 1 – Архитектура веб-приложения на основе хранилища триплетов

Запрос пользователя на языке SPARQL передается от веб-приложения к хранилищу триплетов, где запрос выполняется, а результаты отсылаются обратно клиенту.

Ключевая особенность классического решения в необходимости формирования зачастую в ручную запроса на языке SPARQL, что требует не только специализированных знаний о языке SPARQL, но и знаний о структуре объектов, содержащихся в хранилище триплетов.

Одним из приложений, реализующих подобную функциональность является веб-интерфейс isql компании OpenLink, применяемый в Dbpedia, который наряду с универсальностью доступа к данным, отличается стабильностью работы.

Рассмотрим архитектуру предлагаемого решения, исключая взаимодействие пользователя с хранилищем триплетов через SPARQL (рис.2).

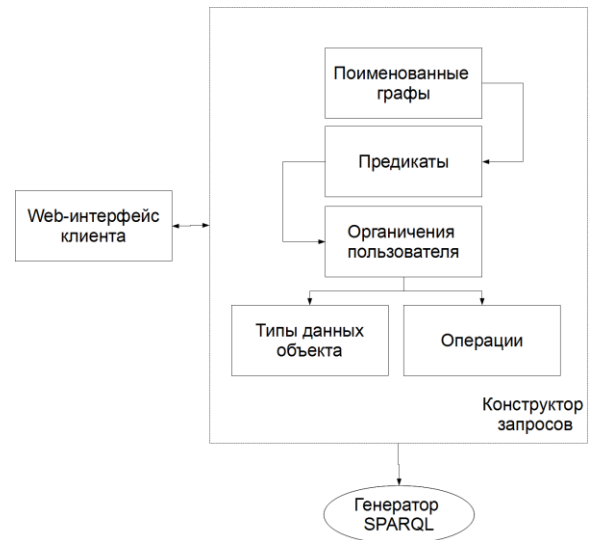


Рисунок 2 – Архитектура веб-интерфейса к хранилищу триплетов на основе конструктора запросов SPARQL

В рамках такого решения пользователь через веб-интерфейс получает доступ к перечню поименованных графов хранилища триплетов, выбор из которого позволит автоматически получить информацию о топологии графа, а именно о составляющих его предикатах и типах данных объектов, что может быть использовано для формирования поискового запроса путем установления пользовательских ограничений (фильтров) на выводимые данные с хранилища триплетов.

Ограничения пользователя определим как набор правил с условиями вывода объектов поименованного графа, типы которых определяют семантику применяемых операций сравнения. В качестве основы для условий вывода определим перечень операций, а именно «=» (равно), «>» (больше), «<» (меньше), которые могут быть применены к объектам.

Семантика операций «=», «>», «<» определяется типом объектов поименованного графа, т.е. каким именно образом будет реагировать конструктор запросов на символьные имена операций «=» «>» «<» определяется типом объектов поименованного графа.

Для целочисленного типа объекта (xsd:integer) операции «=», «>», «<» имеют следующий смысл:

«=» – сравниваемые объекты равны;

«>» – объект, с которым проводится сравнение больше сравниваемого;

«<» – объект, с которым проводится сравнение меньше сравниваемого;

Для строкового типа (xsd:string) под действиями «=», «>», «<» будем подразумевать следующее :

«=» – сравниваемые объекты равны;

«>» – сравниваемый объект является подстрокой объекта, с которым проводится сравнение;

«<>» – объект, с которым проводится сравнение является подстрокой сравниваемого;

Действия «=», «>», «<» для объектов типа «xsd:float» (число с плавающей точкой) имеют такую же семантику, как и соответствующие операции с целочисленным типом объектов.

Таким образом, пользователь может формировать правила вывода содержимого того или иного поименованного графа хранилища триплетов. На основе полученных правил может быть автоматически сформирован типовой SPARQL-запрос к хранилищу триплетов, практические аспекты создания которого рассмотрены в разделе 2.

2. Практические аспекты реализации поискового пользовательского интерфейса к хранилищу триплетов

Архитектура веб-интерфейса к хранилищу триплетов на основе конструктора запросов SPARQL (рис. 2) подразумевает выполнение набора предопределенных действий с помощью SPARQL для динамического получения топологии поименованного графа. К таким действиям относятся:

- получение списка поименованных графов;
- получение списка предикатов в поименованном графе;
- получение типа данных предиката.

Для получения списка всех имеющихся поименованных графов в хранилище триплетов можно использовать следующий запрос на языке SPARQL:

```
SELECT distinct ?g WHERE
{ GRAPH ?g { ?s ?p ?o } }
```

Для получения конкретного поименованного графа или группы, имя которых соответствует некоторому критерию можно использовать следующий запрос на языке SPARQL:

```
SELECT distinct ?g WHERE {
  GRAPH ?g
  { ?s ?p ?o . Filter(regex(?g,
<http://shcherbak.net/>))
}}
```

где из перечня всех графов выбираются те, у которых `http://shcherbak.net/` встречается в имени графа;

Для получения списка предикатов в поименованном графе можно использовать следующий запрос на языке SPARQL:

```
SELECT distinct ?p
FROM <http://shcherbak.net/User>
{ ?s ?p ?o }
```

где из графа `<http://shcherbak.net/User>` будут выбраны все уникальные предикаты и возвращены в виде набора в качестве результата выборки.

Для получения типа предиката можно использовать следующий запрос на языке:

```
SELECT datatype(?o)
FROM <http://shcherbak.net/User>
WHERE {
  ?s ns:date_of_ birthday ?o }
```

где для предиката `ns:date_of_birthday` графа `http://shcherbak.net/User` будет получен тип объекта.

На основе подобных запросов можно строить произвольные запросы на выборку с поименованных графов, например:

```
sparql SELECT ?s ?p ?o
FROM <http://shcherbak.net/User>
WHERE { ?s ?p ?o FILTER ( ?s = ns:2 ) }
```

где с графа `http://shcherbak.net/User` будут выбраны триплеты пользователя `ns:2`.

Кроме того, в качестве ограничений пользователя могут выступать языковые теги (`lang tags`). В этом случае, для вывода информации с графа `http://shcherbak.net/User` можно установить фильтр на вывод объектов на русском языке, у которых языковой тег установлен в значение «ru»:

```
SELECT ?name ?second_name
FROM <http://shcherbak.net/User>
WHERE {
  ?s ns:name ?name.
  ?s ns:second_name ?second_name.
  FILTER ( ?name = 'Иван'@ru && ?second_name =
'Иванов'@ru ) }
```

Для поиска объекта с учетом нескольких пользовательских ограничений можно использовать следующий SPARQL-запрос:

```
PREFIX ns:<http://shcherbak.net/>
SELECT ?o
FROM <http://shcherbak.net/User>
WHERE
{ ?s ns:name ?o
  FILTER regex(?o, "Иван", "i" )
  FILTER regex(?o, "Петров", "i" ) }
```

Реализация поискового интерфейса на основе вышеприведенных запросов (рис.3) предоставляет пользователю вспомогательную информацию, которая может значительно облегчить процедуру составления запроса к хранилищу триплетов и уменьшить время его составления.

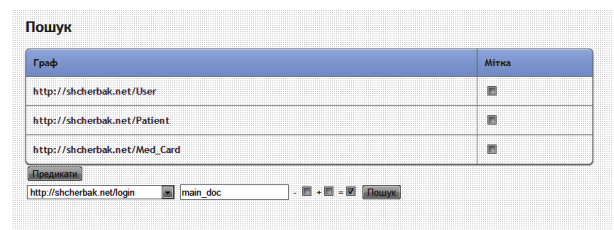


Рисунок 3 – Пример поискового интерфейса

Процедура составления запроса сводится к выбору поименованных графов, представляющих интерес в контексте поиска и наложении

ограничений на предикаты и значения выводимых в результате поиска объектов.

Результаты поиска группируются по принадлежности к некоторому субъекту в табличном представлении (Рис. 4). Таким образом, на клиенте из множества триплетов, полученных в результате выполнения запроса SPARQL на сервере, компоуется запись в удобном для чтения человеком виде.

Пошук

Граф	Мітка
http://shcherbak.net/User	■
http://shcherbak.net/Patient	■
http://shcherbak.net/Med_Card	■

Предикати

http://shcherbak.net/grantname

#	Ім'я	По-батькові	Прізвище	Посада	Відділення	Логін
5	Олександр	Іванович	Шевченко	Хірург	Хірургічне	main_doc

Рисунок 4 – Пример вывода результатов поиска в табличном представлении

Язык RDF позволяет формировать произвольной сложности графовые структуры для представления данных, что составляет как в вычислительном плане, так и в плане эффективного использования проблемы по организации взаимодействия через интерфейс ODBC (англ. Open Database Connectivity) с хранилищем триплетов, поэтому рассмотрим ограничения предложенного решения.

Предполагается, что данные, содержащиеся в хранилище триплетов, структурированы с использованием принципов объектно-ориентированного проектирования, а именно предполагается, что поименованный граф является контейнером (классом) для множества объектов этого класса, однозначно идентифицируемых по субъекту триплета.

Ограничения подразумевают исключительно более удобную работу с хранилищем триплетов для разработчиков программного обеспечения, использующих в своей работе реляционные системы управления базами данных. Без учета ограничений предложенное решение является работоспособным, но реальный смысл генерируемых данных в качестве результатов поиска измениться.

Для учета ограничений данные об объектах предлагается создавать поименованные графы на основе типовых, предлагаемых ниже, SPARQL запросов.

Для создания поименованных графов можно использовать следующий SPARQL-запрос:

```
create graph <http://shcherbak.net/User>
```

где создается граф пользователя «http://shcherbak.net/User» в текущем хранилище триплетов.

Для добавления триплетов в поименованный граф можно использовать следующий SPARQL-запрос:

```
INSERT DATA INTO
<http://shcherbak.net/User>
{ ns:log1 ns:login "dxfg".
  ns:log1 ns:firstname "Иван".
  ns:log1 ns:lastname "Иванович".
  ns:log1 ns:grantname "Иванов".}
```

ЗАКЛЮЧЕНИЕ

В данной работе предложена архитектура универсального поискового интерфейса, который обеспечивает визуальное построение и выполнение запросов к хранилищам триплетов на языке SPARQL. После выполнения запроса пользователь получает контекст с триплетами, соответствующими искомым с учетом заданных ограничений на предикаты и объекты.

Практическая реализация решения создана на языке программирования PHP и прошла успешную апробацию на базе хранилища триплетов OpenLink Virtuoso.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [DuCbarne, 2011] DuCbarne B. Learning SPARQL / B.DuCbarne. - O'Reilly Media: 2011. - 258 с.
 [Powers, 2008] Powers S. Practical RDF / S. Powes. - O'Reilly Media: 2008. - 352 с.
 [Холзнер, 2004] Холзнер С. XML. Энциклопедия, 2-е издание / Стивен Холзнер. - СПб.: Питер, 2004. - 1101 с.:ил.
 [Бек, 2008] Бек К. Шаблоны реализации корпоративных приложений.: Пер. с англ. - М.:ООО "И. Д. Вильямс", 2008. - 176 с.:ил.

PRACTICAL ASPECTS OF SEARCH INTERFACE IMPLEMENTATION FOR SEMANTIC APPLICATIONS BASED ON LINKED DATA

Galushka I.N. *, Zavgorodniy V. V. **, Soloshish S.N. *, Shcherbak S.S. *

*Kremenchuk Mykhailo Ostrohraskyi National University, Kremenchuk, Ukraine
 sergey.shcherbak@gmail.com

** Dneprodzerzhinsk state technical university,
 Dneprodzerzhinsk, Ukraine

The increase of linked data popularity along with information description in the form of RDF-triplets caused the necessity of studying the procedures for interaction with triple stores. A unique search interface have been developed, which enables visual querying the triple stores. These visual queries are automatically converted into SPARQL query language that is used for triple store accessing. After the query is executed, a user gets the desired context with triplets according to the set constraints for predicates and object



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 33:518/519

СЕМАНТИЧЕСКИЕ ВЭБ-СЕРВИСЫ ДЛЯ ИНТЕГРАЦИОННЫХ РЕШЕНИЙ

Вишняков В.А. *, Бородаенко Ю.В. **

* *Минский институт управления, г. Минск, Республика Беларусь*
vish2002@list.ru

** *Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*
jborodaenko@mail.ru

В докладе предложена модель интеграционного решения с элементами семантического представления данных. Исследовано направление повышения функциональности и гибкости интеграционного решения в Интранете, основанное на технологиях семантического Вэба. Это повышает степень интеллектуализации распределенной обработки информации, связанной с использованием онтологий

Ключевые слова: семантический Вэб, RDF технология, SWWS.

ВВЕДЕНИЕ

Проблема интеграции приложений неразрывна связана с проблемой повышения степени автоматизации обработки данных в Web: в то время как пространство Web продолжает экспоненциально расти, и включать в себя все новые информационные ресурсы, средства их обработки ограничены отсутствием семантики данных. Решением данной проблемы занимаются исследователи в области Semantic Web, ставящие перед собой цели повысить эффективность распределенного взаимодействия информационных систем. Подход, применяемый разработчиками Semantic Web, среди которых – консорциум W3C, основан на расширении существующего Web средствами выделения семантики, предназначенной для компьютерного восприятия. Архитектура Semantic Web предполагает наличие у любой информации, находящейся в сети, связанный с этой информацией точный смысл, обеспечиваемый следующими технологиями [Berners-Lee T., 2001]:

- стандарты представления знаний посредством семантической разметки в Web – язык RDF;

- онтологии для семантической базы терминов и осуществления логического вывода.

1. Семантические Вэб-сервисы

Выражение семантики с помощью RDF обеспечивается структурированием данных

посредством триплетов: объект описания, его свойство, значение. Однозначная идентификация объектов и их свойств обеспечивается с помощью URI (Universal Resource Identifier), значения допускается представлять в символьной форме. Следует выделить два приложения RDF, одно из которых — средство моделирования, позволяющее структурировать ресурсы в Web в виде триплетов. В рамках данного направления RDF является самостоятельной технологией, нашедшей применение, например в качестве внутреннего представления данных в браузере Mozilla. Другим приложением RDF является использование его основ в логических языках описания и сопоставления онтологий (средство RDFS, Schema RDF). Онтологии предназначены для описания предметной области в терминах отношений между сущностями и их ограничениями (constraints), которые необходимы информационным системам для восприятия семантики с целью анализа, сравнения, сопоставления данных, а также вывода новых знаний из уже имеющихся.

Одним из направлений Semantic Web (рисунок 1) является разработка семантических Web-сервисов (SWWS) на базе онтологий, расширяющих WSDL средствами выражения семантики, что позволяет существенно увеличить степень автоматизации обработки данных в Web.

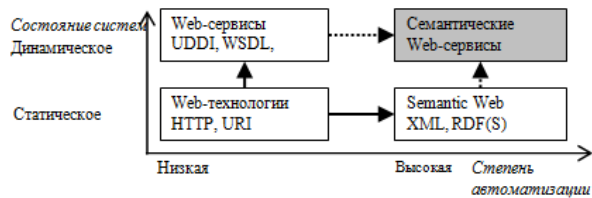


Рисунок 1- Развитие информационного пространства Web

На рисунке 1 отражены направления развития информационных систем: от статических к динамическим системам, и от низкой к более высокой степени автоматизации обработки данных. Свойство динамичности семантических Web-сервисов обеспечивается возможностью модификации параметров его вызова в реальном времени, а высокая степень автоматизации – онтологией (рисунок 2), позволяющей однозначно идентифицировать программе-агенту предназначение, содержание, технические детали вызова конкретного сервиса. Язык описания онтологии Web-сервиса DAML-S (DAML-based Web Service Ontology) содержит классы ServiceProfile, ServiceModel и ServiceGrounding, описывающие характеристики сервиса (рисунок 2).

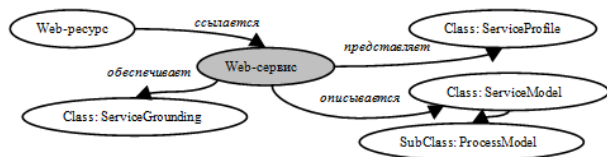


Рисунок 2 - Семантическое описание Web-сервиса

Следует отметить, что принципы семантических Web-сервисов будут использованы при разработке модели качества интеграционных решений. Среди проблем разработки и распространения технологий Semantic Web выделяется проблема наличия критической массы сервисов и программ-агентов, являющихся условием развития семантической инфраструктуры. Технологии, позволяющие агентам обращаться к сервисам, основаны на RDF и онтологиях; языки коммуникаций с другими агентами включают в себя разработки по созданию формата обмена знаниями KIF (Knowledge Interchange Format), языка обработки знаний KQML (Knowledge Query and Manipulation Language) и др. [Bass L., 2003]

Следует также отметить, что при реализации модели взаимодействия компьютерных систем возникают существенные проблемы безопасности и доверия, при этом представляется желательным опережающее развитие технологий защиты открытых систем по сравнению с их технической готовностью. Потребность в развитых механизмах защиты и оценки достоверности информации обуславливается открытой моделью данных, позволяющей объединять метаданные из разных источников. Разработчиками Semantic Web ведется активный поиск и разработка средств обеспечения безопасности информации и доверия к источникам, не ущемляющих идеи ее свободного распространения.

2. Модель SWWS для интеграционного решения

Дополним модель качества интеграционного решения для корпоративной сети элементами семантического представления данных, рассмотренных выше. [Borodaenko J., 2003] Развитие технологий семантических Web-сервисов (SWWS) [Terziyan V., 2003] обусловлено необходимостью их автоматического вызова и использования различными интеллектуальными агентами. Для обеспечения понимания программ-агентам предназначения Web-сервиса необходимо сопровождение его онтологией или семантическим описанием. Использование на предприятии технологии структурного представления семантики Web-сервиса, основанной на языке DAML+S, обеспечивает программ-агентам следующие возможности [Fencel D.,]:

- поиск требуемого SWWS и точное определение его предназначения с использованием класса ServiceProfile;
- запуск и выполнение SWWS с использованием класса ServiceGrounding;
- композиция SWWS для достижения определенных бизнес-целей (класс ServiceModel);
- мониторинг, позволяющий определять свойства SWWS и следить за его выполнением (класс ServiceModel).

Применение вышеуказанных технологий семантического представления данных и интеллектуальных программ-агентов способствует интеллектуализации распределенной обработки данных в Интернете и за счет этого их использования в Интранете, повысит эффективность обработки бизнес-процессов с уменьшением человеческого фактора (рисунок 3).

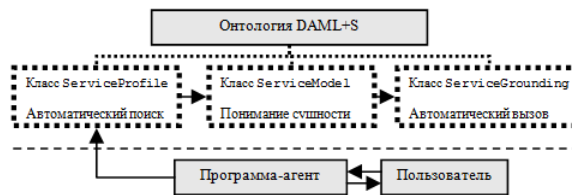


Рисунок 3 – Взаимодействие программы-агента и SWWS

Появление в пространстве Web необходимого количества онтологий, покрывающей все области человеческой деятельности, и развитие программно-информационных средств семантического Web (системы логического вывода, языки семантических запросов, хранилища знаний на основе сетевой модели данных RDF) обеспечат распространение семантических Web-сервисов и их внедрение на предприятиях. Это позволит отыскивать и комбинировать Web-сервисы, удовлетворяющие требованиям бизнес-процессов, сформулированным на языке высокого уровня.

На рисунке 4 предложена схема взаимодействия агента с пользователями, сервисами, другими

агентами, которые накапливают и обмениваются знаниями.

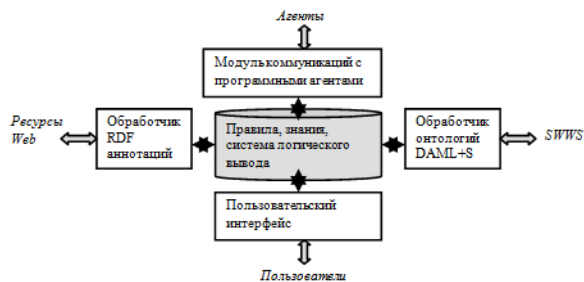


Рисунок 4 – Структура семантического взаимодействия программы-агента.

Использование на производстве технологий семантического Web, обеспечивающих агентов возможностями самостоятельно обнаруживать, запускать, комбинировать и следить за исполнением Web-сервисов без участия человека, является элементом интеллектуализации распределенного взаимодействия информационных систем [Borodaenko J., 2003].

3. СТРУКТУРА ИСПОЛЬЗОВАНИЯ SWWS

Рассмотрим предложенный вариант использования SWWS для интеграционных решений, построенных на основе семантических децентрализованных P2P сетей (peer-to-peer, тип коммуникации «равный с равным») [Бородаенко Ю. В. 2005]. В качестве архитектуры соединения приложений предприятия может быть использована топология гиперкуба P2P (HyperCub P2P, HyperCuP), уменьшающая загруженность сети сообщениями, поскольку каждый узел (приложение) связывается напрямую лишь с соседними узлами (рисунок 4). Кроме того, узлы P (peer) могут быть объединены в кластеры, и быть связаны лишь с суперузлом SP (super-peer), выполняющего функции маршрутизации сообщений и обработки запросов на основе собственного RDF репозитория.

Последний состоит из семантического описания метаданных о собственных узлах (индексы SP/P, superpeer/peer) и соседних суперузлах (индексы SP/SP, superpeer/superpeer). Благодаря семантической индексации, все узлы P2P сети идентифицированы и не требуют наличия центрального сервера для адресации и маршрутизации сообщений. Алгоритмы присоединения и удаления узлов, а также параметры модели HyperCuP могут быть найдены в [[Вишняков, В.А., 2010]. Следует выделить следующие принципы интеграционной модели P2P:

- архитектура P2P на основе топологии гиперкуба;
- описание узлов и суперузлов на основе RDF модели;
- использование транспортной архитектуры JXTA;
- использование функций составления и реализации запросов, интегрирующей ресурсы

образовательной среды (например, разработанных для информационной системы [Edutella, 2002]).

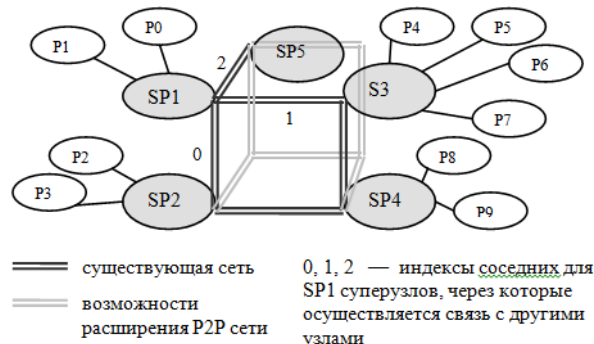


Рисунок 5 - Топология сети RDFPeers на основе графа HyperCuP

Следует отметить, что технологии семантических Web-сервисов находятся на стадии разработки и мало распространены в современном Web. Для массового внедрения архитектуры семантических Web-сервисов необходимо накопление критической массы онтологий и основанных на них приложений, использующих базы знаний и поддерживающих RDF.

4. Оценка функциональности гибкости модели SWWS

Главным приложением семантических Web-сервисов является повышение автоматизации и гибкости производственных процессов. Метрикой гибкости производственных процессов, расширяющей модель качества элементами семантического представления данных, является семантическое взаимодействие приложений [Бородаенко Ю. В. 2005]. Оценочные элементы данной метрики SWWS приведены в таблице 1 и включают автоматизацию поиска, вызова, композиции и мониторинга исполнения процесса. Низкий вес данных элементов (0,5) обусловлен тем, что технологии семантических Web-сервисов находятся на стадии разработки и мало распространены в современном Web. Для массового внедрения архитектуры семантических Web-сервисов необходимо накопление критической массы онтологий и основанных на них приложений, использующих базы знаний и поддерживающих RDF.

Способность к взаимодействию является ключевой функциональностью интеграционного решения, предложим метрику, отражающую возможность создания более сложных программных модулей (объектов или сервисов) из модулей низкого уровня или композицию (таблица 2). Рекурсивная композиция обеспечивается технологией BPEL, поддерживающей динамический вызов Web-сервисов во время исполнения бизнес-процесса, что позволяет решению. SWWS присвоить значение 1.

Таблица 1 - Оценки гибкости интеграционных решений с использованием технологий семантического Web

Оценочные элементы гибкости	Вес	S W W S
Автономность функций	1	1
Отсутствие состояния	1	1
Открытые стандарты описания функций	1	1
Отделение логики процесса от логики бизнеса	2	1
Выделение логики процесса в документ XML	1	1
Автоматизация поиска подпроцесса	0,5	1
Автоматизация вызова подпроцесса	0,5	1
Автоматизация композиции подпроцессов	0,5	1
Автоматизация мониторинга	0,5	1
Итого по гибкости	8	1

Таблица 2 - Оценки способности интеграционных решений к взаимодействию

Атрибуты	Вес	S W W S
Рекурсивная композиция	1	1
Отсутствие зависимости от состояния	1	1
Открытые стандарты описания функций	1	1
Автоматизация поиска программного модуля	1	1
Автоматизация вызова программного модуля	1	1
Итого		1

Заключение

Интеграционное решение на основе семантических Web-сервисов является одним из ключевых направлений разработок в области повышения автоматизации производственных процессов и интеллектуализации распределенного взаимодействия. Появление в Web критической массы онтологий и развитие технических средств семантического Web (системы логического вывода, языки семантических запросов, хранилища знаний) обеспечат распространение семантических Web-сервисов и их внедрение на предприятиях, что позволит отыскивать и комбинировать Web-сервисы, удовлетворяющие требованиям бизнес-процессов, сформулированным на языке высокого

уровня. Исследовано направление повышения функциональности и гибкости интеграционного решения в Интранете, основанное на технологиях семантического Web и повышающее степень интеллектуализации распределенной обработки информации на предприятии, связанной с использованием онтологий.

Библиографический список

- [Berners-Lee T., 2001] Berners-Lee T. The Semantic Web / T. Berners-Lee, J. Hendler, O. Lassila // Scientific American, May 2001. — P.28-37.
- [Bass L., 2003] Bass L. Software Architecture in Practice, Second Edition / L. Bass, P. Clements, R. Kazman. - Addison Wesley Professional, 2003. - 560 p.
- [Borodaenko J., 2003] Borodaenko J. Intelligence Business Process Approach to Electronic Commerce / V. Vishnyakov, Borodaenko J. // Нейронные сети и искусственный интеллект: материалы 3-й Межд. конф., Минск, БГУИР, ноябрь 2003.— С.196-200.
- [Terziyan V., 2003] Terziyan V. Semantic Web Enabled Web Services: State-of-Art and Industrial Challenges / V. Terziyan, O. Kononenko. In: M. Jeckle and L.-J. Zhang (eds.), Web Services - ICWS-Europe 2003, Lecture Notes in Computer Science. — Springer-Verlag, 2003.— Vol. 2853.- P. 183-197.
- [Fencel D., 2000] Fencel D., The Semantic Web and Its Languages / D. Fencel // IEEE Intelligent Systems.— 2000.— vol.15, no. 6. - P. 67-77.
- [Вишняков, В.А., 2010] Вишняков, В.А., Интеллектуальные системы в управлении: учеб.-метод. комплекс / В.А. Вишняков. — Минск: Изд-во МИУ, 2010. — 364с.
- [Edutella: A , 2002] Edutella: A P2P Networking Infrastructure Based on RDF. / Nejd W., Wolf B., Qu C., Decker S // In Proceedings of the 11th International World Wide Web Conference, Hawaii, USA, May, ACM Press, 2002. — P. 604-615.
- [Бородаенко Ю. В. 2005] Интеграционные решения для корпоративной информационной системы. // Известия белорусской инженерной академии. — 2005.— 2(20)/1.— С. 142-144
- [Вишняков В.А., 2011] Вишняков, В.А., Модели и средства интеграции приложений, маркетинга, аутсорсинга, обработки знаний в компьютерных сетях: монография / В.А. Вишняков, Ю.В.Бородаенко, Д.С., Бородаенко. - Минск, МИУ, 2011. — 350с).

SEMANTIC WEB-SERVICES FOR INTEGRATION DECISIONS

Vishniakou U.A. *, Borodaenko J.V.**

* Minsk Management Institute, Minsk, Belarus
vish2002@list.ru

** Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus
jborodaenko@mail.ru

The model of integration decision with elements of data semantic representation is given. Direction of the higher of functionality and flexibility for integration decision is investigated on the Semantic Web technologies base in Intranet. It high the state of intelligence of distributed information processing connecting with the otology use.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

SEMANTIC EVALUATION AT LARGE SCALE

Yatskevich M.I., Apanasovich V.V.

*School of Business and Management of Technologies, Belarusian State University
Minsk, Republic of Belarus*

yatskevichN@sbmt.by

apanasovich@sbmt.by

The Semantic Web technology often called Web 3.0 is aiming at making the information easily accessible for humans on the Web to be accessible for computers. The Semantic Web technologies are based on the notion of an ontology namely specification of conceptualization for a given domain. The growth and success of the Semantic Web is built upon a wide range of semantic technologies. The results obtained will facilitate adoption of the Semantic Web technologies within business environment and will be used for semantic technology adoption within School of Business and Management of Technologies.

Keywords: semantic web; evaluation; ontology; reasoning.

INTRODUCTION

Semantic technologies play a critical role in the recent advances in both the Web (the Semantic Web) and corporate knowledge management by providing ways to express knowledge and data in an automated way for different purposes such as information retrieval or data integration.

The evaluation of such technologies is crucial for their sustained improvement and adoption, allowing users to assess the suitability of current technologies to their needs. Some initiatives have already created a basis for semantic technology evaluation, such as those in the areas of ontology matching [Euzenat et. al, 2010], ontology engineering [Garcia-Castro et. al, 2009, Garcia-Castro et. al, 2010b], ontology reasoning [Horrocks et. al, 1998, Massacci et. al, 2000], semantic search [Kaufmann, 2007] or semantic web services [Klusch et. al, 2012, Petrie et. al, 2009].

At the heart of the EU-funded Framework 7 SEALS Project is the development of the SEALS Platform [Garcia-Castro et. al, 2010a]: an open infrastructure for the evaluation of semantic technologies that offers independent computational and data resources for the evaluation of those technologies. SEALS evaluation campaign included tools from five different semantic technology fields (ontology engineering, semantic search, semantic web services, ontology matching, storage and reasoning) were formally evaluated [Nixon et. al, 2011]. The evaluation results demonstrated high level of semantic technologies development

highlighting critical elements of the semantic technologies ecosystem. However, some of the promises of the semantic technologies namely soundness and completeness of results had not been achieved by all the systems taking part in the evaluation.

The paper is structured as follows. Preliminaries of Semantic Web technologies are described in Section 2. Section 3 introduces technologies to be evaluated and their evaluation methodology while Section 4 briefly discusses the evaluation outputs.

1. Preliminaries

Semantic Web technologies are based on the several standardized languages (see Figure 1 for an updated stack). While URI and UNICODE provide a way for identification and dealing with text, eXtended Markup Language (XML) defines a way for introducing structured data. Resource Description Framework (RDF) introduces a way for defining graphs from XML data. RDF data is composed from triples corresponding to two nodes of the graph and labelled arc between them. Resource Description Framework Schema (RDFS) introduces ontological dimension allowing definition of classes, properties and is-a relationships among classes. Ontology Web Language (OWL) introduces more advanced constructs including cardinality restrictions, nominal, property restrictions, inheritance and composition. OWL language has a direct counterpart in description logics [Baader et. al, 2002] a family of reasoning formalisms restricting first-order logic to obtain sound and complete reasoning

procedures. OWL introduces several profiles namely OWL Lite allowing for 0 or 1 cardinality constraints; OWL DL corresponding to decidable description logic and OWL Full allowing for maximum expressivity. The OWL 2.0 introduced OWL 2 EL allowing for reasoning in polynomial time; OWL 2 QL designed for maximum interoperability with relational databases; and OWL 2 RL designed for interoperability with rules-based systems.

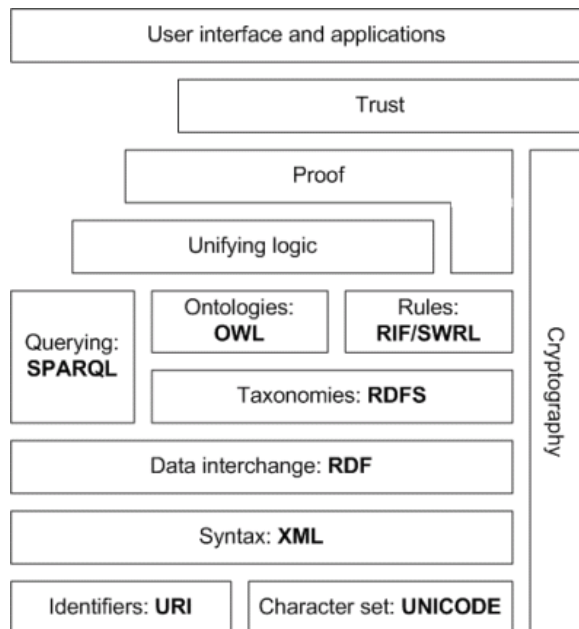


Figure 1. Semantic Web technologies stack.

SPARQL language allows for RDF data querying while Rule Interchange Format (RIF) and Semantic Web Rule Language (SWRL) allow adding ontological knowledge. The work on upper levels of the stack namely unifying logic, proof and trust is under way.

2. Target Technologies

The SEALS Project has identified five core technology areas which lie at the heart of the Semantic Web. The evaluations within these areas provide invaluable insights into the technologies themselves; insights which can be, and are being, used to improve performance of Semantic Web tools.

2.1. Ontology Engineering Tools

Two types of tools support ontology engineering tasks: ontology editors, which are user-oriented and allow creating and maintaining ontologies mainly through user interfaces, and ontology management programming interfaces, which are developer-oriented and allow the creation and maintenance of ontologies through programming interfaces. Conformance and interoperability evaluations [Garcia-Castro et. al, 2009, Garcia-Castro et. al, 2010b] use groups of ontologies defined in specific ontology languages as test data; these evaluations are performed by making tools process ontologies (coming either from test data or from other tools) and analysing the processed ontology (usually by comparing the processed ontology with that used as input).

2.2. Ontology Reasoning Tools

Description Logics (DLs) [Baader et. al, 2002] are a family of logic-based knowledge representation formalisms designed to represent and reason about the knowledge of an application domain in a structured and well-understood way. Besides their formal knowledge representation languages, DLs also provide inference services. The aim of such services is to extract new implied information out of the explicitly stated information. Every knowledge representation language usually offers a different set of inference services. The most widely used inference services include: class satisfiability, classification, logical entailment, ontology satisfiability, and instance retrieval. In order to interact with other systems an ontology reasoner must conform to standard input formats and must be able to provide standard inference services.

2.3. Ontology Matching Tools

Matching ontologies consists of finding a set of correspondences (alignment) between two different ontologies. A wide diversity of systems have been proposed, which can be classified according to the many features that can be found in ontologies (e.g., labels, structures, instances, semantics), or with regards to the techniques they use (e.g., statistics, combinatorics, semantics, linguistics, or machine learning) [Euzenat et. al, 2007b]. The most commonly used criterion for evaluating matching systems is the compliance of matcher alignments with respect to the expected reference alignments. Metrics such as precision and recall are largely adopted for quantitatively evaluating matching tools. Other evaluation criteria are efficiency, in terms of runtime and memory consumption, and scalability using large sets of tests; semantic measures, where the proximity between alignments is measured instead of their strict equality [Ehrig et. al, 2005, Euzenat, 2007a]; and task-specific evaluations, where alignments are evaluated according to their usage in some specific task.

2.4. Semantic Search Tools

State-of-the-art semantic search approaches are characterized by their high level of diversity both in their features as well as their capabilities. Such approaches employ different styles for accepting the user query (e.g., forms, graphs, keywords) [Uren et. al, 2007] and apply a range of different strategies during processing and execution of the queries. They also differ in the format and content of the results presented to the user. All of these factors influence the user's perception of performance and usability. Semantic search technologies can be evaluated on the basis of different criteria and metrics [Wrigley et. al, 2010, Kaufmann, 2007]. At the core of any search task is the retrieval of pertinent information; search evaluations employ several questions which are applied to a particular ontology and dataset. Since (for ontology-based search) the answer set for each question is finite and known a priori, the measures of precision and recall are used. We are also interested in how tools cope with increasingly large datasets (scalability). Since search is

an inherently user oriented task, evaluation must also consider metrics such as how long it takes for a query to be executed.

2.5. Semantic Web Service Tools

Semantic Web Service (SWS) technologies enable the automation of discovery, selection, composition, mediation and execution of web services by means of semantic descriptions of their interfaces, capabilities and non-functional properties. SWS provide a layer of semantics for service interoperability by relying on a number of reference service ontologies and semantic annotation extension mechanisms. The evaluation of SWS technologies is currently being pursued by a number of initiatives using different evaluation methods (e.g., see [Klusch et. al, 2012, Petrie et. al, 2009]). Although these initiatives have succeeded in creating an initial evaluation community in this area, they have been hindered by the difficulties in creating large-scale test suites and by the complexity of manual testing to be done.

3. Project outputs

In this paper we will concentrate on the outputs of the most recent SEALS evaluation campaign.

3.1. Ontology Engineering Tools

The evaluation was focused on conformance, interoperability and scalability criteria. Three ontology management frameworks and three ontology editors have been evaluated in the evaluation of the first two criteria. In the RDFS conformance task only three out of six systems had outputted the same ontologies they had in input. In OWL Lite five out of six systems had outputted the same ontologies they had in input and one achieved more than 95% precision. In OWL DL conformance tasks five out of six system had achieved 95% of precision or more while one system demonstrated more than 75% of precision. In content pattern, expressive pattern and content pattern full conformance tasks five systems demonstrated precision of 90% and more. In OWL Full conformance tasks two tools had demonstrated 100% of precision while the rest were in 10-80% range. For what concerns interoperability RDF(S) results 3 out of 6 tools had achieved 99% of interoperability. In OWL Lite interoperability tests the same level of interoperability had been achieved for all 5 systems. In OWL DL interoperability the level of 76% or higher had been achieved for all 6 systems while in OWL Full interoperability 3 of the systems had demonstrated 74% or higher levels of interoperability while the rest enjoyed 19% or lower levels of interoperability. In scalability tests 9 ontology managing frameworks and ontology editors had competed on 4 real world and artificially generated datasets on the tasks ranging from 0.5 to 1500 Mb. Ontology managing frameworks had outperformed the editors up to several orders of magnitude.

3.2. Ontology Reasoning Tools

The evaluation was focused on performance, interoperability and scalability criteria. Four description logic reasoners had participated in the evaluation campaign in classification, class satisfiability, ontology satisfiability, entailment and instance retrieval tasks. For classification tasks the “golden standard” classifications were available for about half of the 102 ontologies used in the task. The OWL DL supporting systems demonstrated 80-85% of precision in the task. The average reasoning time for all ontologies was within 3 seconds range while the time cut off limit was set to 1 hour. Two ontologies for one of the reasoners and one for the other had not been decided within the given time frame. In OWL EL tests all the ontologies had been classified correctly and average execution time was within tenth of a second for all the systems. For OWL DL class satisfiability tasks both systems demonstrated 95% or more of precision while execution times were within 0.5 seconds. In OWL EL class satisfiability tests all the systems demonstrated correct results while the reasoning time not exceeded tenth of a second. For OWL DL ontology satisfiability tasks both systems demonstrated precision in 97%-100% range. The average reasoning time was about 0.5 seconds for all the systems. All OWL EL ontologies have been solved within 0.2 seconds in average. For entailment tests one of the tools was able to solve 98% of tasks while the others less than 25%. For non-entailment tests OWL DL systems were able to solve more than 85% of tasks while OWL EL systems had not solved any of the tasks. For instance retrieval tests all the tasks were solved correctly. In scalability evaluations one of the systems demonstrated exponential behaviour in class and ontology satisfiability depending on the class counts while in other cases the behaviour of the systems was in line with complexities of the corresponding logics.

3.3. Ontology Matching Tools

The evaluation was focused on quality and scalability criteria. The rich set of ontologies has been collected including multilingual ontologies in nine languages. The size of ontologies ranged from tenth to tenth of thousands of classes and from tenth to hundreds of properties. Fourteen systems took part in the evaluation. Precision ranged from 1% to 99% in English and 1% to 97% in multilingual settings while recall was in 1% to 93% in English and 1% to 51% in multilingual settings. The tasks were solved in 7 to 66494 seconds depending on size.

3.4. Semantic Search Tools

The evaluation has been split into 2 phases user-in-the-loop and automated. The user-in-the-loop phase is concerned with asking a number of search questions from human subjects while in automated phase systems are assessed automatically. The datasets from geography and software engineering domains were used for the evaluation. Eight systems had participated in the evaluation. Twenty persons answered the questionnaire regarding usability of the tools. Perceived usability

ranged from 32.5% to 63.75% while answer found rate was in 20% to 80% range and input time ranged from 19.9 to 102.52 seconds. The query times for automated phase ranged from 0.5 to 169.7 seconds.

3.5. Semantic Web Service Tools

Three datasets were used for retrieving web services that match to a given partial specification. Three systems participated in the task. Precision ranged from 13% to 98% while recall was in 4% to 92% range.

Conclusion

The evaluation results demonstrated importance of mutual assessments and perceived scalability for a wide range of semantic technologies. The results obtained will facilitate adoption of the Semantic Web technologies within business environment and will be used for semantic technology adoption within School of Business and Management of Technologies.

References

- [Baader et. al, 2002] F. Baader, D. McGuinness, D. Nardi, and P. Patel-Schneider. The Description Logic Handbook: Theory, implementation and applications. Cambridge University Press, 2002.
- [Ehrig et. al, 2005] M. Ehrig and J. Euzenat. Relaxed precision and recall for ontology matching. In B. Ashpole, M. Ehrig, J. Euzenat, and H. Stuckenschmidt, editors, Proc. of the Workshop on Integrating Ontologies, volume 156, page 8. CEUR-WS.org, August, 2005.
- [Euzenat, 2007a] J. Euzenat. Semantic precision and recall for ontology alignment evaluation. In Proc. of the 20th International Joint Conference on Artificial Intelligence, pages 248–253, Hyderabad, India, 2007.
- [Euzenat et. al, 2010] J. Euzenat, A. Ferrara, C. Meilicke, A. Nikolov, J. Pane, F. Scharffe, P. Shvaiko, H. Stuckenschmidt, O. Sv.ab-Zamazal, V. Svatek, and C. Trojahn dos Santos. Results of the Ontology Alignment Evaluation Initiative 2010. In Proc. of the 5th Workshop on Ontology Matching, pages 85–117, Shanghai, China, 2010.
- [Euzenat et. al, 2007b] J. Euzenat and P. Shvaiko. Ontology matching. Springer, Heidelberg(DE), 2007.
- [Garcia-Castro et. al, 2010a] R. Garcia-Castro, M. Esteban-Gutierrez, and A. Gomez-Perez. Towards an infrastructure for the evaluation of semantic technologies. In Proc. of the eChallenges 2010 Conference, October 27-29 2010.
- [Garcia-Castro et. al, 2009] R. Garcia-Castro and A. Gomez-Perez. RDF(S) Interoperability Results for Semantic Web Technologies. International Journal of Software Engineering and Knowledge Engineering, 19(8):1083–1108, 2009.
- [Garcia-Castro et. al, 2010b] R. Garcia-Castro and A. Gomez-Perez. Interoperability results for Semantic Web technologies using OWL as the interchange language. Web Semantics: Science, Services and Agents on the World Wide Web, 8:278–291, November 2010.
- [Horrocks, 1998] I. Horrocks and P. F. Patel-Schneider. DL systems comparison. In Proc. of the 1998 Description Logic Workshop, volume 11, pages 55–57, 1998.
- [Kaufmann, 2007] E. Kaufmann. Talking to the Semantic Web — Natural Language Query Interfaces for Casual End-Users. PhD thesis, Faculty of Economics, Business Administration and Information Technology of the University of Zurich, September 2007.
- [Klusch et. al, 2012] M. Klusch, A. Leger, D. Martin, M. Paolucci, A. Bernstein, and U. Kuester. Annual International Contest S3 on Semantic Service Selection. <http://www-ags.dfki.uni-sb.de/~klusch/s3/>
- [Massacci et. al, 2000] F. Massacci and F. M. Donini. Design and results of TANCOS-2000. In Proc. of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, pages 52–56, London, UK, 2000. Springer-Verlag.
- [Nixon et. al, 2011] L. Nixon, R. Garcia-Castro, S. N. Wrigley, M. Yatskevich, C. T. D. Santos, and L. Cabral. The state of semantic technology today – overview of the first seals evaluation campaigns.

In Proc. of the 7th International Conference on Semantic Systems, 2011.

[Petrie et. al, 2009] C. Petrie, T. Margaria, H. Lausen, and M. Zaremba. Semantic Web Services Challenge: Results from the First Year. Springer, 2009.

[Uren et. al, 2007] V. Uren, Y. Lei, V. Lopez, H. Liu, E. Motta, and M. Giordano. The usability of semantic search tools: a review. The Knowledge Engineering Review, 22(4):361–377, 2007.

[Wrigley et. al, 2010] S. N. Wrigley, D. Reinhard, K. Elbedweihy, A. Bernstein, and F. Ciravegna. Methodology and campaign design for the evaluation of semantic search tools. In Proc. of the International Workshop on Semantic Search, 2010.

КРУПНОМАСШТАБНАЯ ОЦЕНКА СЕМАНТИЧЕСКИХ ТЕХНОЛОГИЙ

Яцкевич Н.И., Апанасович В.В.

**Институт Бизнеса и Менеджмента
Технологий, Белорусский
Государственный Университет, Минск,
Республика Беларусь
yatskevichN@sbmt.by
apanasovich@sbmt.by*

Проект Оценка Семантической Масштабируемости создает открытую и устойчивую платформу, в которой все аспекты процесса оценки, его размещения и выполнения принимаются во внимание для большинства типов технологий.

ВВЕДЕНИЕ

Технологии семантического веба, известные как Веб 3.0, направлены на предоставление компьютерам того же уровня понимания информации что и людям. Они основаны на понятии онтологии или спецификации концептуализации данной предметной области.

ОСНОВНАЯ ЧАСТЬ

Развитие и успех семантического веба построены на широком спектре технологий, а именно, редакторов онтологий, систем автоматического принятия решений построенных на онтологических принципах, систем нахождения онтологических отображений, технологий семантического поиска информации и веб служб. Оценка качества и совместимости этих технологий критична для их устойчивого развития и распространения. Результаты оценки продемонстрировали высокий уровень развития семантических технологий и подчеркнули наиболее важные их элементы.

ЗАКЛЮЧЕНИЕ

Полученные результаты продемонстрировали важность оценки взаимодействия и масштабируемость всего спектра семантических технологий и будут использованы для их внедрения в Институте Бизнеса и Менеджмента Технологий Белорусского Государственного Университета.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

РАЗРАБОТКА WEB-САЙТОВ СО СЛОЖНОСТРУКТУРИРОВАННЫМ КОНТЕНТОМ НА ОСНОВЕ СЕМАНТИЧЕСКИХ СЕТЕЙ

Колб Д.Г., Фурман О.Д.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

kolb@bsuir.by

olgasharpio@gmail.com

Приведен подход к построению семантически структурированных web-сайтов. Кратко описана модель web-сайтов данного класса, рассмотрена идея организации навигационного поиска в web-сайтах данного класса. Предложен метод отображения класса семантических сетей, используемых для разработки семантически структурированных web-сайтов, на модель RDF.

Ключевые слова: семантическая сеть; интеллектуальная система; Semantic web.

ВВЕДЕНИЕ

Результаты исследования независимых аналитических агентств Gartner Group и Real Story Group за 2011-2012 гг. показывают, что одной из важнейших тенденций развития систем управления web-контентом, в частности в корпоративных web-сайтах, является переход от web-сайтов с простой структурой к крупным интегрированным порталам корпоративных знаний. Указанная тенденция позволяет существенно повысить эффективность использования накопленного корпоративного опыта. С точки зрения разработчика, порталы – это web-сайты, характеризующиеся представлением сложноструктурированной информации и требующие многолетней информационной поддержки. Основной ценностью в них являются накапливаемые за время поддержки объемы корпоративных знаний.

Под сложноструктурированной информацией в данном случае понимаются многоуровневые гипертекстовые информационные структуры, в которых имеют место связи не только между сетями (первичными элементами гипертекста), но и между связями, а также между целыми информационными структурами. Связи в таких иерархических информационных структурах могут быть не только бинарными, но и многоместными; не только позитивными, но и негативными и нечеткими; не только стационарными во времени, но и не стационарными; не только константными, но и переменными. Примерами web-сайтов, оперирующих такой информацией, являются

научные, образовательные, энциклопедические порталы, порталы проектных организаций и др. Очевидно, что для разработки web-сайтов указанного класса необходимо использовать модели, позволяющие формально представлять всё многообразие связей, присутствующих на web-сайте. Нам представляется, что наиболее подходящей моделью для представления сложноструктурированной информации является семантическая сеть.

Проблема разработки web-сайтов, в основе которых лежат семантические сети, всё чаще входит в сферу интересов многих ученых всего мира. С одной стороны решение этой проблемы позволит перейти на новый качественный уровень Всемирной Паутины – Semantic Web, с другой стороны позволит преодолеть еще одну ступеньку на пути к созданию интеллектуальных систем и заложить платформу для создания интеллектуальных систем для Internet.

Жизненный цикл процесса разработки web-сайтов включает следующие основные задачи:

- наполнение контента web-сайта (в том числе и мультимедийного);
- оптимизация контента для поисковых машин;
- разработка средств навигации и поиска по web-сайту;
- обеспечение переносимости контента web-сайта.

Современные CMS с различным уровнем полноты решают указанные задачи, однако остается

и ряд не решенных задач, решение которых невозможно получить без использования методов и средств искусственного интеллекта. В число таких задач входят:

- обеспечение высокого уровня релевантности поисковых запросов ответам;
- семантическая структуризация и оптимизация страниц web-сайтов с учетом формальной спецификации, которую может “понять” и проиндексировать поисковая машина;
- обеспечение кроссплатформенности контента web-сайта не только между различными версиями программных средств, а и между различными программными средствами реализации.

В рамках данной работы авторы попытаются рассмотреть возможные пути решения, указанных проблем, на основе средств, которые предлагаются в рамках направления Semantic Web.

1. Модель семантически структурированного web-сайта

1.1. Зачем нужны модели web-сайтов?

Рассматривая решения, которые в настоящее время предлагаются для совершенствования web-ресурсов указанного класса, необходимо отметить два значимых направления работ:

1. разработка методов и средств спецификации страниц web-сайта, которые позволяют осуществлять лёгкую индексацию web-страниц и, соответственно, обеспечить более релевантный поиск;
2. разработка методов и средств спецификации информации, которые позволяют представлять сложноструктурированные виды информации, в том числе, такие как теоремы, аксиомы, логические высказывания и др.

К первому направлению можно отнести ряд работ, позволяющих на основе модификации языка разметки осуществить семантическую разметку web-страницы [Микроформаты, 2012], [Микроданные, 2012], [RDFa, 2012] или её семантическую аннотацию [SMW, 2012], [SMW+, 2012], [KnowWE, 2012]. Средствами этого направления, как правило, присуща малая выразительная мощность [Хорошевский, 2008]. Однако использование этих средств существенно облегчает работу по семантическому анализу содержимого web-страницы, которая в настоящее время чаще всего осуществляется с помощью средств Text mining.

Ко второму направлению относятся работы, в основе которых лежит представление web-ресурсов в виде семантических сетей на основе модели RDF, используя базовые RDF-словари RDFS, OWL или OWL 2. Выразительная мощность указанных средств прежде всего, определяется практической направленностью задач, которые пытаются решать различные научные, участвующие в разработке стандартов направления Semantic Web. С точки

зрения практической используемости для разработки web-сайтов данные средств тяжелы в освоении и далеко не всегда могут удовлетворить требования разработчика интеллектуальных систем для Internet [Хорошевский, 2008].

Возникает дилемма – использовать маловыразительные средства, которые адаптированы для задач промышленной разработки web-приложений или использовать средства, обладающие большей выразительной мощностью, однако не адаптированные для промышленной разработки.

В рамках данной работы авторы попытаются предложить решение, которое позволит интегрировать два приведенных выше направления разработки web-приложений, на основе использования моделей представления знаний более высокого абстрактного уровня. Такой подход с точки зрения авторов позволит с одной получить промежуточное решение для прикладной промышленной разработки web-сайтов на базе семантических сетей. И даст толчок к эволюционированию существующих в настоящее время средств разработки web-приложений, основанных на семантических сетях.

1.2. Компоненты семантической модели web-сайта

Семантически структурированными web-сайтами назовём специальный класс интеллектуальных систем, построенных на базе семантических сетей и гипертекстовой модели представления информации. Модель семантически структурированного web-сайта задается следующим образом:

$$M = \{O, S_A, S_P, O_{hyp}, M_L, M_S, I\}, \quad (1)$$

где

O – онтология предметной области, по которой разрабатывается web-сайт, в рамках которой специфицированы сущности предметной области и отношения между ними;

S_A – множество классов статей семантически структурированного web-сайта (атомарных семантических единиц), которое строится на основании онтологии O предметной области;

S_P – множество классов страниц семантически структурированного web-сайта, каждый из которых состоит из некоторого множества классов статей и каждый элемент, которого связан с другим элементом отношением порядка следования. Такие связи позволяют определить последовательность изложения подачи «материала» предметной области содержимого web-страницы, который будет размещен на web-странице;

O_{hyp} – онтология мультимедийных документов, которая строится с учетом концепции языка гипермедийных сетей, который был разработан для семантической спецификации мультимедийных данных. В его основе лежит универсальная система метаданных «Дублинское ядро» и однородные

семантические сети с базовой теоретико-множественной интерпретацией [Колб, 2009].;

M_L – языковые средства представления информации в виде однородных семантических сетей;

M_S – модель семантического навигационного поиска, которая состоит из семейства операций обработки однородных семантических сетей, обеспечивающих навигацию по семантически структурированному web-сайту;

I – способ отображения модели семантически структурированного web-сайта на модель RDF.

Указанную модель семантически структурированного web-сайта будем называть SC-моделью web-сайта, так как в её основе лежит

предметной области, по которой разрабатывается web-сайт. Класс статей (семантических окрестностей) A_i задается следующим образом:

$$A_i = \{E, S_R, S_{Aij}\}, \quad (2)$$

где

E – сущность предметной области, по которой разрабатывается семантически структурированного web-сайта;

S_R – множество связей отношений, каждое из которых специфицировано в онтологии O таким образом, что сущность E является первым компонентом связки данного отношения;

S_{Aij} – множество статей частного вида, связанных с

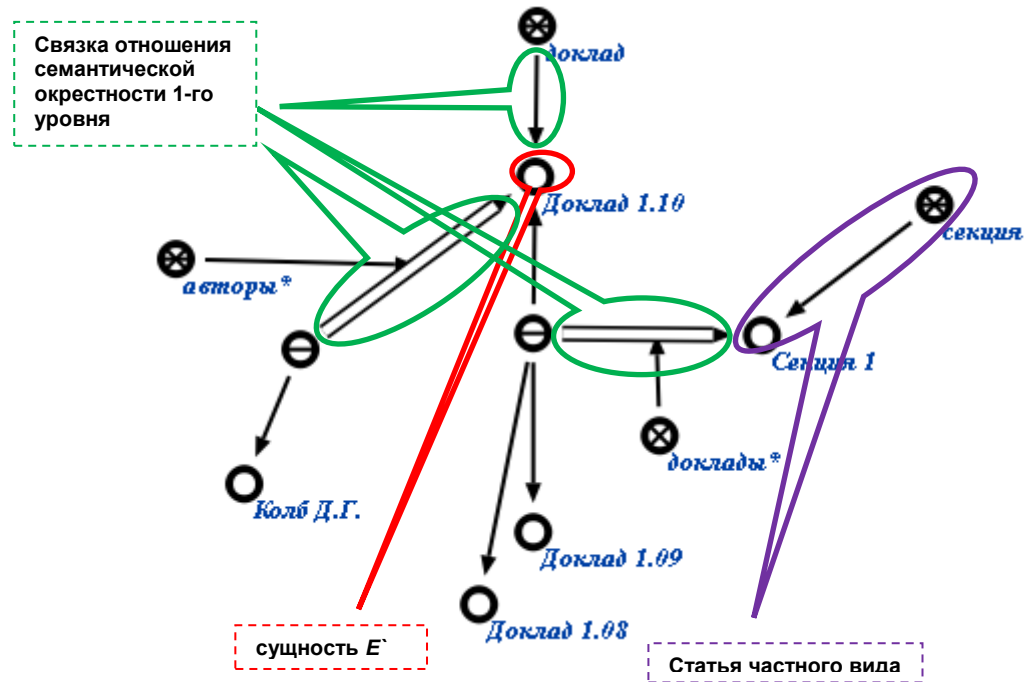


Рисунок 1 – Семантическая окрестность понятия Доклад 1.10

универсальный базовый способ представления семантических сетей в виде однородных семантических сетей с базовой теоретико-множественной семантической интерпретацией. Такой универсальный способ представления семантических сетей описан в работе [Голенков, 2011] и назван SC-кодом (Semantic Code).

Информационные конструкции, представленные в SC-коде, называются SC-графами. Кроме SC-кода, вводится способ символической визуализации SC-графов названный SCn-кодом (Semantic Code natural) [Голенков, 2011]. Текст SCn-кода представляет собой совокупность SCn-статей, каждая из которых является описанием семантической окрестности (см. рисунок 1) некоторого соответствующего ей элемента SC-графа.

Каждый элемент множества S_A задает класс атомарных семантических единиц (статей) семантически структурированного web-сайта. В рамках данной работы статьёй семантически структурированного web-сайта назовем семантическую окрестность некоторой сущности E

сущностью E посредством множества связей отношений S_R .

Каждая семантическая окрестность A_i задается на множестве отношений, связки которых являются элементами S_R , и может характеризовать сущность E с различной степенью глубины. Семантическую окрестность, характеризующую сущность E с глубиной n , назовем семантической окрестностью уровня n . Полной семантической окрестностью сущности E будем называть такую семантическую окрестность, которая включает связки всех отношений, в которые включен знак данной сущности.

Рассмотрим пример формального представления статьи семантически структурированного web-сайта, выделив её элементы согласно (2). Пусть E' некоторое понятие, которое описывается в рамках статьи A_i' . На рисунке 1 с помощью средств SCg-кода [Голенков, 2011] представлена семантическая окрестность второго уровня для сущности E' – Доклад 1.10. Множество связей отношений, которые являются элементами S_R' в рамках статьи A_i' на рисунке 1 подписаны указателем “связка

отношения семантической окрестности 1-го уровня”. Множество статей частного вида S_{A_i} в рамках статьи A_i на рисунке 1 подписаны указателем “статья частного вида”, здесь у данного множества один элемент соответствующий семантической окрестности сущности **Секция 1 A_0** . Указанная сущность связана с сущностью **Доклад 1.10** через связку отношения **доклады***. A_0 имеет непустое множество S_{0K} . Остальные сущности присутствующие на рисунке (**доклад**, **Доклад 1.09**, **Доклад 1.08**, **Колб Д.Г.**) в рамках данной статьи связаны только с сущностью E и, соответственно, не образуют статей частного вида. Используя конструкции, разобранные в данном примере, мы можем характеризовать любую сущность предметной области с «уровнем подробности», который определяет разработчик web-сайта исходя из своих потребностей. В рамках нашей работы этот «уровень подробности» мы называем семантической окрестностью понятия уровня n .

1.3. Модель семантического навигационного поиска

В семантически структурированных web-сайтах каждый переход между страницами является переходом от одной семантической окрестности некоторой сущности к другой семантической окрестности. Сам переход приводит к выполнению поисковой операции, которая инициируется на основании семантической спецификации операции, формируемой разработчиком web-сайта. Такую операцию можно расценивать как решение некоторой задачи семантического информационного поиска. Решение этой задачи может приводить к инициированию как обычных поисковых операций, так и операций логического вывода. Приведенная идея организации навигационного поиска характерна для интеллектуальных систем, и хотя она высказывалась многими авторами, не поддерживается в поисковых моделях, используемых в Semantic web и являющихся наиболее развитыми моделями семантического поиска.

Наиболее распространённые модели поиска Semantic Web используют для семантических запросов конечный набор операций, которые работают только на основе графовых шаблонов поиска. Такие шаблоны неплохо работают на небольших семантических сетях, на которых легко задать конфигурацию искомого графа. Однако в web-сайтах уровня корпорации, объёмы информации таковы, что возрастает необходимость в сложных запросах, в которых количество элементов шаблона может определяться десятками тысяч элементов. Поэтому появляется необходимость принципиально новых подходов к организации поиска на семантических сетях.

В настоящее время одним из популярных решений указанных выше трудностей является совершенствование языка SPARQL, предложенного сообществом Semantic web, в направлении

расширения его синтаксиса таким образом, чтобы описывать графовый шаблон поиска на основании регулярного выражения или некоторой графовой грамматики [Anyanwu, 2007], [Alkhateeb, 2009], [Pérez, 2010]. Однако такой подход предполагает, что разработчик должен знать топологию семантической сети. Данная информация не всегда бывает доступна разработчику интеллектуальной системы. Поэтому нам кажется, что для построения навигационной поисковой модели для семантически структурированных web-сайтов наиболее эффективен подход, который используется в вопросно-ответных системах [Сулейманов, 2001]. Такой подход дает возможность строить машину поиска как семейство поисковых операций, которые работают с семантической спецификацией вопроса (запроса) пользователя. Типология операций не ограничивается операциями поиска по графовому шаблону, а может быть расширена до любых классов операций, например, таких как операции логического вывода.

Модель навигационного поиска для семантически структурированных web-сайтов, построенных на основе модели (1), зададим следующим образом:

$$M_s = \{L_q, L_a, W\}, \quad (3)$$

где

L_q – язык вопросов, с помощью которого в виде формальных текстов осуществляется описание (спецификация) запрашиваемых (искомых) фрагментов семантической сети;

L_a – язык представления ответов, с помощью которого осуществляется выделение формальных текстов, являющихся ответами, и описание их связи с формальными текстами, которые представляют вопросы, соответствующие указанным ответам;

W – семейство информационно-поисковых операций, каждая из которых реагирует на соответствующий ей семантический тип вопроса и выполняет соответствующую ей поисковую процедуру на семантической сети.

Модель информационного поиска (3) относится к классу моделей вопросно-ответного поиска и позволяет обеспечить семантически структурированный web-сайт базовыми средствами навигационного поиска.

В основе языка вопросов L_q поисковой модели (3) предлагается использовать формальную теорию вопросов [Белнап, 1981]. На этой теории для предлагаемой в работе поисковой модели разработана типология вопросов, которая определяет семантическую мощность предлагаемых средств поиска и задает базовые навигационные операции поисковой модели.

В типологии вопросов выделены следующие типы вопросов:

- *запрос всех элементов множества*: чаще всего заданное множество – это конечное множество из элементов некоторой структуры;

- *запрос внешней информационной конструкции*, представленной некоторым файлом в том или ином формате;

- *что-это-вопрос*, запрашивающий основные сведения об указываемой сущности – семантической окрестности, "центром" которой является знак указываемой сущности.

В рамках данной модели поиска для каждого типа вопроса можно поставить в соответствие некоторый тип ответа. Такие типы ответов определяются структурой представления ответа и видами знаний, которые в них представлены. В общем случае между типами вопросов и ответов нет взаимно-однозначного соответствия. Это означает, что на один и тот же вопрос могут быть получены различные типы ответов.

В современных вопросно-ответных системах принято, что ответ на заданный вопрос формулируется в "естественно-языковой манере вопроса" [Сулейманов, 2001]. В рамках предлагаемой модели эта идея может быть выражена формой представления ответа на некотором внешнем языке, который понятен пользователю семантически структурированного web-сайта и используется для представления информации на его страницах.

Поисковая машина начинает работать в процессе сёрфинга пользователя семантически структурированного web-сайта. Каждый переход по ссылке к другой странице, инициирует операцию, которая работает в соответствии со следующим алгоритмом:

Шаг 1. Анализируется вопрос, поступивший на вход поисковой системе. Вопрос представлен на языке вопросов, который позволяет осуществить его семантическую спецификацию. Анализ семантической спецификации позволяет выявить следующие характеристики вопроса: тип задаваемого вопроса, наличие ответа на данный вопрос (если данный вопрос уже формулировался прежде), входные данные для поисковой операции, которая соответствует вопросу данного класса;

Шаг 2. На основе семантической спецификации, в которую входят входные данные для операции, информация о типе операции, происходит выбор операции из семейства операций поисковой машины. После поиска операции генерируется условие инициирования операции

Шаг 3. Выполнение поисковой операции и формировании ответа на поставленный вопрос.

Шаг 4. Чистка семантической сети от сгенерированных поисковой операцией вспомогательных информационных конструкций.

Шаг 5. Уточнение типа ответа для поставленного перед поисковой машиной вопроса, для этого анализируется семантическая спецификация вопроса, если информации о предпочтительном классе ответа нет, то используется прямой ответ.

Шаг 6. Формирование визуального представления вопроса, то есть генерация web-

статьи содержащей ответ на вопрос.

Семейство информационно-поисковых операций W задает набор процедур поиска в соответствии с набором типов вопросов и для базовой модели семантического информационного поиска выглядит следующим образом:

- *операция поиска всех элементов заданного конечного множества* (чаще всего – это множество из элементов некоторой структуры, статей СС-сайта одного класса);

- *операция поиска внешней информационной конструкции*, представленной некоторым файлом в том или ином формате;

- *операция поиска семантической окрестности сущности*, запрашивающая основные сведения об указываемой сущности – семантической окрестности, "центром" которой является знак указываемой сущности.

Каждая из указанных операций описывается на основании графового шаблона поиска, который специфицируется с помощью средств SC-кода.

2. Связь модели web-сайта с платформой Semantic web

Для разметки страниц web-сайта в соответствии с приведенной моделью языков разметки, которые используются в настоящее время не достаточно. В частности представление информации на страницах web-сайта с использованием модели RDF зачастую является сложным и избыточным [Хорошевский, 2011]. А для класса web-сайтов, которые используются в рамках данной работы, не обеспечивает необходимой выразительной мощности. Исходя из приведенных соображений, предлагается использовать язык разметки семантически структурированных гипертекстов.

2.1. Язык разметки семантически-структурированных гипертекстов для представления web-страниц

В качестве языка разметки гипертекстов, в основе которых лежат семантические сети, предлагается использовать SCnML (SCn Markup Language) – язык разметки семантически структурированных гипертекстов [Голенков, 2011], который задается следующим образом:

$$M_L = \{T_e, T_c, T_{comp}, T_q, R\},$$

где

T_e , – множество *корневых тегов описываемых сущностей* – множество текстовых обозначений сущностей предметной области. Описание каждой такой корневой сущности представляет собой SCnML-текст, соответствующий некоторой SCn-статье.

T_c – множество *тегов связей* – множество текстовых представлений обозначений связей отношений, используемых в рамках данной SCn-статьи. Такие связки отношений соединяют сущности, описываемые в рамках SCn-статьи с

другими сущностями, обозначения которых представлены в SCn-статье. Множество T_c задается следующим образом:

$$T_c = \{T_{c1}, T_{c2}\}$$

T_{c1} – множество **тегов однокомпонентных связей** – множество текстовых представлений связей отношений, каждая из которых в рамках SCn-статьи всегда связывает сущность с одной и только одной сущностью, обозначение которой присутствует в рамках данной SCn-статьи. Множество T_{c1} задается следующим образом:

$$T_{c1} = \{T_{c1s}, T_{c1r}\},$$

где

T_{c1s} – множество тегов однокомпонентных связей без указанной роли компонента связки отношения, которую они играют в рамках данной связки отношения.

T_{c1r} – множество тегов однокомпонентных связей с указанной ролью компонента связки отношения, которую они играют в рамках данной связки отношения.

T_{c2} – множество **тегов многокомпонентных связей** – множество текстовых представлений связей отношений, каждая из которых в рамках SCn-статьи всегда связывает сущность с одной или более сущностями, обозначения которых присутствуют в рамках данной SCn-статьи.

T_{comp} – множество **тегов компонентов связей** – множество текстовых представлений компонентов отношений, каждый из которых в рамках SCn-статьи является обозначением сущности, связанной с корневой сущностью отношением, тип которого задается соответствующим тегом многокомпонентной связки. Множество T_{comp} задается следующим образом:

$$T_{comp} = \{T_{comp1}, T_{comp2}\}$$

T_{comp1} – множество **тегов компонентов связей без указания роли компонента связки** – множество текстовых представлений знаков сущностей, связанных с сущностями, встречаемыми в рамках SCn-статьи.

T_{comp2} – множество **тегов компонентов связей с указанием роли каждого компонента** – множество текстовых представлений компонентов связей отношений, которые являются обозначением сущностей, связанных с некоторыми сущностями, встречаемыми в рамках SCn-статьи, каждый компонент, связки которого уточняется дополнительно ролью.

T_q – множество **тегов SCnML-запроса** – множество текстовых представлений спецификаций семантических запросов к web-ресурсу, разработанному с помощью семантически структурированных гипертекстов (см. листинг ниже и рисунок 2). Использование таких тегов в SCnML-текстах позволяет исключить дублирование информации в рамках различных SCnML-текстах за счет того, что результаты семантических запросов, задаваемых с помощью данного тега, включаются в

SCn-статью и, тем самым, позволяют расширить спецификацию сущности, описываемой в рамках SCn-статьи.

R – множество правил размещения тегов при записи SCnML-текстов. На множестве SCnML-тегов заданы следующие типы отношений: **быть корневым тегом***, **быть родительским тегом***. Введены следующие правила формирования SCnML-текстов:

- каждый SCnML-тег связан с другим SCnML-тегом понятием уровня. В рамках данной работы уровень показывает «через сколько отношений» связана корневая сущность с некоторой другой сущностью в рамках данной статьи. Уровень позволяет задать отношение между родительским и дочерним тегом и определяет, для какой сущности определены связки отношений, расположенные на один уровень ниже, и к какой

OSTIS-2013

= Конференция OSTIS-2013

= международная конференция

= OSTIS-2012

§ Время проведения:

• с_:2013.02.21

• по_:2013.02.23

§ Программный комитет:

• Гаврилова Т. А.
• Голенков В. В.
• Гулякина Н. А.
• Грибова В. В.
• Ефименко И. В.
• Заболеева-Зотова А. В.
• Загоруйко Ю. А.
• Иванюк А. А.

полный состав программного комитета...

§ Пункты питания участников:

• Ресторан "ДжоМалунама"
• Ресторан-бистро "Лидо"
• Кафе "Жбан"
• Столовая "Амкодор"

Результат работы запроса поиска элементов множества программный комитет -2013

Рисунок 2 – SCn-статья, сформированная системой.

связке относятся сущности, являющиеся компонентами связей;

- на одной web-странице могут располагаться SCnML-тексты, соответствующим нескольким SCn-статьям;

- SCnML-текст, соответствующий одной SCn-статье может входить в состав SCnML-текста, соответствующего другой SCn-статье;

- теги в рамках SCnML-текста записываются по следующим правилам:

- первым следует корневой тег, тег корневой сущности;

- за корневым тегом может следовать только тег связки;

- тегу компонента связки всегда предшествует тег связки.

Каждой SCn-статье в рамках семантически структурированного web-сайта соответствует некоторый SC-граф, семантически эквивалентный ей. Ввиду того, что SCnML – это средство разметки SCn-статей, любому SCnML-тексту также соответствует некоторый SC-граф.

Подход к разметке web-страниц на основе SCnML позволяет обеспечить независимость семантической модели предметной области от языка разметки при верстке страниц с семантически-структурированным гипертекстом.

Листинг, приведенный ниже, представляет фрагмент SCnML-текста, реализованный с помощью средств wiki-разметки и соответствующий фрагменту SCn-статьи, использованной при спецификации сущности “Конференция OSTIS-2013”.

```

{{SCnFieldConcept|OSTIS-2013}}
{{SCnFieldSpecConSyn|1|Конференция OSTIS-2013}}
{{SCnFieldSpecConMemberSet|1|международная конференция}}
{{SCnFieldSpecConMemberEl|2|OSTIS-2012}}
{{SCnFieldSpecConDate|1}}
{{SCnFieldCompEnumWithAttr|2|1|c|2013.02.21|SCnFieldSpecConDateBegin}}
{{SCnFieldCompEnumWithAttr|2|1|по|2013.02.23|SCnFieldSpecConDateEnd}}
{{SCnFieldSpecConProgramCommittee|1}}
{{SCnFieldCompQueryEnum|2|программный комитет OSTIS-2013|полный состав программного комитета ...|SCnFieldSpecConProgramCommittee}}
{{SCnFieldSpecConFoodItem|1}}
{{SCnFieldCompEnum|2|1|Ресторан "ДжоМалунгма"|SCnFieldSpecConFoodItem}}
{{SCnFieldCompEnum|2|1|Ресторан-бистро "Лидо"|SCnFieldSpecConFoodItem}}
{{SCnFieldCompEnum|2|1|Кафе}}

```

```

"Жбан"|SCnFieldSpecConFoodItem}}
{{SCnFieldCompEnum|2|1|Столовая
"Амкодор"|SCnFieldSpecConFoodItem}}

```

В листинге с помощью тега *SCnFieldConcept* обозначается корневая сущность. Теги в листинге, начинающиеся с префикса *SCnFieldSpecCon*, являются тегами связей соответствующих отношений: *SCnFieldSpecConSyn* – тег однокомпонентной связи отношения синонимии; *SCnFieldSpecConMemberSet* – тег однокомпонентной связи отношения “быть элементом множества”; *SCnFieldSpecConDate* – тег многокомпонентной связи отношения “время проведения”; *SCnFieldSpecConProgramCommittee* – тег многокомпонентной связи отношения “программный комитет”; *SCnFieldSpecConFoodItem* – тег многокомпонентной связи отношения “пункты питания”. Теги, начинающиеся с префикса *SCnFieldComp*, являются тегами компонентов связей соответствующих отношений: *SCnFieldCompEnumWithAttr* – тег компонента многокомпонентной связи, с возможностью указания роли каждого компонента; *SCnFieldCompQueryEnum* – тег SCnML-запроса, результат которого будет выдаваться в виде набора тегов компонентов связей без указания роли каждого компонента; *SCnFieldCompEnum* – тег компонента многокомпонентной связи. У каждого тега связи и тега компонента связи указывается атрибут, являющийся номером уровня. В рамках данного примера номер уровня указывается вторым атрибутом, нумерация начинается с единицы.

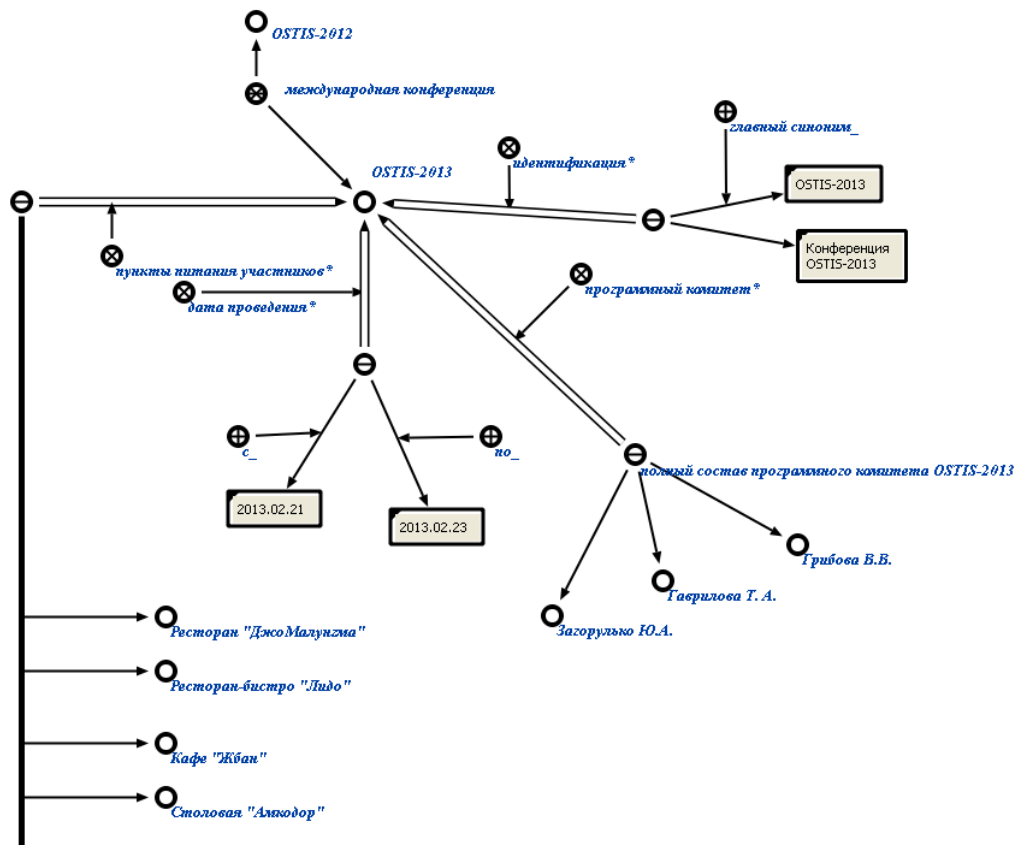


Рисунок 3 – SC-граф, который задает рассматриваемый листинг

После обработки данного листинга инструментальными средствами, на базе платформы Mediawiki, будет сгенерирована SCn-страница, приведенная на рисунке 2.

Каждой однокомпонентной связке соответствует обозначение на SCn-странице специальным маркером. Все многокомпонентные связи отображаются одним маркером и различаются лишь идентификатором отношения, экземпляр которого они обозначают. Роли компонентов связей указываются через двоеточие перед соответствующим компонентом связи, при этом каждое обозначение роли дополнено знаком “_”. Как будет выглядеть SC-граф рассматриваемого примера показано на рис. 3.

2.2. Преобразование семантически-структурированных гипертекстов в RDF-графы

В основе метода преобразования лежит представление элементов алфавита SC-кода с помощью одного из базовых словарей RDF в рамках некоторого инструментального средства, поддерживающего RDF. Метод отображения SC-моделей на RDF-модель I_{SC} задается следующим образом:

$$I_{SC} = \{S_{SC}, ARDF, R\},$$

где

S_{SC} – множество SC-текстов прикладной SC-модели некоторого web-сайта. Такие SC-тексты могут включать, в том числе и тексты программ обработки SC-текстов, представленных средствами SC-кода.

$ARDF$ – множество SC-элементов, представленных в терминах средств метаописаний Semantic Web – RDFS, OWL или OWL2.

R – множество правил перехода от SC-текстов к RDF-текстам. Такие правила оформляются в соответствии с некоторым синтаксическим представлением как SC-текстов, так и RDF-текстов.

Для каждого элемента алфавита SC-кода составляется правило преобразования. Приведем примеры таких правил (таблица 1) для некоторых элементов алфавита SC-кода, которые характеризуют особенности представления данного способа кодирования семантических сетей в RDF.

Из таблицы 1 видно, что для обозначения как sc-узлов, так и sc-дуг в RDF-графе используются узлы. Такой подход обусловлен непригодностью RDF-модели к представлению множественных отношений и связей между узлами [Трофимов, 2011]. Преобразование на основе указанных в таблице 1 правил дает возможность хранить тексты SC-кода в современных RDF-хранилищах.

В качестве базовых предикатов для представления в RDF семантических сетей в SC-коде введены две связи отношения инцидентности – **компонент_sc-коннектора**, **второй_компонент_sc-коннектора**; в качестве

Таблица 1 – Правила преобразования элементов алфавита SC-кода в RDF-представление с помощью RDFS (нотация N3)

<p>Элемент узлового типа (sc-узел)</p> <pre>@prefix: <http://www.ostis.net/> . @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> . @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> . :sc-элемент a rdfs:Class . :sc-узел a rdfs:Class; rdfs:subClassOf :sc-элемент . :xi rdfs:subClassOf :sc-узел.</pre>
<p>Элемент дугового неориентированного типа (sc-ребро)</p> <pre>@prefix: <http://www.ostis.net/> . @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> . @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> . :sc-элемент a rdfs:Class . :sc-коннектор a rdfs:Class ; rdfs:subClassOf :sc-элемент . :sc-ребро a rdfs:Class ; rdfs:subClassOf :sc-коннектор . :компонент_sc-коннектора a rdf:Property ; rdfs:domain :sc-элемент ; rdfs:range :sc-коннектор . :e rdfs:subClassOf :sc-ребро. :xi :компонент_sc-коннектора :e. :xj :компонент_sc-коннектора :e.</pre>
<p>Элемент дугового ориентированного типа (sc-дуга общего вида)</p> <pre>@prefix: <http://www.ostis.net/> . @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> . @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> . :sc-элемент a rdfs:Class . :sc-коннектор a rdfs:Class ; rdfs:subClassOf :sc-элемент . :sc-дуга_общего_вида a rdfs:Class ; rdfs:subClassOf :sc-коннектор . :компонент_sc-коннектора a rdf:Property ; rdfs:domain :sc-элемент ; rdfs:range :sc-коннектор . :второй_компонент_sc-коннектора a rdf:Property ; rdfs:domain :sc-элемент ; rdfs:range :sc-коннектор . :e rdfs:subClassOf :sc-дуга_общего_вида. :xi :компонент_sc-коннектора :e. :xj :второй_компонент_sc-коннектора :e.</pre>

базовых классов элементов семантической сети введены классы RDF-сущностей, которые соответствуют элементам алфавита SC-кода – **sc-элемент**, **sc-коннектор**, **sc-узел**, **sc-ссылка**, **sc-коннектор**, **sc-ребро**, **sc-дуга общего вида**. Для обеспечения унифицированного представления текстов SC-кода на основе структурно-семантической типологии элементов SC-кода и на основе RDFS-словаря разработана онтология элементов SC-кода.

Учитывая сказанное, алгоритм отображения прикладной SC-модели на RDF-модель в соответствии с предлагаемым методом выглядит следующим образом:

Шаг 1. Описание прикладной модели семантически структурированного web-сайта на каком-либо из SC-языков.

Шаг 2. Выбор одного из базовых RDF-словарей для отображения на платформу Semantic Web элементов SC-кода (уточнение какой из RDF-словарей мы будем использовать, RDFS, OWL или OWL 2). Выбор такого словаря определяется практической целесообразностью.

Шаг 3. Определение предметных словарей, которые будут использованы для представления прикладной модели.

Шаг 4. Реализация правил перехода от выбранного синтаксического представления SC-текстов к выбранному синтаксическому представлению RDF-текстов.

Шаг 5. Тестирование прикладной RDF-модели семантически структурированного web-сайта.

3. Апробация модели семантически структурированного web-сайта

На основании анализа современных инструментариев, используемых при построении семантических систем, был сформулирован ряд ключевых принципов разработки семантически структурированных web-сайтов:

- процесс разработки интеллектуальных систем – процесс коллективный и требующий постоянного согласования, поэтому должна быть использована гибкая и расширяемая платформа, ориентированная на коллективную работу;

- процесс разработки интеллектуальных систем должен протекать открыто и к такому процессу должен привлекаться как можно более широкий круг участников. Поэтому используемые средства должны быть открытыми и легко доступными;

- платформа разработки семантических и интеллектуальных систем должна обеспечивать простой интерфейс для подключения механизмов удаленного управления контентом для того, чтобы была возможность удаленной автоматизированной обработки контента.

Сформулированные принципы стали определяющими при выборе программных средств разработки семантически структурированных сайтов. В качестве платформы проведенных разработок, была выбрана известная и популярная платформа Mediawiki версии 1.16.2. Платформа Mediawiki является легко расширяемой платформой для коллективной разработки текстовых документов любого вида, поддерживающая весь цикл разработки документа, включающий хранение различных версий документа, средства редактирования документов, удобные внутренние средства разметки документов для размещения в сети Internet. Для удаленной работы с документами платформа Mediawiki предоставляет сервис-ориентированное API, разработанное на базе REST-архитектуры. Основным языком разработки платформы является язык PHP, который в настоящее время является одним из основных средств разработки web-сайтов для сети Internet. В качестве платформы для хранения данных Mediawiki использует несколько различных современных СУБД, среди которых MySQL, PostgreSQL, SQLite. Выбранная платформа эффективно используется во многих Open source проектах, самым известным, из которых, является

проект Wikipedia, развиваемый сообществом Wikimedia Foundation.

Для апробации результатов, описанных в рамках данной работы, в выбранной платформе были разработаны программные средства, которые обеспечивают представления сложноструктурированной информации с помощью языка SCnML. Для обеспечения переноса содержимого web-сайта на другие CMS, которые поддерживают язык OWL, был разработан специальный транслятор, работающий через REST API Mediawiki и позволяющий получать содержимое web-сайта в виде текстов языка OWL. Указанные программные средства были опробованы при разработке интеллектуальной справочной системы по геометрии и сайте конференций OSTIS.

ЗАКЛЮЧЕНИЕ

В работе описан подход к построению web-сайтов на основе семантических сетей, которые характеризуются наличием сложноструктурированного контента. Результаты, описанные в работе, могут быть использованы при решении, задач разработки прикладных интеллектуальных систем, в качестве вспомогательных средств коллективной разработки баз знаний, а также для повышения качества информационной поддержки традиционных web-сайтов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Микроформаты, 2012] Вики о микроформатах [Электронный ресурс]. – 2012. - Режим доступа: http://microformats.org/wiki/Main_Page-ru. – Дата доступа: 30.11.2012
- [Микроданные, 2012] Schema.org [Электронный ресурс]. – 2012. - Режим доступа: <http://schema.org/>. – Дата доступа: 30.11.2012
- [RDFa, 2012] Adida, Ben RDFa in XHTML: Syntax and Processing A collection of attributes and processing rules for extending XHTML to support RDF /Ben Adida [et al] //W3C Working Group Note 14 October 2008 [Электронный ресурс]. – 2012. - Режим доступа: <http://www.w3.org/TR/2008/REC-RDFa-syntax-20081014> – Дата доступа: 30.11.2012
- [SMW+, 2012] Semantic Enterprise Wiki Homepage [Электронный ресурс]. – 2012. - Режим доступа: http://www.smwplus.com/index.php/Semantic_MediaWiki_Plus – Дата доступа: 30.11.2012
- [SMW, 2012] Semantic Wiki Homepage [Электронный ресурс]. – 2012. - Режим доступа: <http://semantic-mediawiki.org/>
- [KnowWE, 2012] KnowWE – Knowledge Wiki Environment ! [Электронный ресурс]. – 2012.– Режим доступа: <http://sourceforge.net/projects/knowwe/> – Дата доступа: 30.11.2012
- [Колб, 2009] Колб, Д.Г Средства просмотра баз знаний интеллектуальных систем/ Д.Г. Колб//Вестник БрГТУ.–Минск, 2009. №5 (59).– С. 58-62
- [Белнап, 1981] Белнап, Н. Логика вопросов и ответов/ Н. Белнап, Т. Стил // Москва – “Прогресс”, 1981, 288 стр.
- [Сулейманов, 2001] Сулейманов, Дж.Ш. Исследование базовых принципов построения семантического интерпретатора вопросно-ответных текстов на естественном языке в АОС // Education Technology & Society 4(3). 2001.- с. 178-192
- [Anyanwu, 2007] Anyanwu K. SPARQL2: Towards Support for Subgraph Extraction Queries in RDF Databases./K. Anyanwu, A. Maduko, A. Sheth. // WWW 2007, P. 797–806.
- [Alkhateeb, 2009] Alkhateeb, F., Extending SPARQL with regular expression patterns (for querying RDF)/ F. Alkhateeb, J.-F.

Baget, J. Euzenat.// In Web Semantics: science, Services and Agents on the World Wide Web. – 2009. – № 7(2). – P 57–73

[Pérez, 2010] Pérez, Jorge nSPARQL: A Navigational Language for RDF/ Jorge Pérez, Marcelo Arenas, Claudio Gutierrez// Journal of Web Semantics. – 2010, 8(4). – P. 255-270.

[Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В. Ф. Хорошевский // Искусственный интеллект и принятие решений. - 2008. - № 1. - С.80-97.

[Голенков, 2011] Голенков, В. В. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н. А. Гулякина //Материалы международной научно-технической конференции OSTIS-2011, Мн.: БГУИР, 2011. - 21-58 стр.

[Трофимов, 2011] Трофимов, И. В. Эволюция выразительных способностей языка OWL /И. В. Трофимов // Программные системы: теория и приложения : электрон. научн. журн.2011. № 4(8), с. 85–94. URL: http://psta.psiras.ru/read/psta2011_4_85-94.pdf

DEVELOPMENT OF WEB-SITES WITH A COMPLEX CONTENT BASED ON SEMANTIC NETWORKS

Kolb D.G., Furman O.D.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

kolb@bsuir.by

olgasharpio@gmail.com

The approach to development of semantic structured web sites are offered. The model of web sites of this class is briefly described, the idea the organization of navigation search in web sites of this class is considered. The method for mapping of a class of the semantic networks used for development of semantic structured web sites, to the RDF model is offered.

INTRODUCTION

The problem of development of semantic web sites even more often enters into the sphere of interests of many scientists of the whole world. On the one hand the solution of this problem will allow to pass to new qualitative level of the World Wide Web – Semantic Web, on the other hand will allow to overcome one more step on a way to creation of intellectual systems and to put a platform for creation of intellectual systems for Internet.

Life cycle of process of development of web sites includes the following main objectives:

- filling of a content of a web site (including multimedia);
- optimization of a content for search engines;
- development of tools of navigation and search in a web site;
- providing shipping of a content of a web site.

Modern WCMS solve the specified problems with various level of completeness, however there is also a number of not solved tasks which decision can't be received without use of methods and artificial intelligence techniques.

MAIN PART

Considering solutions which are proposed now for improvement of web sites with a complex content, it should be noted two significant directions of works:

1. development of methods and tools of the specification of pages of a web site which allow to carry out easy indexation of web pages and, respectively, to provide more relevant search;

2. development of methods and tools of the specification of information which allow to represent complex types of information, including, such as theorems, axioms, logical statements and etc.

There is a dilemma – to use expressionless tools which are adapted for problems of industrial development of web sites or to use tools possessing the bigger expressive capacity, however not adapted for industrial development.

Within this work authors will try to propose the solution which will allow to integrate two directions of development of web sites given above, on the basis of use of models of representation of knowledge of higher abstract level. Such approach from the point of view of authors will allow to receive with one the intermediate decision for applied industrial development of web sites on the basis of semantic networks. Also will give an impetus to an evolution of development tools of the web sites based on semantic networks existing now.

Semantic structured web sites we will call a special class of the intellectual systems constructed on the basis of semantic networks and hypertext model of representation of information. The model of semantic structured web site is set as follows:

$$M = \{O, S_A, S_P, O_{hyp}, M_L, M_S, I\}, \quad (1)$$

where

O – ontology of web site's subject area;

S_A – a set of classes of articles of semantic structured web site (atomic semantic units);

S_P – a set of classes of pages of semantic structured web site, each of which consists of some set of classes of articles and each element which is connected with other element the sequence relation;

O_{hyp} – ontology of multimedia documents;

M_L – language tools of submission of information in the form of uniform semantic networks;

M_S – model of semantic navigation search on semantic structured web site;

I – method for mapping of model of semantic structured web site to the RDF model.

CONCLUSION

In work approach to creation of web sites on the basis of semantic networks which are characterized by existence of a complex content is described. Results described in article can be used at the decision, problems of development of applied intellectual systems, as supportive applications of collective development of knowledge bases, and also for improvement of quality of information support of traditional web sites.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

РЕАЛИЗАЦИЯ ХРАНИЛИЩА УНИФИЦИРОВАННЫХ СЕМАНТИЧЕСКИХ СЕТЕЙ

Корончик Д. Н.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

denis.koronchik@gmail.com

В работе описана одна из возможных реализаций хранилища унифицированных семантических сетей. Приводится описание принципов хранения элементов семантической сети и связей между этими элементами. Описанное хранилище может быть использовано для работы различных интеллектуальных систем в среде интернет.

Ключевые слова: графовые базы данных, унифицированные семантические сети, хранение семантических сетей.

ВВЕДЕНИЕ

Процесс проектирования интеллектуальной системы, с помощью технологии OSTIS [OSTIS, 2012], сводится к двум этапам: разработке логико-семантической модели (sc-модели) проектируемой системы и разработке интерпретатора этой модели. Может существовать достаточно большое количество интерпретаторов sc-моделей (в том числе и для одной платформы), они могут отличаться по функционалу, быстродействию и другим критериям. Все они входят в библиотеку компонентов и могут быть использованы многократно. В настоящее время ведутся работы по разработке интерпретатора sc-моделей, который ориентирован на работу в среде Интернет. Он состоит из следующих подсистем:

- программная реализация sc-хранилища (ориентирована на хранение sc-графов [Голенков, 2012]);
- программная реализация языка SCP (язык программирования ориентированный на обработку sc-графов [Голенков и др, 2001]);
- программная реализация компонентов ядра внешних интерфейсов (набор программных средств ориентированных на взаимодействие системы с внешней средой [Корончик, 2012]).

В данной работе более детально будут рассмотрены принципы, которые лежат в основе программной реализации sc-хранилища, так как оно является базовым компонентом всего

интерпретатора и взаимодействие всех подсистем осуществляется именно с его помощью.

Под sc-хранилищем понимается информационная подсистема, предназначенная для хранения sc-графов (тексты, записанные с помощью SC-кода, любой sc-граф представляет собой семантическую сеть). Основными функциями sc-хранилища являются:

- организация хранения sc-графов;
- предоставление программного интерфейса для добавления, удаления и извлечения хранимой информации (ассоциативный доступ);
- поддержка функций администрирования хранимой информации (распределение прав доступа);
- поддержка асинхронного выполнения запросов через программный интерфейс.

В настоящее время разрабатывается большое количество программных систем, которые также решают проблему хранения информации в виде графов. Это так называемые графовые базы данных [graph-database, 2012]. Наиболее популярные среди них Neo4j [Neo4j, 2012], HyperGraphDB [HyperGraphDB, 2012], BigData [bigdata, 2012]. Конечно можно использовать их для хранения sc-графов, но так как производительность sc-хранилища является важнейшим фактором влияющим на производительность системы в целом (так как все подсистемы взаимодействуют между собой через него), то указанные выше системы не достаточно эффективны, так как они изначально не учитывают специфику хранимого sc-графа.

1. Описание sc-хранилища

Так как sc-хранилище является частью интерпретатора sc-моделей ориентированного на работу в среде Интернет, то к нему предъявляются следующие требования:

- высокая производительность – минимизация времени затрачиваемого на выполнение операций добавления, удаления и доступа к хранимой информации;
- минимальные затраты памяти и дискового пространства для хранения sc-текстов;
- масштабируемость – возможность простого добавления вычислительных мощностей при увеличении нагрузки.

За хранение sc-графов в рамках sc-хранилища отвечает программная реализация sc-памяти. Она имеет сегментную адресацию, а адрес хранимого в ней sc-элемента будем называть sc-адресом. Использование сегментной адресации позволяет быстро получать доступ к sc-элементу по его адресу, а также минимизировать проблемы с ограничениями оперативной памяти (сегменты можно сохранять на диск и загружать их по мере надобности). От количества и размера сегментов зависит максимальное количество sc-элементов, которые можно хранить в sc-памяти.

В настоящее время максимальное количество сегментов и максимальное количество элементов, которые можно хранить в одном сегменте, равно 2^{16} . Это позволяет хранить в памяти до 2^{32} элементов. Таким образом, sc-адрес имеет размер 4 байта и состоит из двух частей: номер сегмента (2 байта), смещение в сегменте (2 байта).

Каждая ячейка sc-памяти хранит информацию об одном sc-элементе и имеет структуру, представленную в таблице 1.

Таблица 1 – Структура ячейки sc-памяти

Поле	Размер
Тип sc-элемента	2 байта
Временная метка создания	4 байта
Временная метка удаления	4 байта
sc-адрес первого коннектора в списке входящих в элемент коннекторов (для которых описываемый sc-элемент является вторым компонентом)	4 байта
sc-адрес первого коннектора в списке выходящих из элемента коннекторов (для которых описываемый sc-элемент является первым компонентом)	4 байта
Количество элементов в списке выходящих коннекторов	4 байта
Количество элементов в списке входящих коннекторов	4 байта
Дополнительные данные об элементе	33 байта

Как видно из таблицы 1, на хранение одного sc-элемента требуется 59 байт. Рассмотрим поля ячейки более подробно:

- **тип sc-элемента.** Каждому биту этого поля соответствует некоторый тип. Принадлежность к тому или иному типу sc-элемента указывается наличием 1 в соответствующем бите поля. Если

же значение поля равно нулю, то ячейка считается пустой. Полный список типов приведен в таблице 2.

Таблица 2 – Таблица битов кодирующих тип sc-элементов

Бит	Кодируемый тип или признак
1	sc-узел
2	sc-ссылка
3	sc-ребро
4	sc-дуга общего вида
5	sc-дуга принадлежности
6	sc-константа
7	sc-переменная
8	позитивная sc-дуга / узел, обозначающий n-арную связьку
9	негативная sc-дуга / узел, обозначающий структуру
10	нечеткая sc-дуга / узел, обозначающий ролевое отношение
11	нестационарная sc-дуга / узел, обозначающий бинарное, неролевое отношение
12	стационарная sc-дуга / узел, обозначающий понятие, не являющееся отношением
13	узел, обозначающий абстрактный объект
14	узел обозначающий материальный объект

Из таблицы видно, что начиная с восьмого, последующие биты могут кодировать различные признаки. Интерпретация кодируемого признака зависит лишь от того является ли хранимый элемент sc-дугой (первый признак) или же sc-узлом (второй признак). Работа с типами sc-элементов при такой организации их хранения сводится к простым побитовым “и” и “или”.

- **временная метка создания/удаления sc-элемента.** Данные поля используются для обеспечения асинхронного доступа к содержимому sc-памяти. Наличие в этих полях значения 0, означает, что элемент еще не был создан или же еще не удален;
- **sc-адрес первой входящего/выходящего коннектора.** Каждый sc-элемент имеет два списка инцидентности: по входящим и выходящим дугам. В этих полях хранятся sc-адреса первых sc-коннекторов для каждого из списков инцидентности;
- **количество элементов в списке входящих/выходящих коннекторов.** Эти поля используются для оптимизации доступа к элементам. К примеру, если нам надо найти sc-дугу выходящую из элемента *A* в и входящую в элемент *B*, то нам необходимо просмотреть либо список выходящих из элемента *A* или же список входящих в элемент *B* sc-коннекторов. Когда длина обоих списков известна, то проход лучше осуществить по наименьшему из них.

Хранение sc-ссылок

SC-ссылки – это sc-элемент, который обозначает либо определенный файл, который можно просматривать или в котором закодирована в определенном формате некоторая внешняя, инородная для SC-кода информационная конструкция, либо некоторую компьютерную систему, с которой можно взаимодействовать [Голенков, 2012]. Хранение содержимого sc-ссылки (путь к файлу, путь к ресурсу, адрес

информационной системы, содержимое файла и т. д.) храниться на файловой системе. Важным является то, что множество различных sc-ссылок могут ссылаться на разные файлы, которые имеют одинаковое содержимое. Было бы разумно не хранить содержимое одинаковых файлов дважды. Для этого при создании sc-ссылки и указании файла на который она ссылается, вычисляется hash-сумма содержимого с помощью алгоритма SHA256. В результате получается строка из 32 символов, которая и хранится в поле дополнительных данных для sc-ссылок. Само же содержимое копируется в файл (создаваемый sc-памятью рядом с сохраняемыми сегментами), путь к которому строится на основании hash-суммы. Рядом с этим файлом создается файл, в котором хранятся sc-адреса всех ссылок ссылающихся на файлы с изоморфным содержимым. Таким образом, для того чтобы найти все sc-ссылки ссылающиеся на файлы с указанным содержимым, нам необходимо лишь вычислить hash-сумму искомого образца и проверить наличие файла по пути вычисляемому из hash-суммы и если он существует, то вернуть список хранящихся sc-адресов.

Хранение sc-коннекторов

Сложностью при хранении sc-коннекторов является то, что нельзя предугадать, сколько их будет связано с элементом. Поэтому хранить отдельно для каждого элемента списки инцидентности за пределами сегментов не является разумным. Было решено использовать следующее свойство коннектора: *каждый коннектор всегда имеет начальный и конечный элемент*. Другими словами каждая sc-дуга входит в два списка смежности: выходящих и входящих дуг. На основании этого было решено хранить списки смежности прямо в сегментах, используя поле дополнительных данных для sc-коннекторов (таблица 3).

Таблица 3 – Структура ячейки с дополнительными данными для sc-коннекторов

Поле	Размер
sc-адрес элемента являющегося первым компонентом sc-коннектора	4 байта
sc-адрес элемента являющегося вторым компонентом sc-коннектора	4 байта
sc-адрес следующего коннектора в списке выходящих коннекторов	4 байта
sc-адрес следующего коннектора в списке входящих коннекторов	4 байта

Как видно из таблицы, каждый sc-коннектор хранит адрес следующего за ним в списке входящих и выходящих sc-коннекторов. Пример хранения списков инцидентности sc-коннекторов представлен на рисунке 1, где справа представлен хранимый граф. Каждый элемент на рисунке пронумерован от 1 до 7, где 1,4,2 и 3-й элементы – это sc-узлы, а 7, 6 и 5-й – sc-дуги. Ячейки, в которых хранятся элементы, изображены в виде небольших таблиц, с номером в первой строке. Под номером изображаются поля ячеек (не все поля элемента, а лишь необходимые для хранения списков

инцидентности) и их значения. Поясним эти поля:

- **first_out_arc** – sc-адрес первого коннектора в списке выходящих из указанного элемента коннекторов;
- **first_in_arc** – sc-адрес первого коннектора в списке входящих в указанный элемент коннекторов;
- **begin_addr** – sc-адрес элемента являющегося первым компонентом sc-коннектора;
- **end_addr** – sc-адрес элемента являющегося вторым компонентом sc-коннектора;
- **next_out_arc** – sc-адрес следующего выходящего в некоторый sc-элемент коннектора, (в указанном списке все sc-коннекторы имеют одинаковый первый компонент. Другими словами это список выходящих из некоторого sc-элемента дуг);
- **next_in_arc** – sc-адрес следующего входящего в некоторый sc-элемент, коннектора (в указанном списке все sc-коннекторы имеют одинаковый второй компонент. Другими словами это список дуг, входящих в некоторый sc-элемент)

Стрелками показаны переходы для списков инцидентности по выходящим sc-коннекторам для элементов 1 и 4. Аналогичным образом хранятся списки инцидентности для выходящих sc-коннекторов (дуг).

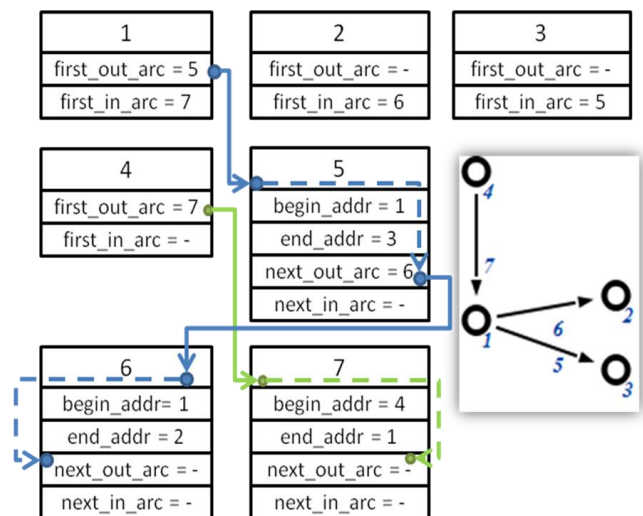


Рисунок 1 – Пример хранения списков инцидентности

Описанный выше метод хранения списков инцидентности позволяет значительно повысить плотность хранимой информации, а также обеспечить фиксированный размер сегментов. Очевидно, что при таком подходе всегда можно спрогнозировать необходимый объем памяти M необходимый для хранения того или иного sc-графа. Он всегда будет вычисляться по формуле 1.

$$M = N * S; \quad (1)$$

где N – количество sc-элементов в графе, S – размер ячейки (в байтах) необходимый для хранения одного sc-элемента

ЗАКЛЮЧЕНИЕ

Описанные выше приемы позволили реализовать первую версию sc-хранилища, которая имеет следующие характеристики по быстродействию:

- размер sc-элемента 64 байта, таким образом, размер сегмента 4 Мб. Размер полного sc-хранилища 256 Гб (использованы все ячейки);
- скорость заполнения sc-хранилища sc-узлами - 5116188.644108 узлов/сек;
- скорость заполнения sc-хранилища sc-дугами - 1833769.525978 sc-дуг/сек;
- скорость загрузки сегментов с диска в память – 305.56 сегмент/сек;
- скорость выгрузки сегментов из памяти на диск – 12 сегмент/сек;

Полученные данные отражают работоспособность sc-хранилища в однопоточном варианте исполнения, в качестве тестовой машины использовался ноутбук с 4 Гб оперативной памяти, процессором Intel Core i3 2.27 ГГц. Тесты проводились на операционной системе Ubuntu 12.04 32-bit.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Голенков и др, 2001] Представление и обработка знаний в графодинамических ассоциативных машинах /В. В. Голенков, [и др]; – Мн. : БГУИР, 2001

[Голенков, 2012] Графодинамические модели параллельной обработки знаний / В. В. Голенков, Н. А. Гулякина // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» - Минск, 2012

[Корончик, 2012] Семантические модели мультимодальных пользовательских интерфейсов и семантическая технология их проектирования / Д. Н. Корончик // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» - Минск, 2012

[bigdata, 2012] bigdata [Электронный ресурс]. – 2012. – Режим доступа: <http://www.bigdata.com/bigdata/blog/> - Дата доступа: 21.11.2012.

[graph-database, 2012] graph-database [Электронный ресурс]. – 2012. – Режим доступа: <http://www.graph-database.org> - Дата доступа: 20.11.2012.

[HyperGraphDB, 2012] HyperGraphDB – A Graph Database [Электронный ресурс]. – 2012. – Режим доступа: <http://www.hypergraphdb.org> – Дата доступа: 21.11.2012

[neo4j, 2012] Neo4j [Электронный ресурс]. – 2012. – Режим доступа: <http://www.neo4j.org> – Дата доступа: 20.11.2012.

[OSTIS, 2012] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2012. - Режим доступа: <http://ostis.net>. – Дата доступа: 10.11.2012

IMPLEMENTATION OF STORAGE FOR UNIFIED SEMANTIC NETWORKS

Koronchik D. N

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

denis.koronchik@gmail.com

This article describes storage for unified semantic networks. It contains description of principles, that used to store semantic networks oriented for processing.

INTRODUCTION

There are several systems developed today to handle the problem of storing large amounts of graph data. But for each type of data and set operations different systems differ in suitability. But all of them not efficient to store unified semantic networks that used to store information in OSTIS project.

MAIN PART

Development of intelligent system with OSTIS technology consists of two parts: development of unified logical model of system, development of interpreter for the last one. Storage of unified semantic networks (sc-storage) is a main part of model interpreter.

SC- storage is a software system that provides storage of sc-graphs (unified semantic networks described with SC-code). It used segment addressing. Address of each element of sc-graph (sc-element) consists of two parts: segment number, offset in segment. Each segment – is a fixed size array of cells. Cells used to store data just for one sc-element and have a fixed size – 59 bytes. There are three types of stored sc-elements: sc-node, sc-link, sc-connector. Each cell has common data, that suitable for all element types and 33 bytes to store type specific information. For example, last 33 bytes store first and second, components for sc-connectors. For sc-links this 33 bytes contains hash sum of data calculate by SHA256 algorithm.

CONCLUSION

Described principles were implemented in sc-storage prototype. The last one was implemented with C programming language. Each sc-element in sc-storage has 64 bytes size. So full size of data that can be stored in implemented sc-storage is 256Gb or (2^{32} sc-element).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.272.43+004.272.32

ПОСТРОЕНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ВЕКТОРНОГО ПРОБЛЕМНО-ОРИЕНТИРОВАННОГО ПРОЦЕССОРА

Вереник Н.Л., Татур М.М.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

nick.verenik@gmail.com

tatur@i-proc.com

В статье изложен подход к проектированию средств аппаратной поддержки вычислений при построении интеллектуальных систем на базе семантических сетей. В качестве научной основы принята концепция платформенной независимости верхнего логико-семантического уровня системы от ее аппаратной реализации. Рассмотрены основные этапы проектирования: разработка архитектуры векторного проблемно-ориентированного процессора, разработка программной модели и обоснование выбора элементной базы.

Ключевые слова: семантическая сеть; параллельный процессор; семантическая обработка информации.

ВВЕДЕНИЕ

В данной работе под семантической сетью понимается графовая структура, в которой определенным образом закодирована полезная информация. В целом, аппарат семантических сетей позволяет разрабатывать формальные алгоритмы процессов поиска и обработки информации посредством анализа и изменения состояний элементов и конфигурации семантической сети. Несмотря на широкую распространенность семантических сетей в качестве модели представления знаний, общепринятой теории, определяющей методы кодирования и переработки информации в семантических сетях, до сих пор не существует.

Современные интеллектуальные системы характеризуются большими объемами хранимой и перерабатываемой информации, многоуровневой иерархической структурой знаний, активным использованием метаинформации. Как следствие, резко возрастает сложность используемых семантических сетей, структура которых становится неоднородной и нерегулярной, а алгоритмы обработки информации основываются, как правило, на эвристиках, отличных от системы к системе. Все это обуславливает актуальность проблемы создания эффективной аппаратной платформы с параллельной архитектурой, ориентированной на класс задач семантической обработки.

Согласно ряду проведенных исследований

[Голенков и др., 2012], теоретическим фундаментом необходимой унификации может выступить концепция, основная идея которой заключается в платформенной независимости создаваемых методов, алгоритмов и программ семантической обработки информации (т.е. независимости верхнего уровня интеллектуальной системы) от их аппаратной реализации. При этом нижний уровень интеллектуальной системы представляется абстрактной графодинамической машиной (ГДМ), непосредственно осуществляющей обработку данных (рис. 1). Процесс обработки информации в ГДМ трактуется как графодинамический процесс, т.е. как процесс преобразования графовой структуры семантической сети, в ходе которого меняется не только состояние элементов структуры, но и ее конфигурация (появляются и удаляются вершины, изменяются связи между ними).

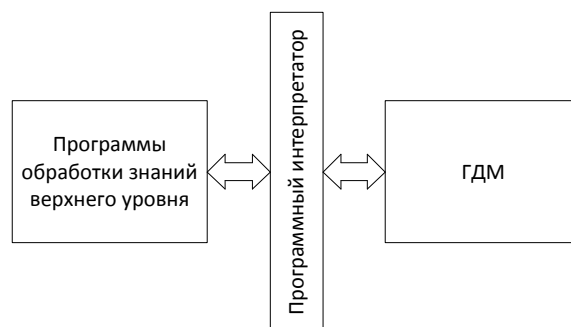


Рисунок 1 – Платформенно независимая интеллектуальная система

В качестве конкретной реализации абстрактной ГДМ может выступать любая вычислительная система с любой архитектурой, с любой аппаратной и программной платформами. Как следствие, их выбор будет влиять на трудоемкость программирования и производительность, которую они смогут обеспечить. Платформенная независимость достигается за счет программного интерпретатора, осуществляющего трансляцию программ семантической обработки верхнего уровня в формат команд конкретной ГДМ.

В настоящей работе рассматривается один из возможных вариантов реализации ГДМ с использованием оригинальной параллельной архитектуры.

1. Подход к построению архитектуры проблемно-ориентированного процессора

При разработке программ низкого уровня для ГДМ на основе мультитядерных РС, кластеров или суперЭВМ с универсальными параллельными архитектурами возникает ряд проблем связанных со сложностью, нерегулярностью и иерархичностью семантических сетей. В частности, распределение задач между процессорами и организация их взаимодействия, распределение данных между блоками памяти, высокая вычислительная сложность и нерегулярность вычислений создают неразрешимые проблемы при распараллеливании алгоритмов и, в конечном счете, приводят к эффекту замедления роста производительности при увеличении числа процессорных элементов, сформулированному в виде закономерностей Густавсона-Барсиса [Gustafson, 1988] и Амдаля-Уэра [Amdahl, 1967]. Так, по опыту эксплуатации системы «Эльбрус» при решении задач с нерегулярной структурой графа вычислений, эффективность обработки достигает максимума при работе четырех-пяти. При подключении большего количества процессоров эффективность начинает падать из-за растущих затрат на обеспечение взаимодействия процессоров.

В качестве альтернативы в [Байрак и др., 2012], [Вереник и др., 2012a], [Вереник и др., 2012b] предложен подход к построению архитектуры проблемно-ориентированных параллельных процессоров, который может быть использован для создания ряда специализированных устройств, реализующих ГДМ, сравнимых по производительности с суперЭВМ, а по стоимости – с универсальными мультитядерными компьютерами. Суть подхода состоит в формальной трансформации исходного графа семантической сети к некоторому регулярному графу, что позволит эффективно распараллелить вычислительные алгоритмы семантической обработки данных. В качестве типовой архитектуры ГДМ предлагается использовать SIMD-архитектуру магистрального типа (Single Instruction Multiply Data) с локальной

оперативной памятью для каждого процессорного элемента (рис. 2). Данная архитектура широко применяется во многих вычислительных системах и известна как одна из наиболее технологичных и простых в использовании.

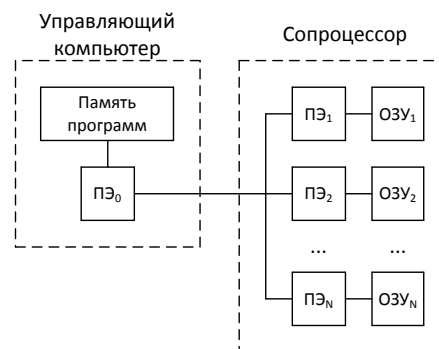


Рисунок 2 – Обобщенная блок-схема вычислительного комплекса

Фактически, знания интеллектуальной системы рассредоточены по множеству процессорных элементов (ПЭ) сопроцессора и имеют линейную регулярную структуру, соответствующую предложенному формату данных. Каждый ПЭ отвечает за доступ к единственному участку знаний системы, что позволяет исключить проблемы в организации межпроцессорного взаимодействия (устранение конфликтов доступа и т.п.). Функционально ПЭ сравним с простым компаратором, т.е. его аппаратная сложность минимизирована, чтобы позволить реализовать максимальное их количество на одном чипе.

Конструктивно вычислительный комплекс можно построить по одной из типовых схем, включающей в себя управляющий компьютер в качестве устройства управления и подключаемый сопроцессор, реализующий параллельные вычисления. Обмен данными между частями комплекса будет осуществляться по одному из стандартных типов интерфейсов.

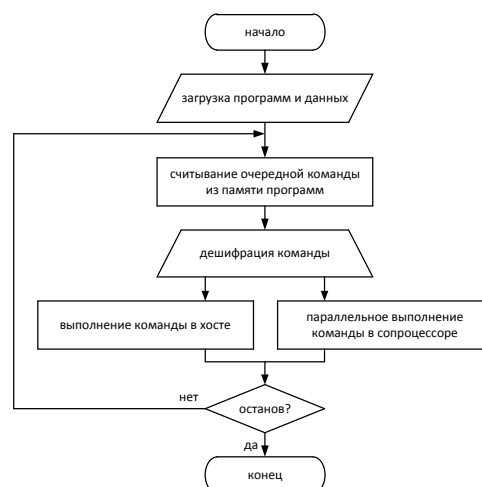


Рисунок 3 – Общий алгоритм функционирования вычислительного комплекса

В память управляющего компьютера записывается текст исполняемой программы (в терминах макрокоманд ГДМ), здесь же производится выборка команд, их дешифрация и передача для выполнения в сопроцессоре либо в хосте. На рис. 3 изображен общий алгоритм функционирования вычислительного комплекса.

Отличительной особенностью архитектуры является проблемная ориентация сопроцессора на операцию ассоциативного поиска. Операциям записи в этом устройстве соответствуют операции добавления, изменения либо удаления вершин или дуг графа, а операциям чтения – ассоциативный поиск данных по заданному шаблону. Отсутствие связей между ПЭ сопроцессора, а также организация локальной памяти обеспечивают линейную зависимость времени поиска от числа элементов ПЭ. Таким образом, достигается главная задача сопроцессора – обеспечение максимального быстродействия при выполнении операций ассоциативного поиска.

Из опыта проектирования сложных программно-аппаратных комплексов можно выделить следующие этапы разработки:

- разработка архитектуры и общих принципов функционирования;
- разработка программной модели и верификация архитектуры;
- выбор элементной базы, приобретение соответствующей аппаратной платформы, разработка макета;
- получение и оценка граничных технических характеристик;
- формирование ТЗ на разработку опытного образца; оптимизация технических решений под заданные технические условия.

2. Программная модель проблемно-ориентированного процессора

Прежде чем формулировать ТЗ на разработку аппаратного прототипа сопроцессора необходимо детализировать архитектуру и верифицировать алгоритмы вычислений. Для этого необходимо создать программную модель на РС, посредством которой выполнить ряд тестовых задач и проанализировать результаты выполнения.

Программная модель, в первую очередь, должна отражать совокупность функциональных возможностей (систему команд) вычислительного комплекса, которые прямо либо косвенно могут быть использованы программистом в ходе вычислений. Но также при разработке программной модели необходимо стремиться адекватно отразить ключевые блоки, межблочные информационные связи, форматы команд, способы размещения данных в памяти и доступа к ним. В ходе моделирования тестовая задача должна быть представлена так, как это ожидается в разрабатываемом вычислительном комплексе,

чтобы наиболее полно проверить принципы программного управления и ход параллельных вычислений, заложенные в архитектуру. В дальнейшем такая модель может стать основой программного эмулятора, позволяющего создавать и отлаживать прикладное программное обеспечение для вычислительного комплекса.

На рис. 4 представлена упрощенная диаграмма классов программы, позволяющей моделировать функциональные возможности и архитектурные свойства процессора. Ниже приведены ее основные особенности:

- текст исполняемой программы записывается в терминах макрокоманд процессора и хранится в отдельном текстовом файле;
- поддерживается программный интерфейс для интеграции интерпретатора с языка верхнего уровня;
- реализована возможность настройки атрибутов элементов графа (задавать как сам набор атрибутов, так и возможные принимаемые значения для каждого атрибута);
- реализована возможность настройки основных параметров процессора, таких как: размер локальной памяти, приходящийся на один ПЭ; количество ПЭ каждого типа; объем памяти системы и т.п.;
- все настройки системы хранятся во внешних файлах, позволяя легко управлять большим количеством различных конфигураций системы;
- существующая архитектура программной модели ориентирована на дальнейшую разработку программного обеспечения для разрабатываемого проблемно-ориентированного процессора.

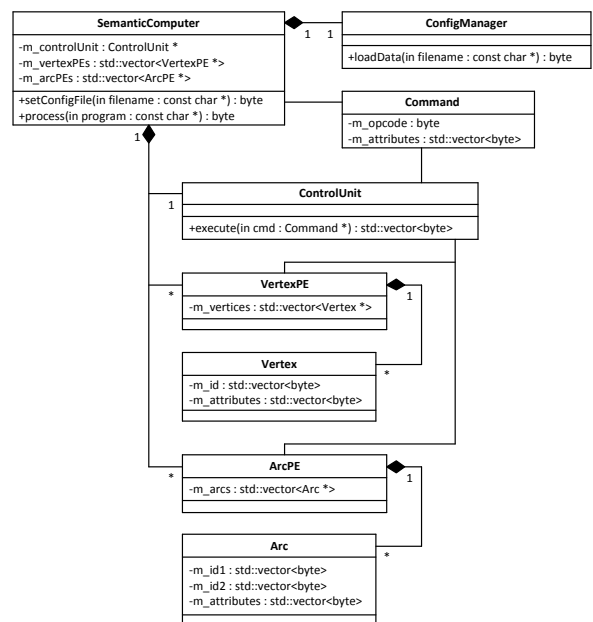


Рисунок 4 – Упрощенная диаграмма классов программной модели

3. ОБОСНОВАНИЕ ВЫБОРА ЭЛЕМЕНТНОЙ БАЗЫ И РАЗРАБОТКА МАКЕТА

В ходе этапа аппаратного проектирования в общем случае решаются следующие задачи:

- выбор элементной базы;
- выбор (приобретение или разработка) аппаратной платформы;
- отображение оригинальной архитектуры на аппаратной платформе (программирование «железа»).

Особенности элементной базы коренным образом будут отражаться на технических характеристиках разрабатываемого устройства. На сегодняшний день имеет смысл рассматривать следующие варианты элементной базы для аппаратной реализации сопроцессора.

1. FPGA или DSP (локальная память реализуется во внутренней блочной памяти). Подход имеет ограничения на объем внутренней блочной памяти (по сути, объем хранимой семантической сети), зато время доступа к памяти минимально, т.к. легко можно обеспечить доступ каждого ПЭ к своей локальной памяти. Данный вариант является наиболее простым при макетировании проблемно-ориентированных процессоров.

2. FPGA или DSP + внешние ОЗУ для реализации локальной памяти. Вариант не имеет принципиальных ограничений на объем локальной памяти, но время доступа к внешней памяти возрастает. Кроме того, для каждого внешнего ОЗУ необходим контроллер доступа к памяти, соответственно возрастают массогабаритные характеристики и энергопотребление устройства.

Замечание 1. По ряду очевидных причин не указан вариант разработки оригинальной элементной базы для реализации проблемно-ориентированных процессоров.

Замечание 2. Промежуточные и комбинированные конструктивные варианты, такие как каскадное включение FPGA или DSP, в работе не рассматривались из соображений простоты изложения.

Заключение

При создании прикладных интеллектуальных систем на основе семантических сетей существует проблема распараллеливания алгоритмов обработки информации при использовании многопроцессорных систем с классическими архитектурами. Предложен подход к решению проблемы на основе оригинальной проблемно-ориентированной архитектуры. Научным фундаментом выступает концепция платформенной независимости, при которой верхний уровень интеллектуальной системы разрабатывается независимо от нижнего (аппаратного) уровня.

В качестве аппаратной платформы

интеллектуальной системы предлагается разработать графодинамическую машину с архитектурой SIMD типа. Алгоритмической основой для такой архитектуры является предложенный механизм преобразования произвольной семантической сети в классический граф с линейной регулярной структурой. Полученный граф позволяет легко распараллелить вычислительные алгоритмы.

Рассмотрены этапы разработки перспективного вычислительного комплекса, включая создание программной модели проблемно-ориентированного процессора, а также краткое обоснование выбора элементной базы для создания макета.

Библиографический список

[Amdahl, 1967] Gene M. Amdahl. Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities / Gene M. Amdahl // AFIPS Conference Proceedings (30). – pp. 483–485.

[Gustafson, 1988] John L. Gustafson. Reevaluating Amdahl's Law / John L. Gustafson // Communications of the ACM 31(5). – pp. 532–533.

[Байрак и др., 2012] Параллельные процессоры для построения интеллектуальных систем / С. А. Байрак, Д. Н. Одинок, М. М. Татур, Ф. Филипов, М. Мунос // Открытые семантические технологии проектирования интеллектуальных систем : материалы II Междунар. научн.-техн. конф. (Минск, 16-18 февраля 2012 г.) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2012. – С. 135–140.

[Вереник и др., 2012a] Вереник Н. Л. Концепция построения графодинамической машины с SIMD-архитектурой / Н. Л. Вереник, М. М. Татур // Информационные технологии и системы 2012 (ИТС 2012) : материалы международной научной конференции, БГУИР, Минск, Беларусь, 24 октября 2012 г. / редкол. : Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2012. – С. 178–179.

[Вереник и др., 2012b] Вереник Н. Л. Разработка проблемно-ориентированных процессоров семантической обработки информации / Н. Л. Вереник, Е. Н. Сейткулов, М. М. Татур // Электроника инфо. – 2012. – № 8. – С. 95–98.

[Голенков и др., 2012] Голенков В. В. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования / В. В. Голенков, Н. А. Гулякина // Открытые семантические технологии проектирования интеллектуальных систем : материалы II Междунар. научн.-техн. конф. (Минск, 16-18 февраля 2012 г.) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2012. – С. 23–52.

BUILDING INTELLIGENT SYSTEM ON VECTOR ASIP

Nick L. Verenik, Mikhail M. Tatur

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

nick.verenik@gmail.com

tatur@i-proc.com

The article considers one of the approaches to design of computing hardware support while building applied intelligent system on semantic networks. The concept of platform independence of system logic-semantic high-level from their hardware implementation is proposed as a scientific basis. The primary design phases are given: vector ASIP development, programming model development, electronic components choice reasoning.

СЕКЦИЯ 3.

**СЕМАНТИЧЕСКИЕ МОДЕЛИ, СРЕДСТВА И МЕТОДЫ
КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ И ПАКЕТОВ
ПРОГРАММ, ОРИЕНТИРОВАННЫХ НА ОБРАБОТКУ ЗНАНИЙ**

SECTION 3.

**SEMANTIC MODELS, TOOLS AND METHODS OF COMPONENT
DESIGN KNOWLEDGE BASES AND PROGRAM PACKAGES FOR THE
PROCESSING OF KNOWLEDGE**



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

КОНЦЕПТУАЛЬНЫЕ «КАРКАСЫ» ПЛОХО ОПРЕДЕЛЕННЫХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ

Кулинич А.А.

Федеральное Государственное учреждение науки Институт проблем управления им В.А. Трапезникова, Российской академии наук, г. Москва, Российская Федерация

kulinich@ipu.ru

Исследуются модели представления знаний о плохо определенных предметных областях в виде концептуальных «каркасов», построенных на основе знаний об одном из объектов этой предметной области. Рассмотрены вопросы экспертного построения онтологии предметной области на основе ее концептуального «каркаса».

Ключевые слова: онтология; концептуальный «каркас»; плохо определенная предметная область.

ВВЕДЕНИЕ

В настоящее время в системах поддержки принятия решений (СППР) все более широкое применение находят формальные модели экспертных знаний, представленные онтологиями разных уровней. Онтологический подход в СППР расширяет возможности поддержки принятия решений при решении множества классов задач, множеством различных методов.

Важной особенностью онтологического подхода создания СППР является его открытость и возможность обработки разнородных данных, хранящихся различных базах данных или серверах сети Интернет. Открытость СППР, построенной на онтологических принципах выражается в возможности ее расширения, в том числе и за счет уже существующих онтологий предметных областей. Задачами поддержки принятия решений, решаемые в рамках онтологического подхода – это: поддержка выбора метода решения задач; классификация информации; задача интеграции разнородной информации; информационный поиск релевантной для поддержки принятия решения информации, интерпретация результатов решения задачи на естественном языке.

Узким местом при создании онтологической СППР в плохо изученной предметной области можно считать построение онтологий предметной области. Обычно онтологии строятся экспертом, но в последнее время появляются исследования автоматического или автоматизированного построения онтологий на основе анализа текстов, описывающих предметную область.

При автоматическом построении онтологии, на первом этапе построения осуществляется подбор текстов, отражающих закономерности предметной области. Далее применяют один из следующих подходов: полный лингвистический анализ, т.е. морфологический, синтаксический и семантический анализы текста; лексико-синтаксические шаблоны [Рабчевский, 2009], позволяющие выделить из текста основные понятия и отношения «Класс-Подкласс» («Isa») между ними; статистические методы, позволяющие выделить основные понятия и отношения «Класс-Подкласс» на множестве всех слов и словосочетаний с использованием эвристики [Мозжерина, 2011]; на основе продукций [Найханова, 2008] с применением генетического и автоматного программирования; на основе анализа формальных понятий (FCA) [Ganter, 1999].

Качество перечисленных подходов, опирающихся на анализ естественного языка, в значительной степени зависит от качества анализаторов текста. Известные теоретические трудности обработки естественного русского языка не позволяют говорить о высоком качестве построенных онтологий. Однако, онтологии предметных областей, получаемые таким способом могут быть использованы для поддержки работы экспертов при построении онтологий предметных областей.

Интересны работы по автоматизированному построению онтологий на основе экспертно сформированного представительного множества объектов предметной области. В работе [Baader, 2000] исследуются вопросы поддержки построения онтологии предметной области методом «Bottom-Top». Здесь выделяется представительное

множество объектов предметной области, описанных на языке дескриптивной логики (ДЛ). Для выделенного множества объектов строится формальный контекст, а затем концептуальная решетка понятий (онтология) методами формального анализа понятий [Ganter, 1999].

К сожалению, в работе [Baader, 2000] не исследованы вопросы формирования представительного множества объектов для построения онтологии. Но этот вопрос интересен для построения онтологий плохо определенных предметных областей. Например, можно ли построить онтологию предметной области по одному объекту из этой области?

Рассмотрим одну из возможных теоретических моделей представления экспертных знаний о предметной области в условиях неопределенности [Чечкин, 1992]. Здесь, все знания о предметной области представляются в виде множества сведений X . Любые сведения об объекте этой предметной области (информация о свойствах объекта, значениях этих свойств) называется элементарным сведением об объекте и определяется тройкой: $(p)\delta_i(x_0) \in X$, где $(p) \in [1, 0]$ – степень истинности наличия свойства δ_i у объекта x_0 . Если сведение истинно ($p=1$), то ее опускают при описании объекта, т.е. $(1)\delta_i(x_0) = \delta_i(x_0) \in X$.

Информация о том, что у объекта x_0 нет свойства δ_i , $\neg\delta_i(x_0)$ также считается сведением об объекте. Если у объекта x_0 имеются два свойства $\delta_1(x_0)$ и $\delta_2(x_0)$, то конъюнкция и дизъюнкция этих свойств $\delta_1(x_0) \wedge \delta_2(x_0)$, $\delta_1(x_0) \vee \delta_2(x_0)$ также будут сведениями об объекте.

Т.е., определены операции над элементарными сведениями $\delta_i(x_0)$ – (\wedge, \vee, \neg) , результаты которых также являются сведениями об объекте, $\delta_1(x_0) \wedge \delta_2(x_0) \in X$, $\delta_1(x_0) \vee \delta_2(x_0) \in X$. Утверждается, что все сведения об объекте образуют дистрибутивную решетку $(L(x_0), \wedge, \vee)$, которая называется решеткой понятий. Информацией об объекте считается не просто элементарные сведения об объекте $\{\delta_i(x_0)\}$, но и все возможные их обобщения, и сведения, полученные с помощью элементарных логических преобразований (\wedge, \vee, \neg) .

Таким образом, объект реального мира, определенный множеством элементарных сведений $\{\delta_i(x_0)\}$ порождает структуру сведений – решетку понятий $(L(x_0), \wedge, \vee)$, которая несет информацию не только об объекте x_0 , но и информацию о предметной области и ее структурной организации.

В этой работе исследуется вопрос поддержки построения онтологии плохо определенной предметной области по одному объекту из этой предметной области. Предлагается формальными методами строить концептуальный «каркас» предметной области, который затем применяется в экспертной процедуре построения онтологии этой области.

1. Концептуальный «каркас» онтологии предметной области

Среди многих определений онтологии выделим следующее: онтология предметной области – это кортеж:

$$O^d = \langle C, A, R, D \rangle, \quad (1)$$

где C – множество классов предметной области, A – множество атрибутов классов, R – отношение частичного порядка на множестве классов, $R \subseteq C \times C$, D – множество доменов (экземпляры класса).

При таком определении отношения R в онтологии определен класс отношений «Isa» (отношение Класс-Подкласс). Считается, что два класса $c_i, c_j \in C$ в онтологии O^d находятся в отношении - «Isa» если между атрибутами $a_i, a_j \in A$ и доменами $d_i, d_j \in D$ класса определены следующие зависимости $a_i \subset a_j$ & $d_i \supset d_j$. В этом случае класс c_i называется надклассом, а класс c_j – подклассом.

В онтологическом моделировании используют термины: класс (надкласс), атрибуты класса, экземпляр класса. Далее мы будем использовать синонимичную терминологию: понятие, обобщенное понятие – это класс, надкласс; содержание понятия – это атрибуты класса; объем понятия – это множество экземпляров класса.

В дескриптивной логике (ДЛ) [Baader, 2004], используемой в настоящее время для формального описания онтологий, на множестве экземпляров предметной области выделяется наиболее специфический объект – *msc* (*most specific concept*), имеющий более подробное описание свойств (большее число признаков), записанных на языке ДЛ, по сравнению с остальными объектами.

Допустим, что эксперт определил объект v^{msc} в некоторой предметной области, и будем считать, этот объект наиболее специфический объект этой предметной области.

Формально определим понятие (класс) этого специфического объекта тройкой: $\langle d_i, F(d_i), V(d_i) \rangle$, где d_i – имя понятия, $F(d_i) = \{f_{ij}\}$ – содержание понятия (множество признаков), $V(d_i) = \{v^{msc}\}$ – объем понятия, где v^{msc} – наиболее специфический объект, имеющие признаки $F(d_i)$.

Пусть $B(F(d_i)) = \{\emptyset, 2^{F(d_i)}\}$ – булеан содержания $F(d_i)$ (множества признаков) понятия d_i , где $2^{F(d_i)}$, множество всех подмножеств содержания $F(d_i)$. Известно, что элементы булеана образует частично упорядоченное множество по включению его элементов, т.е. решетку $(B(F(d_i)), \wedge, \vee)$.

Как видим, элементы булеана образуют частично упорядоченное множество, так же как и классы онтологии в определении (1).

Алгебраическую решетку, образованную элементами булеана можно считать прообразом онтологии предметной области, если сделать два

следующих допущения:

1. Любой элемент $F(d_i^H) \in B(F(d_i))$, $H \in [1, 2^{|F(d_i)}|]$, полученной решетки формально будем считать содержанием понятия d_i^H обобщающего понятие d_i , если $F(d_i^H) \subseteq F(d_i)$.

Это допущение означает, что в решетке $(B(F(d_i)) \wedge, \vee)$ любое подмножество $F(d_i^H) \in 2^{F(d_i)}$ может быть интерпретировано как множество атрибутов i -го класса онтологии A_i , т.е. $p(F(d_i^H)=A_i)$, где i -номер класса в онтологии, $p \in \{0, 1\}$ - степень истинности того, что подмножество d_i^H содержания специфического понятия d_i в формальной алгебраической решетке $B(F(d_i))$ является множеством атрибутов класса онтологии предметной области, т.е. $F(d_i^H)=A_i$.

2. Отношение включения содержаний элементов решетки $(F(d_i^H) \subseteq F(d_i))$ будем считать отношением «Класс-Подкласс» (*Isa*), при условии, что для объемов этих понятий выполняется достаточное условие $V(d_i) \subseteq V(d_i^H)$.

Это допущение означает, что если определены (например, экспертным способом) элементы объема (экземпляры класса) удовлетворяющие условию $V(d_i) \subseteq V(d_i^H)$, то степень истинности p из первого допущения принимает значение «истина», т.е. $p=1$.

Определение 1. Решетку $K(d_i)=(B(F(d_i)) \wedge, \vee)$ всех подмножеств содержания начального понятия d_i будем называть **концептуальным «каркасом»** онтологии плохо определенной предметной области.

В концептуальном «каркасе» онтологии определены: множество признаков понятия $F(d_i^H)$ (атрибутов классов A); отношения частичного порядка R на множестве всех подмножеств содержания понятия $F(d_i)$, но не определены; абстрактные имена понятий d_i^H (классов C) и их объемы $V(d_i^H)$ – экземпляры класса.

Концептуальный «каркас» считается необходимой структурой онтологии предметной области, к которой принадлежит понятие d_i . Это означает, что атрибуты $A=\{a_j\}$ классов C онтологии предметной области O^d , построенной экспертом будут принадлежать концептуальному «каркасу» $K(d_i)$ объекта d_i этой предметной области, т.е. $a_j \in K(d_i)$, $\forall j$.

Определение 2. Обобщенное понятие d_i^H в концептуальном «каркасе» онтологии $K(d_i)$ будем называть реальным, если разности объемов обобщенного и необобщенного понятия непустое множество, $V(d_i^H) \setminus V(d_i) \neq \emptyset$, иначе это понятие будем называть виртуальным.

Определение 3. Онтологией предметной области O^d называется подмножество элементов концептуального «каркаса» $O^d \subseteq K(d_i)$, в котором все

содержания обобщенных понятий $F(d_i^H)$ реальны – $V(d_i^H) \setminus V(d_i) \neq \emptyset$, $\forall i, H$, и определены их имена d_i^H .

Пример концептуального «каркаса» и экспертно построенной онтологии предметной области геометрических фигур приводится в работе [Кулинич, 2006]. Здесь определен наиболее специфический объект – прямоугольный треугольник как: плоская геометрическая фигура; ограничена тремя сторонами; один угол прямой. Его содержание определено двоичным вектором, включающим три единицы (1,1,1). Все собственные подмножества множества признаков этого понятия также обозначаем двоичным вектором, в котором отсутствующие признаки обозначаются нулем. Например, двоичный вектор (1,1,0) определяет содержание понятия с признаками: плоская геометрическая фигура и ограничена тремя сторонами – это треугольник.

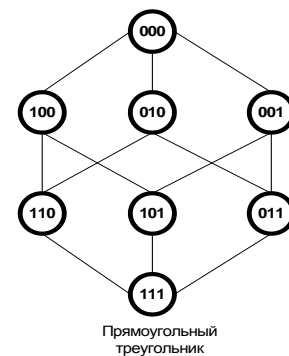


Рисунок 1 – Концептуальный «каркас» геометрических фигур

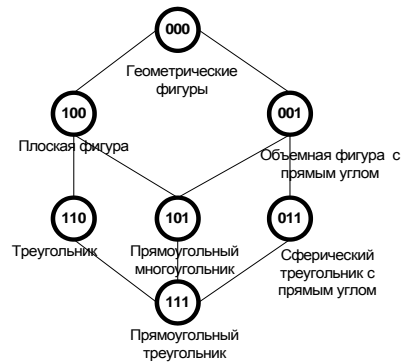


Рисунок 2 – Онтология геометрических фигур

Концептуальный «каркас» прямоугольного треугольника приведен на рисунке 1. Онтология геометрических фигур, построенная на основе этого концептуального «каркаса» с использованием экспертных процедур приведена на рисунке 2.

Общее число обобщенных понятий, для которых эксперт должен определить имя и объем этого понятия определяются по формуле (при условии, что число значений признаков одинаково):

$$N = n^k,$$

где, N – число обобщенных понятий в концептуальной решетке; n – число значений каждого признака; k – число признаков исходного понятия.

2. Качественные концептуальные «каркасы» в семантических пространствах

Одной из моделей представления экспертных знаний для анализа их организации является модель субъективного семантического пространства, которое представляется как система категорий индивидуального сознания, при помощи которой происходит оценка и классификация объектов, понятий [Психологический словарь, 1996]. Размерность семантического пространства определяется числом признаков понятия. Каждая ось семантического пространства соответствует одному из признаков понятия, а сами понятия представляются в виде точек. Положение точки, характеризующее понятие, определяется значениями его признаков. Для оценки и классификации понятий в семантическом пространстве принимается гипотеза о том, что это пространство является метрическим, что подтверждают эксперименты [Shepard, 1966]. В плохо определенной предметной области субъект наблюдает один объект, который представляется в семантическом пространстве, размерность которого определяется числом признаков понятия, характеризующего наблюдаемый объект.

Пусть есть объект ν^0 – имеет множество признаков $F=\{f_i\}$. Пусть известны множества возможных значений каждого признака наблюдаемого объекта – $Z_i=\{z_{ik}\}$.

Определение 4. Множество $Z_i=\{z_{ik}\}$ значений признака f_i будем называть качественным доменом этого признака, считая, что все элементы этого множества строго упорядочены, т.е. $z_{ik} > z_{ik+1}, \forall k$.

Определение 5. Семантическим пространством $SS(\nu^0)$ объекта ν^0 , будем называть пространство, определяемое прямым произведением качественных доменов всех его признаков, т.е. $SS(\nu^0)=\times_{i,j} Z_{ij}$.

Определение 6. Объект ν_i^0 в семантическом пространстве $SS(\nu^0)$ определяется вектором значений всех его признаков $\nu_i^0=(z_{ik}, \dots, z_{im}), \nu_i^0 \in SS(\nu^0)$.

Похожие объекты в семантических пространствах расположены на небольшом расстоянии (считаем, что в семантическом пространстве задана метрика) и принадлежат одному классу (понятию).

Рассмотрим окрестность точки семантического пространства, представляющей объект $\nu_i^0=(z_{ik} \pm \varepsilon_{ik}, \dots, z_{im} \pm \varepsilon_{im})$, где $\pm \varepsilon_{ik} \in Z_i$.

Определение 7. Окрестность $z_{ik} \pm \varepsilon_{ik}$ значений i -го признака объекта $\nu_i^0=(z_{ik}, \dots, z_{im})$, $z_{ik} \in \pm \varepsilon_{ik}$, в котором не меняется имя класса, к которому этот объект принадлежит, будем называть интервалом толерантности класса по этому признаку $\Delta_i=[z_{ik} + \varepsilon_{ik}, z_{ik} - \varepsilon_{ik}]$.

Определение 8. Подпространство семантического пространства $SS(d^0) \subseteq SS(\nu^0)$, полученное прямым произведением интервалов толерантности признаков объекта ν^0 будем называть базовым понятием или классом объекта ν^0 :

$$SS(d^0)=\times_i [z_{ik} + \varepsilon_{ik}, z_{ik} - \varepsilon_{ik}] = \times_i \Delta_i, SS(d^0) \subseteq SS(\nu^0).$$

Для дальнейших рассуждений сделаем допущение, что любая точка семантического пространства является экземпляром понятия. Т.е. $(z_{ik}, \dots, z_{im}) \in Z_i, \forall i, k, m, z_{ik}, \exists \nu_i \in SS(\nu^0)$.

Это допущение позволит нам в дальнейших рассуждениях опускать утверждение: «Если существует экземпляр». Т.е. мы считаем, что реальный объект определен в любой точке семантического пространства.

Утверждение 1. Подпространство $SS(d^H) \subset SS(\nu^0)$, определяющее понятие d^H является обобщением базового понятия d^0 , определенное подпространством $SS(d^0) \subset SS(\nu^0)$, в том и только в том случае если $SS(d^0) \subset SS(d^H)$.

Пусть задано базовое понятие d^0 , для множества признаков, которого определены интервалы толерантности каждого признака, $\Delta_i=[z_{ik} - \varepsilon_{ik}, z_{ik} + \varepsilon_{ik}]$ и подпространство $SS(d^0)$.

Утверждение 2. Качественным положительным обобщением понятия d^0 по признаку q является понятие, имеющее области толерантности для признака q : $\Delta_q^h=[z_{qk} - \varepsilon_{qk}, \sup Z_q]$ и, соответственно, определенное подпространством: $SS(d^H)=\times_{i(i \neq q)} \Delta_i \times [z_{qk} - \varepsilon_{qk}, \sup Z_q]$

Утверждение 3. Качественным отрицательным обобщением понятия d^0 по признаку q является понятие, имеющее области толерантности для признака q : $\Delta_q^h=[\inf Z_q, z_{qk} + \varepsilon_{qk}]$ и, соответственно, определенное подпространством: $SS(d^H)=\times_{i(i \neq q)} \Delta_i \times [\inf Z_q, z_{qk} + \varepsilon_{qk}]$

Утверждение 4. Качественным обобщением понятия d^0 по k -признакам является понятие, имеющее интервалы толерантности по необобщенным признакам равными интервалам толерантности базового понятия, а интервалы толерантности по обобщенным признакам соответственно равны $\Delta_q^h=[\inf Z_q, z_{qk} + \varepsilon_{qk}]$ или $[z_{qk} - \varepsilon_{qk}, \sup Z_q]$.

Отличительной особенностью качественного обобщения является, то, что при любом числе возможных значений признаков в семантическом пространстве рассматриваются только три следующих интервала значений:

- базовый интервал – $[z_{ik} - \varepsilon_{ik}, z_{ik} + \varepsilon_{ik}]$;
- больше базового интервала – $[z_{ik} - \varepsilon_{ik}, \sup Z_i]$ – положительное обобщение;
- меньше базового интервала – $[\inf Z_i, z_{ik} + \varepsilon_{ik}]$ – отрицательное обобщение.

Сложность построения качественного концептуального «каркаса» равна – $N=n^3$, где n – число признаков.

Для всех элементов множества подпространств $\{SS(d_i^H)\}$, $\forall i, H$, определены следующие свойства:

1. Рефлексивности: $\forall a, SS(d_a^H) \subset SS(d_a^H)$;
2. Антисимметричности: $\forall i, q, SS(d_i^H) \subset SS(d_q^H) \wedge SS(d_q^H) \subset SS(d_i^H) \Rightarrow S(d_i^H) = SS(d_q^H)$;
3. Транзитивности: $\forall a, b, c, SS(d_a^H) \subset SS(d_b^H) \wedge SS(d_b^H) \subset SS(d_c^H) \Rightarrow SS(d_a^H) \subset SS(d_c^H)$.

Наличие этих свойств означает, что на множестве подпространств $\{SS(d_i^H)\}$ определен частичный порядок всех подмножеств.

Утверждение 5. Для любой пары подпространств из множества $\{SS(d_i^H)\}$ определены верхняя и нижняя границы.

Если для пары подпространств $\{SS(d_i^H)\}$ определены верхняя и нижняя границы и они удовлетворяют свойствам 1-3, то на множестве подпространств определена решетка, которая является концептуальным «каркасом» предметной области.

Определение 9. Качественным концептуальным «каркасом» в семантическом пространстве будем называть решетку подпространств $(\{SS(d_i^H)\} \wedge, \vee)$ этого семантического пространства.

Пример структуризации семантического пространства для объекта с двумя признаками в виде качественного концептуального «каркаса» показан на рисунках 3-6.

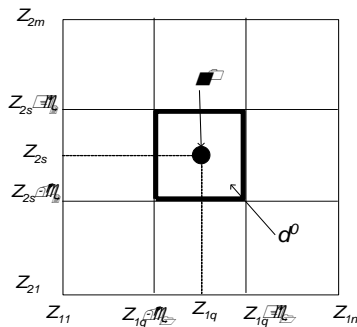


Рисунок 3 – Базовое понятие в семантическом пространстве

Базовое понятие (Рисунок 3) определено как подпространство:

$$SS(d^0) = [Z_{1q} + \varepsilon_1, Z_{1q} - \varepsilon_1] \times [Z_{2s} + \varepsilon_2, Z_{2s} - \varepsilon_2].$$

Графическое представление всех обобщений (d^1, d^2, d^3, d^4) базового понятия по одному признаку представлено на Рисунке 4. Подпространства обобщенных понятий определяются из следующих соотношений:

$$\begin{aligned} SS(d^1) &= [Z_{1q} - \varepsilon_1, Z_{1n}] \times [Z_{2s} + \varepsilon_2, Z_{2s} - \varepsilon_2]; \\ SS(d^2) &= [Z_{1q} + \varepsilon_1, Z_{1q} - \varepsilon_1] \times [Z_{2s} - \varepsilon_2, Z_{2m}]; \\ SS(d^3) &= [Z_{1q} + \varepsilon_1, Z_{11}] \times [Z_{2s} + \varepsilon_2, Z_{2s} - \varepsilon_2]; \\ SS(d^4) &= [Z_{1q} + \varepsilon_1, Z_{1q} - \varepsilon_1] \times [Z_{2s} + \varepsilon_2, Z_{21}]. \end{aligned}$$

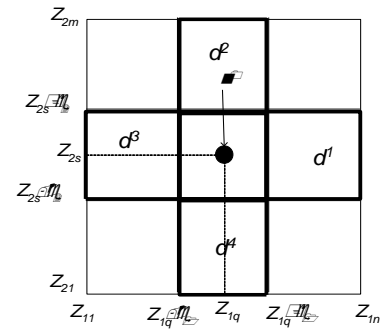


Рисунок 4 – Обобщения базового понятия

На рисунке 5 показаны обобщенные понятия (d^5, d^6, d^7, d^8) обобщающие базовое понятие по двух признакам.

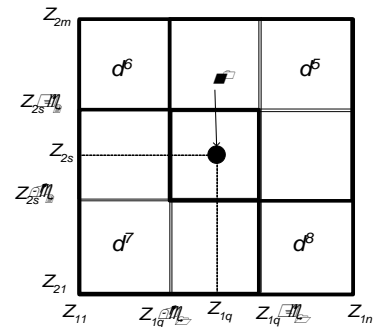


Рисунок 5 – Обобщения понятий (d^1, d^2, d^3, d^4) .

Подпространства обобщенных по двум признакам понятий определяются их соотношений:

$$\begin{aligned} SS(d^5) &= [Z_{1q} - \varepsilon_1, Z_{1n}] \times [Z_{2s} - \varepsilon_2, Z_{2m}]; \\ SS(d^6) &= [Z_{1q} + \varepsilon_1, Z_{11}] \times [Z_{2s} - \varepsilon_2, Z_{2m}]; \\ SS(d^7) &= [Z_{1q} + \varepsilon_1, Z_{11}] \times [Z_{2s} + \varepsilon_2, Z_{21}]; \\ SS(d^8) &= [Z_{1q} - \varepsilon_1, Z_{1n}] \times [Z_{2s} + \varepsilon_2, Z_{21}]. \end{aligned}$$

Качественный концептуальный «каркас» для абстрактного объекта с двумя признаками показан на рисунке 6.

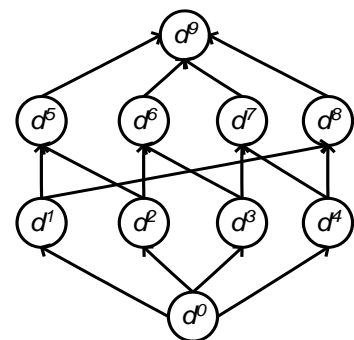


Рисунок 6 – Качественный концептуальный «каркас» в семантическом пространстве

3. Зависимости признаков и структура концептуального «каркаса»

Несколько снизить трудоемкость построения концептуального «каркаса» можно, если учитывать закономерности строения наиболее специфического объекта предметной области. Эти закономерности

выражаются функциональными зависимостями между признаками объекта.

В этом случае понятие определяем четверкой:

$$\langle d_i, F(d_i), \varphi(F(d_i)), V(d_i) \rangle,$$

где $\varphi(F(d_i))$ – зависимость значений признаков понятия d_i .

Задание функциональной зависимости на множестве значений признаков означает, что множество всех подмножеств $B(\varphi(F(d_i)))$ будет подмножеством булеана $B(F(d_i))$, т.е. $B(\varphi(F(d_i))) \subseteq B(F(d_i))$.

Утверждение 6. Все подмножества содержания понятия с зависимостями его признаков $B(\varphi(F(d_i)))$ образуют частично упорядоченное множество по включению его элементов, т.е. концептуальный «каркас» онтологии.

В качестве примера рассмотрим концептуальный «каркас» прямоугольного треугольника (Рисунок 1). Однако, второе свойство – геометрическая фигура «ограничена тремя сторонами» заменим другим определением – «ограничена тремя прямыми». Такое определение позволяет сформулировать зависимость, выраженную правилом: $\varphi(F(d_i)) = (\text{Если Геометрическая фигура ограничена тремя прямыми, То она Плоская})$. Это правило порождает концептуальный «каркас», показанный на рисунке 7 (отсутствуют узлы $F(d_3) = (011)$ и $F(d_5) = (010)$).



Рисунок 7 – Онтология геометрических фигур с учетом зависимостей признаков

Рассмотрим семантическое пространство $SS(\nu^0)$ некоторого объекта ν^0 . Пусть определено его базовое понятие ($SS(d^0)$) – класс. Пусть определен концептуальный «каркас» в виде решетки подпространств семантического пространства.

Пусть между значениями признаков разных объектов существует функциональная зависимость (определены отношения).

Сделаем ряд допущений. Считаем, что если определены зависимости между значениями признаков объекта, принадлежащему классу, то эти зависимости справедливы для любого объекта этого класса. Сделаем еще более сильное допущение:

зависимость между признаками некоторого объекта считается законом, действующим в предметной области, к которой этот объект принадлежит.

Утверждение 7. Если для признаков объектов некоторого семантического пространства на множестве значений его признаков определены отношения $R \subseteq SS(\nu^0)$, то концептуальный «каркас» онтологии предметной с заданной функциональной зависимостью признаков будет включать только те классы обобщенных понятий концептуального «каркаса», подпространству которых принадлежат элементы этого отношения. Т.е. $K(d^0) = \{SS(d_i^H) \mid SS(d_i^H) \cap R \neq \emptyset\}$.

В качестве примера рассмотрим многозначный формальный контекст представленный в таблице 1. Здесь перечислены пять сортов яблок, для каждого из которых определены значения двух признаков: *Цвет* из множества значений {Зеленый, Желтый, Красный} и *Вкус* из множества значений {Кислый, Кисло-сладкий, Сладкий}.

Таблица 1 – Многозначный формальный контекст

№	Сорт/ Признак	Цвет	Вкус
1	Антоновка	Зеленый	Кислый
2	Фуджи	Красный	Сладкий
3	Белый налив	Желтый	Кисло-сладкий
4	Апорт	Красный	Кисло-сладкий
5	Ренет	Желтый	Кислый

Для выявления зависимости между значениями признаков воспользуемся аппаратом анализа формальных понятий [Ganter, 1999], позволяющим по формальному контексту определить концептуальную решетку и выявить закономерности предметной области в виде правил, отражающих зависимости значений признаков перечисленных объектов. Были получены следующие правила:

1. Если **Вкус** Кислый, То **Цвет** Зеленый или Желтый.
2. Если **Вкус** Кисло-сладкий, То **Цвет** Желтый или Красный.
3. Если **Цвет** Желтый, То **Вкус** Кислый или Кисло-сладкий.
4. Если **Цвет** Красный, То **Вкус** Сладкий или Кисло-сладкий.

Правила, полученные на основе анализа, могут быть сформулированы экспертом, знакомым с этой предметной областью. Ниже (см. рисунки 8-11) рассмотрен пример построения качественного концептуального «каркаса» для желтого яблока, имеющего кисло-сладкий вкус на основе экспертных представлений о выше названных закономерностях предметной области (правил 1-4).

На рисунке 8 определено базовое понятие – «Яблоко» с признаками: Цвет – Желтый; Вкус – Кисло-сладкий.

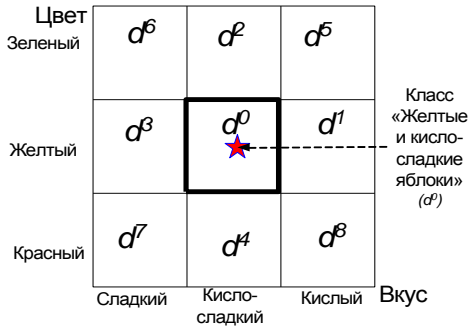


Рисунок 8 – Базовое понятие – Желтое, кисло-сладкое яблоко.

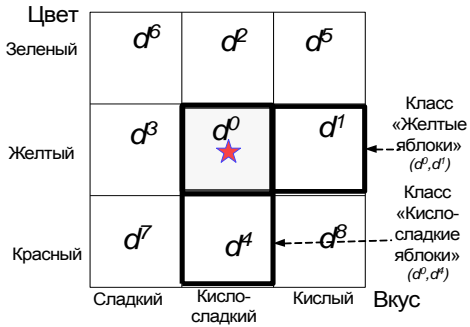


Рисунок 9 – Обобщения базового понятия по одному признаку

На рисунке 8 показано базовое понятие «Желтое, кисло-сладкое яблоко». На рисунке 9 показаны два обобщения первого уровня. По признаку Вкус обобщение осуществляется по правилу 3, т.е. **Если** цвет яблока желтый, **То** вкус - кисло-сладкий или кислый. Это - класс «Желтые яблоки». Обобщение по признаку Цвет строится по правилу 2, т.е. **Если** Кисло-сладкие яблоки, **То** цвет яблок - желтый или красный. Получили класс «Кисло-сладкие яблоки»

На рисунке 10 показаны два обобщения второго уровня - это классы: Красные яблоки имеют вкус кисло-сладкий или сладкий (используется правило 4) и Кислые яблоки имеют цвет желтый или зеленый (используется правило 1). На рисунке 11 показан концептуальный качественный «каркас» этой предметной области.

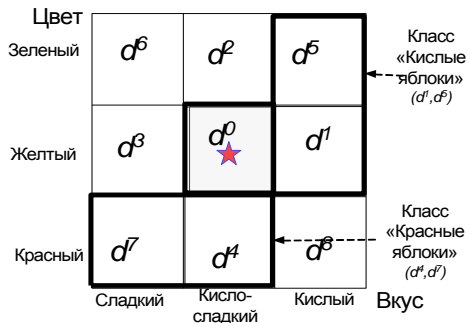


Рисунок 10 – Обобщения базового понятия по двум признакам.

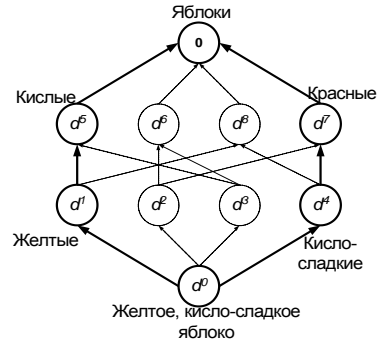


Рисунок 11 – Качественный концептуальный «каркас»

Зависимости между признаками объекта предметной области могут быть заданы графиками функций. В качестве примера рассмотрены два графика (Рисунки 12 и 14). В этом случае концептуальные «каркасы» будут включать только те классы обобщенных понятий, которые пересекает график этих функций (Рисунки 13 и 15).

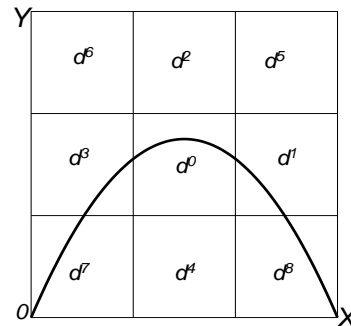


Рисунок 12 – Зависимость признаков $y = -kx^2 + c$.

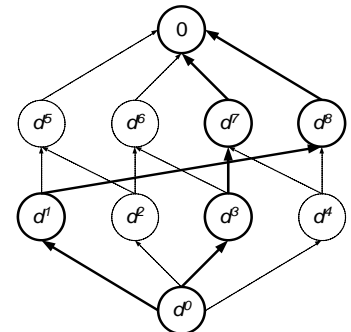


Рисунок 13 – Концептуальный «каркас» онтологии для зависимости признаков $y = -kx^2 + c$.

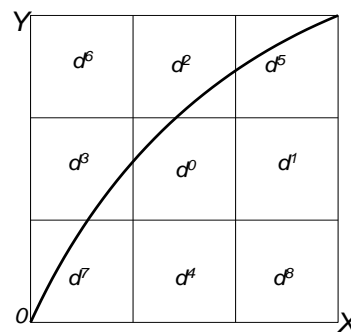


Рисунок 14 – Зависимость признаков $y = x^k (k < 1)$.

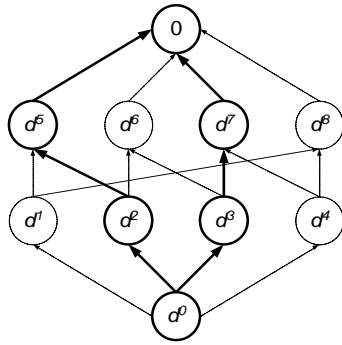


Рисунок 15 – Концептуальный «каркас» онтологии для зависимости признаков $y=x^k$ ($k<1$).

Интерес представляет построение концептуальных «каркасов» процессов динамических систем. Под процессом будем понимать траекторию изменения состояния системы в фазовом пространстве. На рисунке 16 показано изменение состояния маятника. На рисунке 17 показан качественный концептуальный «каркас» этого процесса.

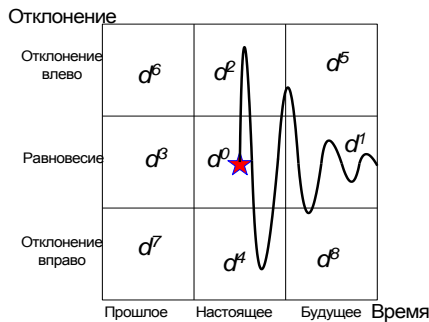


Рисунок 16 – График процесса «Затухающие колебания»

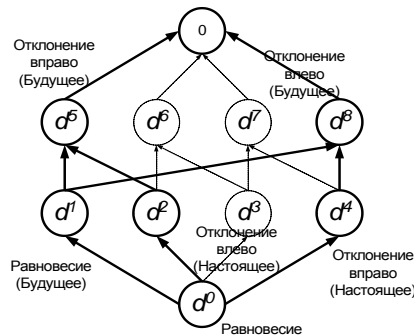


Рисунок 17 – Концептуальный «каркас» процесса «Затухающие колебания»

ЗАКЛЮЧЕНИЕ

Предложенная модель представления знаний и метод построения онтологий на основе концептуальных «каркасов» направлены на активизацию интеллектуальной деятельности и креативности эксперта на этапах построения онтологии плохо определенной предметной области.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Baader, 2000] F. Baader and R. Molitor. Building and Structuring Description Logic Knowledge Bases Using Least Common Subsumers and Concept Analysis. In B. Ganter and G. W. Mineau, eds., Proceedings of the 8th International Conference on

Conceptual Structures (ICCS 2000), volume 1867 of Lecture Notes in Computer Science, pp. 292-305. Springer-Verlag, 2000.

[Baader, 2004] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. The Description Logic Handbook: Theory, Implementation, and Applications.

[Ganter, 1999] Ganter, B. and Wille, R.: Formal Concept Analysis, Mathematical Foundations, Springer, 1999.

[Shepard, 1966] Shepard R.N. Metric structures in ordinal data// J. of Math. Psy-chol. – 1966 -V.3. - №.2.

[Кулинич, 2006] Кулинич А.А. Моделирование динамических процессов в понятийной системе субъекта для генерации креативных решений. Когнитивные исследования: Сборник научных трудов: Вып. 1/ Под редакцией В.Д. Соловьева. – 2006. с. 94-123.

[Мозжерина, 2011] Мозжерина Е. С. Автоматическое построение онтологии по коллекции текстовых документов. Труды 13й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2011, Воронеж, Россия, 2011.

[Найханова, 2008] Найханова Л. В. Методы и модели автоматического построения онтологий на основе генетического и автоматного программирования: Автореф. дис. докт. тех. наук. — Красноярск, 2008. — 36 с.

[Психологический словарь, 1996] Психологический словарь. - М.: Педагогика-Пресс, 1996.

[Рабчевский, 2009] Рабчевский Е.А. Автоматическое построение онтологий на основе лексико-синтаксических шаблонов для информационного поиска. Труды 11-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2009, Петрозаводск, Россия, 2009. стр. 69-77.

[Чечкин, 1991] Чечкин А.В. Математическая информатика. – М.: Наука. Гл. ред. Физ-мат. лит., 1991. – 416 с. – ISBN 5-02-014136-4.

CONCEPTUAL "TEMPLATES" OF THE UNCERTAINTY SUBJECT DOMAINS

A. A. Kulinich

V.A. Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Russia, Moscow.

kulinich@ipu.ru

The model of knowledge representation, and ontology construction method of the uncertain subject domain on the basis of conceptual "templates" is offered.

INTRODUCTION

Existing methods of the automated and automatic construction of subject domain ontology are considered. Their inefficiency at construction ontology in uncertain subject domains is shown.

MAIN PART

Questions of conceptual "template" construction of a subject domain using one object from this area are investigated. Dependence of conceptual "template" structure from dependences of characteristic values of object on which conceptual "template" has been constructed is shown.

CONCLUSION

The offered model of knowledge representations and ontology construction method on the basis of conceptual "templates" are directed on stimulation of intellectual activity and expert creativeness at stages uncertain subject domain ontology construction.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 519.711.74

О ПОНЯТИИ ФОРМАЛЬНОЙ КОМПЕТЕНТНОСТИ НАУЧНЫХ СОТРУДНИКОВ

Крюков К.В., Кузнецов О.П., Суховеров В.С.

*Институт проблем управления им. В.А. Трапезникова РАН,
г. Москва, Россия*

kryukovkirill@yandex.ru

olkuznes@ipu.ru

suhoverov@ipu.ru

Предлагается метод алгоритмического установления компетентности научных сотрудников путем анализа содержания их публикаций. Метод основан на онтологии области научного знания. На основе этого метода осуществляется поиск экспертов, рецензентов, оппонентов, а также выбор команды для выполнения наукоемкого проекта.

Ключевые слова: компетентность, релевантность, онтология, терминология, рецензирование

ВВЕДЕНИЕ

Проблема определения компетентности стара как мир. Она, в частности, возникает всякий раз, когда требуется либо определить, пригоден ли данный человек для выполнения данной работы, либо выбрать из данной группы лиц человека, наиболее пригодного для выполнения данной работы. Для ее решения имеется множество приемов и процедур: заполнение анкет, тестирование, резюме, испытательный срок и т.д. Другой аспект этой проблемы – определение уровня владения определенными знаниями – всегда был существенной частью любого образовательного процесса. В последние десятилетия в рамках работ по управлению персоналом значительное внимание уделяется стандартизации и формализации процедур определения компетентности и попыткам разрабатывать информационные технологии, реализующие эти процедуры.

Очевидно, что для разных предметных областей методы определения компетентности имеют свою специфику. В настоящей работе рассматриваются задачи формализации некоторых процедур определения компетентности, типичных для научных организаций: алгоритмическое установление компетентности сотрудников на основе содержания их публикаций (компетентность, установленную таким методом, будем называть формальной); поиск экспертов, рецензентов, оппонентов, а также выбор команды для

выполнения наукоемкого проекта на основе формальной компетентности.

1. КОМПЕТЕНТНОСТЬ И РЕЛЕВАНТНОСТЬ

В данной работе мы будем исходить из стандартного определения компетенции как комбинации знаний и умений, позволяющей эффективно решать некоторый конкретный набор задач. В работе [Draganidis, 2006] компетенция описывается как следующая совокупность:

- 1) Наименование компетенции;
- 2) Категория, или область, к которой принадлежит компетенция;
- 3) Описание компетенции, объясняющее, что может делать обладатель компетенции;
- 4) Свидетельства компетенции, по наличию которых можно определить, обладает ли сотрудник данной компетенцией.

В дальнейшем мы будем различать компетенцию и компетентность. Компетенция – это понятие, в общем случае не связанное с конкретным лицом; компетентность – это отношение человек-компетенция, означающее, что человек владеет данной компетенцией. Соответственно, к понятию компетенции относятся пп. 1-3; п.4 должен подтверждать наличие отношения компетентности.

Существует множество работ по автоматическому определению компетентностей на основе документов. Упомянем в качестве примеров [Macdonald, 2006], [Ruger, 2009], [Ranwez, 2010]. Все они используют текстовый поиск на предмет

нахождения в документе свидетельств компетенции, которыми являются слова и словосочетания. В большинстве работ описанием компетенции служит запрос пользователя. Свидетельством обладания компетенцией является упоминание слов запроса в текстах документов, автором которых пользователь являлся. При этом могут использоваться методы расширения запросов, например, включение синонимов. Фактически этот подход представляет собой определение релевантности документа запросу.

Конкретизируем поставленную выше задачу определения научных компетенций в терминах приведенного выше определения компетенции.

Наименование компетенции совпадает с ее категорией – это некоторая область научного знания.

Под описанием компетенции мы будем понимать следующее. Формально описание компетенции представляется фрагментом онтологии, структурирующей научную область (скажем, теорию игр или науки об управлении) по отношению тема-подтема. Это формальное описание интерпретируется как владение тематикой, описанной выделенным фрагментом, достаточное для того, чтобы быть экспертом в данной области – рецензентом статей по этой тематике, оппонентом соответствующих диссертаций и т.д.

Источниками свидетельств компетентности будем считать публикации в рецензируемых журналах. Требование рецензируемости существенно, поскольку предлагаемый подход не ставит целью оценить качество публикаций (корректность, новизну, актуальность, стиль и т.д.). Поэтому предполагается, что качество уже положительно оценено рецензентами работы.

Возникает вопрос: что считать свидетельствами искомой компетенции в рассматриваемом документе? Выше отмечалось, что в большинстве предлагаемых методов таким свидетельством является достаточное число совпадений слов и словосочетаний в документе и в запросе. Однако для научных компетенций это не слишком удобно и к тому же не всегда дает нужный результат. Во-первых, в науке много компетенций, свидетельства которых не совпадают с названием. Например, в статье по тематике «Теория алгоритмов» слово «алгоритм» может ни разу не встретиться. Во-вторых, в названиях научных областей часто используются «неспециализированные» слова, которые можно встретить в статьях по любой тематике: тот же «алгоритм», «игра», «множество» и т.д. Научная тематика скорее характеризуется набором терминов, специфических для данной области. С другой стороны, вряд ли следует набор терминов всякий раз включать в текст запроса. Его тогда надо знать заранее – иначе разные пользователи, имея в виду одну и ту же тематику, будут включать в запрос разные термины. Поэтому целесообразно «объективировать» этот набор, т.е.

включить его в описание предметной области.

Предлагаемый подход заключается в следующем. Предметная область научного знания представляется в виде онтологии с двумя типами вершин: вершины-темы и вершины-термины. Вершины-темы образуют основной каркас дерева и связаны между собой отношением тема-подтема, имеющим все таксономические свойства отношений типа класс-подкласс. Вершина-термин связана ровно с одной темой отношением тема-термин, но при этом предполагается, что все нижележащие вершины наследуют этот термин. Все термины являются висячими вершинами. На рисунок 1 приведен малый фрагмент онтологии теории игр, где термины изображены пунктирными прямоугольниками, а отношение тема-термин – пунктирными стрелками.

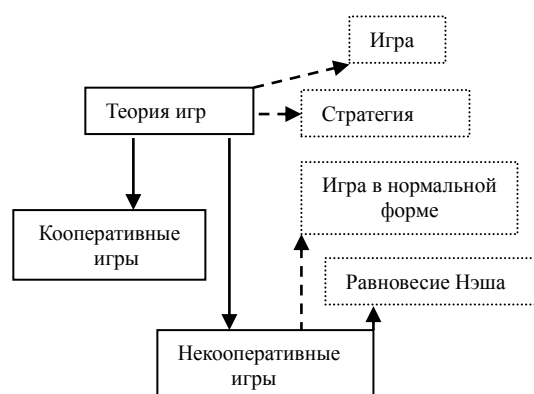


Рисунок 1 - Фрагмент онтологии теории игр

Релевантность документа теме характеризуется использованием основных терминов этой темы; компетентность сотрудника в теме – наличием публикаций, релевантных теме. И релевантность, и компетентность будем оценивать в непрерывной 5-балльной шкале. Интерпретация основных точек этой шкалы для релевантности – следующая.

0 – отсутствие релевантности.

1 – минимальная релевантность (в тексте есть хотя бы одно свидетельство темы);

2 – в тексте кратко упоминается тема (есть некоторое количество свидетельств темы, позволяющее оценить их расположение и разнообразие).

3 – в тексте часто упоминается тема, часть текста посвящена теме;

4 – в тексте достаточно подробно раскрыта тема, нераскрытыми остались лишь некоторые её части;

5 – тема рассмотрена максимально широко и разносторонне.

Интерпретация шкалы компетентности:

0 – отсутствие компетентности,

1 – минимальная компетентность,

2 – у автора есть некоторое представление о теме,

3 – автор работал с темой,

4 – автор хорошо знает тему и обладает достаточно широким кругозором в ней,

5- максимальная компетентность.

2. ФОРМАЛЬНОЕ ОПИСАНИЕ МОДЕЛИ

Опишем предлагаемую модель формально.

Онтология предметной области строится так, как описано выше. Отметим нестандартность дерева онтологии: в нем висячие вершины (вершины-термины) имеются на всех уровнях иерархии. При этом отношение тема-термин не является таксономическим (типа часть-целое или класс-подкласс). Более естественно термины рассматривать не как экземпляры, а как атрибуты соответствующих тем.

Пусть C – множество всех тем онтологии; $c = |C|$; T – множество всех терминов онтологии.

Релевантность документа d темам онтологии определяется следующим образом.

Производится автоматическое аннотирование документа, т.е. поиск в нем всех терминов из T . В результате получается аннотация документа $V^d = \{v_1^d, \dots, v_{|T|^d}\}$, где v_j^d – число вхождений j -го термина в документе d .

Для определения уровня релевантности теме используются три параметра, которые вычисляются на основе найденных в документе терминов:

- 1) общее количество упоминаний основных терминов темы в документе;
- 2) объем темы – число фрагментов документа (например, абзацев или частей страницы), в которых встречались основные термины темы;
- 3) разнообразие основных терминов темы (число различных терминов) в документе.

Исходя из этих соображений, релевантность R_i^d документа d i -й теме вычисляется по следующей формуле

$$R_i^d = \alpha \cdot \frac{\sum_{n=1}^{|J_i|} v_n^d}{P_U} + \beta \cdot \frac{O_i^d}{P_O} + \gamma \cdot \frac{|J_i^d|}{|J_i|} \quad (1)$$

где α, β, γ – настроечные коэффициенты, $\alpha + \beta + \gamma = 5$; J_i – множество всех терминов, относящихся к теме i ; O_i^d – объем темы в документе; J_i^d – множество всех различных терминов темы i , которые встретились в документе; P_U, P_O – нормирующие (так, чтобы соответствующие дроби не превосходили 1) величины для упоминаний и для объема, соответственно. Например, можно положить P_O равным O^d – объему всего документа. Объем темы измеряется с помощью понятия окна. Окно – это либо абзац, либо фиксированное число строк. Тогда O_i^d – число окон, в которых встречается хотя бы один термин темы; O^d – общее число окон.

3. Профилем $PD(d)$ документа d называется его

представление в виде вектора $PD(d) = (R_1^d, \dots, R_c^d)$.

Формула релевантности для группы G документов обобщает формулу (1) и имеет следующий вид:

$$R_i(G) = \alpha \cdot \frac{\sum_{l=1}^{|G|} \sum_{n=1}^{|J_i|} v_n^{d_l}}{P_U} + \beta \cdot \frac{\sum_{l=1}^{|G|} O_i^l}{P_O} + \gamma \cdot \frac{|J_i^G|}{|J_i|} \quad (2)$$

где l – номер отдельного документа из коллекции документов; J_i^G – множество всех различных терминов темы i , которые встретились в G . Нетрудно видеть, что, если G состоит из одного документа, то формула (2) переходит в формулу (1).

Будем считать, что уровень формальной компетентности автора в теме равен уровню релевантности всех его работ этой теме. Тогда профиль $PA(a)$ компетентностей автора a – это вектор релевантностей статей автора всем темам предметной области. Он имеет вид:

$$PA(a) = (R_1^{Da}, \dots, R_i^{Da}, \dots),$$

где Da – все документы автора.

Следует иметь в виду, что предлагаемая оценка компетентности – это оценка снизу, т.е. «формальная компетентность», так как автор может иметь публикации не по всем темам, в которых он реально компетентен.

3. ПРИЛОЖЕНИЯ

Используя профили документов и сотрудников, можно решать задачи выбора экспертов (рецензентов, оппонентов) для оценки документа, консультантов по той или иной теме, а также подбора команды для работы над наукоемким проектом. Например, задача выбора рецензента решается в два этапа: сначала профили документа и кандидатов в рецензенты «очищаются» – в них обнуляются релевантности, которые ниже выбранного порога; затем профили кандидатов ранжируются по степени семантической близости к профилю документа с помощью одной из известных метрик (см., например, обзор [Крюков]).

В настоящее время проводится работа по созданию автоматизированной системы подбора рецензентов для журналов в области теории управления. Разработана онтология наук об управлении и словарь терминов с привязкой к терминам онтологии. Науки об управлении характеризуются большим разнообразием прикладных теорий, математических методов и областей приложения. Поэтому верхний уровень онтологии выглядит следующим образом (рисунок 2):

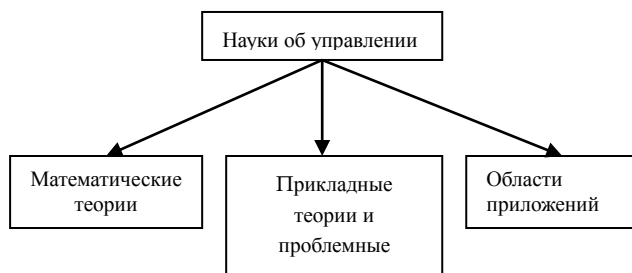


Рисунок 2 – Верхний уровень онтологии

На следующем уровне иерархии под «Математическими теориями» располагаются вершины «Алгебра», «Дифференциальные уравнения», «Математическая логика», «Теория вероятностей» и др., под «Прикладными теориями» – вершины «Теория автоматического управления», «Автоматизация проектирования», «Анализ данных», «Искусственный интеллект», «Передача и обработка сигналов», «Теория организационных систем», под «Областями приложений» - вершины «Вычислительные и коммуникационные системы и сети», «Социально-экономические системы», «Бизнес и финансы», «Технологические процессы», «Робототехника», и др.

Словарь терминов получен путем анализа статей журнала «Автоматика и телемеханика» за последние два года; в дальнейшем он будет пополняться. При формировании словаря важно было выбирать термины, характерные для данной темы; поэтому такие часто встречающиеся, но «безликие» термины, как «функция», «система», «уравнение», в словарь не вошли. Ясно, что такой отбор терминов можно было провести только экспертным путем.

ЗАКЛЮЧЕНИЕ

В работе предложен метод автоматического определения компетентности научных сотрудников путем анализа содержания их публикаций. Метод основан на специальной онтологии области основного знания, ядром которой является иерархия тем научной области, связанных таксономическим отношением тема-подтема. Каждой теме соответствует набор терминов, ее характеризующих. Релевантность документа (публикации) теме вычисляется по пятибалльной шкале, исходя из трех параметров: общее количество упоминаний основных терминов темы в документе; число фрагментов документа (например, абзацев или частей страницы), в которых встречались основные термины темы; разнообразие основных терминов темы (число различных терминов) в документе. Профиль документа определяется как вектор его релевантностей всем темам онтологии. Профиль компетентности сотрудника определяется как профиль группы его публикаций. Показано, как, используя профили документов и сотрудников, осуществлять выбор рецензентов статьи, консультантов по теме и подбор команды для работы над наукоемким проектом.

Работа выполнена при поддержке РФФИ (грант № 11-01-00771).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Draganidis, 2006] Draganidis F., Mentzas G. Competency based management: a review of systems and approaches/ Information Management & Computer Security, 2006, Vol. 14, No. 1, pp. 51-64.
- [Macdonald, 2006] Macdonald C., Ounis, I. Voting for candidates: adapting data fusion techniques for an expert search task/ CIKM 2006: Proc. of the 15th ACM International Conference on Information and Knowledge Management. ACM, New York (2006), pp. 387–396.
- [Ranwez, 2010] Ranwez Sylvie et al. User Centered and Ontology Based Information Retrieval System for Life Science/ Available from Nature Precedings, 2010. <http://precedings.nature.com/documents/5408/version/1/files/npre20105408-1.pdf> (28/03/2011).
- [Ruger, 2009] Ruger S., Zhu J., Song D. Integrating multiple windows and document features for expert finding/ Journal of the American Society for Information Science and Technology, 2009, 60 (4), pp. 694-715.
- [Крюков, 2010] Крюков К.В., Панкова Л.А., Пронина В.А., Суховеров В.С., Шипилина Л.Б. Меры семантической близости в онтологиях// Проблемы управления, 2010, т. 5, с. 2-14.

A CONCEPT OF FORMAL COMPETENCE FOR RESEARCH WORKERS

Kryukov K.V., Kuznetsov O.P., Suhoverov V.S.

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

kryukovkirill@yandex.ru

olkuznes@ipu.ru

suhoverov@ipu.ru

The paper describes the method of competence calculation for research workers by content analysis of their publications. The method uses the special ontology of science domain. The application of this method for a selection of reviewers, opponents and a team for high-tech project is shown.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ИССЛЕДОВАНИЕ РОДОВИДОВЫХ ОТНОШЕНИЙ В ТЕРМИНОЛОГИЧЕСКИХ СЕТЯХ

Мальковский М.Г., Соловьев С.Ю.

Факультет ВМК МГУ имени М.В.Ломоносова, г.Москва, Россия

malk@cs.msu.su

soloviev@glossary.ru

В работе исследуются свойства терминологических сетей как методологии конструирования понятийных структур. Выявлен набор структурно-топологических свойств диаграмм решеток формальных понятий, позволяющий также характеризовать системы родовидовых связей терминологических сетей. Обосновывается выбор фрагмента терминологического пространства в качестве объекта исследования. Приводятся и обсуждаются результаты вычислительного эксперимента, формулируются выводы о возможности привлечения методов формального анализа понятий для построения и верификации терминологических сетей.

Ключевые слова: понятия; терминологические сети; анализ формальных понятий; решетки.

ВВЕДЕНИЕ

Анализ формальных понятий [Ganter, 1999], растущие пирамидальные сети [Гладун, 2004], терминологические сети [Мальковский и др., 2012] – вот далеко не полный перечень подходов к формированию понятийных структур. Несмотря на различия в постановках задач, в методах их решения, в получаемых результатах и в областях применения, вопрос о сравнении родственных подходов является вполне закономерным. В настоящей работе излагается взгляд на терминологические сети как на частично упорядоченные множества – результат анализа формальных понятий.

1. Терминологические сети

Терминологические сети представляют собой подкласс семантических сетей для определений терминов из одной или нескольких проблемных областей [Мальковский и др., 2012]. Построением терминологической сети занимается научный редактор, в помощь которому предоставлен разнообразный программный инструментарий. Деятельность редактора по построению терминологической сети является творческой, но весьма регламентированной, что позволяет (с определенными оговорками) говорить об объективном характере создаваемой им сети.

Вершинами терминологической сети являются

словарные статьи, каждая из которых определяет некоторый термин, его синонимы, а также содержит иную “сопутствующую” информацию. Предполагается, что каждому термину сети соответствует некоторое понятие проблемной области. В данном случае понятие понимается вполне традиционно – как совокупность объектов (объем понятия), обладающих общими свойствами (содержание понятия), отличающих объекты понятия от прочих объектов.

Для терминологических сетей характерно использование ограниченного количества бинарных отношений, связывающих вершины. Так, в проекте “Универсальное терминологическое пространство” (УТП) между определениями 53'477 терминов установлены связи двух типов:

- отношение “это-есть” (34'566 экземпляров);
- отношение “относится-к” (41'003 экземпляра), включающее в себя все прочие типы бинарных отношений.

УТП изменяется во времени за счет вовлечения новой терминологии и корректировки ранее включенных терминов, их определений и связей. В дальнейшем нас будут интересовать связи только первого типа, посредством которых задаются родовидовые отношения между понятиями.

При построении УТП соотношения между понятиями-проблемной-области устанавливает редактор, присваивая некоторым вершинам УТП понятийный статус. С формальной точки зрения понятийная вершина отличается от обыкновенной

тем, что обладает специально сконструированным уникальным именем и может служить входящей вершиной для ориентированных дуг бинарных отношений. Существует довольно обширный арсенал приемов, позволяющих редактору принять решение об учреждении новой понятийной вершины.

Часто информация о семантическом окружении [Гринева-Гринева, 2009] понятия в явном виде содержится в определении термина, причем для родовидовых отношений в практике составления толковых словарей закрепились устойчивые шаблоны описаний. Так из следующего определения:

Таможенная пошлина – налог, взимаемый государством с провозимых через национальную границу товаров по ставкам, предусмотренным таможенным тарифом. По объекту обложения различают ввозимые, вывозимые и транзитные таможенные пошлины. По методу исчисления различают адвалорные, специфические и комбинированные таможенные пошлины.

немедленно вытекают

(А) факт существования понятия “Таможенные пошлины”, наименование которого построено из определяемого термина переходом к множественному числу и объем которого составляют всевозможные “налоги, взимаемые государством с провозимых через национальную границу товаров по ставкам, предусмотренным таможенным тарифом”;

(Б) наличие родовидовой связи между понятиями “Таможенные пошлины” и “Налоги”; и

(В) существование шести подвидов понятия “Таможенные пошлины”, порожденных двумя классификациями на элементах объема:

V.1.1 “Ввозимые таможенные пошлины”,

V.1.2 “Вывозимые таможенные пошлины”,

V.1.3 “Транзитивные таможенные пошлины”,

V.2.1 “Адвалорные таможенные пошлины”,

V.2.1 “Специфические таможенные пошлины” и

V.2.3 “Комбинированные таможенные пошлины”.

В приведенном анализе конкретного определения существенно используются, во-первых, классификационный характер [Гринева-Гринева, 2009] терминологической системы таможенного дела, а, во-вторых, неявно используется гипотеза о возможности выявления объективно существующих понятийных отношений из текстовых определений [Шелов, 2003].

При построении УТП реальные трудности возникают при работе с полиморфными [Шелов, 2003] определениями, допускающими неоднозначные толкования. В этом случае основным приемом структурирования выступает сопоставление определений, позволяющее путем логических выводов и поискам компромиссов выявить/установить связи между понятиями. По результатам сопоставления в УТП возникают общие понятийные вершины – суть – вершины, связанные родовидовыми отношениями с двумя или более

понятийными вершинами более высокого уровня общности. Примеры общих понятий “Музыкальные комедии” и “Географические атласы” представлены на рисунках 5 и 6. Важно отметить, что наличие в УТП общих понятий выводит родовидовую структуру понятийных отношений из класса древовидных иерархий.

При работе с большим количеством понятий естественным образом возникает необходимость их объединения в тематические кластеры, каждый из которых соответствует некоторой проблемной области или отрасли науки. В существующей версии УТП представлены 183 тематических кластера, содержащие от 10 до 100 понятий. С точки зрения техники реализации каждый кластер представляет собой вершину УТП, с которой связаны понятийные вершины; для связи используется особый подвид отношений “относится-к”. Отметим, что вершины терминологической сети, отвечающие кластерам, не входят в состав родовидовой структуры.

2. Анализ формальных понятий

Самый популярный подход к формированию понятийных структур связан с анализом формальных понятий (АФП). Каждое формальное понятие есть пара множеств

Объем // Содержание.

Предполагается, что Объем – это подмножество объектов из известного множества G , а Содержание – подмножество признаков из M , одновременно присущих исключительно объектам Объема. Подмножество K декартова произведения $G \times M$, именуемое контекстом, однозначно порождает множество формальных понятий, на котором рассматривается естественное отношение порядка:

$$(G_1 // M_1) \leq (G_2 // M_2) \Leftrightarrow G_1 \subseteq G_2$$

Установлено [Ganter, 1999], что для заданного контекста K множество формальных понятий образует полную решетку [Биркгоф, 1984], по которой, в свою очередь, однозначно определяется диаграмма Хассе $H(K)$ – см., например, рисунок 1.

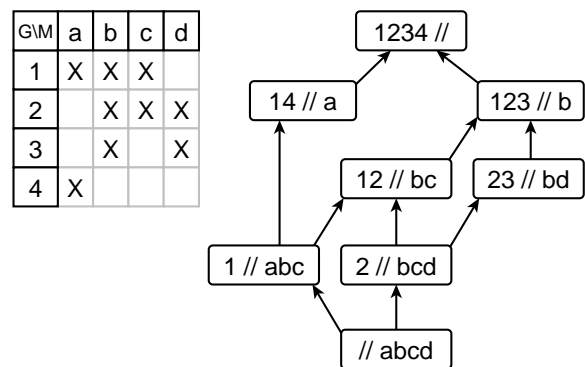


Рисунок 1 – Контекст K_1 и диаграмма Хассе $H(K_1)$

Алгоритмы конструирования решеток по известному контексту известны [Кузнецов, 2004].

На рисунке 1 контекст K_1 задан в виде таблицы, строки которой соответствуют $G = \{ 1, 2, 3, 4 \}$, а столбцы – $M = \{ a, b, c, d \}$; элементы K_1 отмечены в таблице символом X. В общей сложности контекст K_1 позволяет построить восемь формальных понятий, причем два из них – “1234 //” и “// abcd” – фактически от контекста не зависят, они представляют собой “ $G // \emptyset$ ” и “ $\emptyset // M$ ” и играют роли наибольшего элемента I и наименьшего элемента O полной решетки формальных понятий

Простейшие (в некотором смысле) диаграммы полных решеток, именуемые в дальнейшем модельными диаграммами, состоят

- из вершин $O', p_1, \dots, p_n, q_1, \dots, q_m, I'$; и
- из ориентированных ребер
 $(O', p_1), (p_1, p_2), \dots, (p_n, I')$,
 $(O', q_1), (q_1, q_2), \dots, (q_m, I')$.

Конкретный вид модельной диаграммы – рисунок 2(а) и 2(б) – определяется парой чисел $n \geq 1$ и $m \geq 1$, которая записывается в виде формулы $n+m$. Считается, что

- вершине O' отвечает наименьший элемент диаграммы;
- вершине I' отвечает наибольший элемент;
- в каждом ориентированном ребре (a,b) вершина b сопоставлена более широкому понятию, чем вершина a.

Из формальных соображений будем также называть модельными диаграммами двухполюсные сети вида $0+m$ – рисунок 2(в).

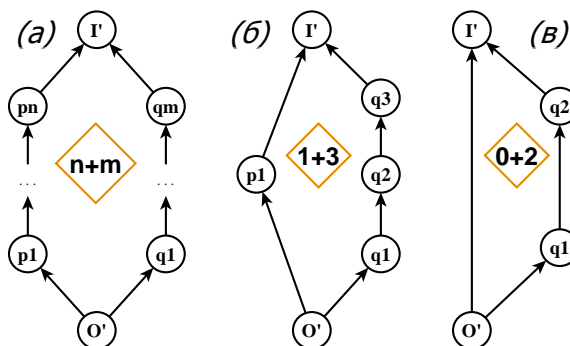


Рисунок 2 – Модельные диаграммы

Сформулируем ряд структурно-топологических характеристик диаграмм Хассе, соответствующих полным решеткам.

Характеристика 1. В диаграмме Хассе полной решетки отсутствуют циклы.

Характеристика 2. В диаграмме Хассе полной решетки обязательно присутствуют наибольший и наименьший элементы I и O.

Характеристика 3. Диаграмма Хассе полной решетки не содержит подсети, изоморфные модельным диаграммам вида $0+m$. Наличие таких подсетей эквивалентно существованию в диаграмме Хассе транзитивных ребер.

Характеристика 4. В диаграммах Хассе полных решеток допускаются специальные подсети,

которые, во-первых, изоморфны модельным диаграммам вида $n+m$, где $n \geq 1$ и $m \geq 1$, и, во-вторых, не содержат элементов I и O. На рисунке 3(а) приводится специальная подсеть вида $1+1$ для диаграммы Хассе $H(K_1)$.

Соответствие двухполюсной сети характеристикам 1-3 позволяет говорить о ее “похожести” на некоторую диаграмму полной решетки. Наличие же в двухполюсной сети специальных подсетей позволяет судить о ее “нетривиальности” как решетки.

3. Классификация понятий

Диаграмма Хассе является двухполюсной сетью без циклов; обратное утверждение неверно. Будем рассматривать внутренние вершины двухполюсных сетей без циклов. Исключая из рассмотрения полюсы, которым зачастую невозможно сопоставить понятия проблемной области, определим три подкласса внутренних вершин. Внутреннюю вершину будем называть

- общей*, если из нее исходят два или более ребер;
- узловой*, если в нее заходят два или более ребер;
- простой*, если в нее заходит ровно одно ребро, исходящее из некоторой другой внутренней вершины.

На рисунке 3(б) приводятся типы вершин для диаграммы Хассе $H(K_1)$.

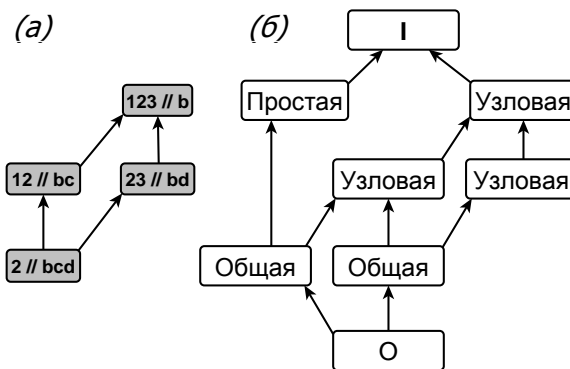


Рисунок 3 – Типы вершин в двухполюсных сетях

Замечание 1. В конкретных диаграммах Хассе могут существовать (а) вершины, не соответствующие ни одному приведенному классу вершин, (б) узловые общие вершины и (в) узловые простые вершины. Например, в диаграмме $H(K_4)$ – рисунок 4 – вершина “3 // bde” не относится ни к одному из трех классов.

Замечание 2. Наличие в диаграммах Хассе общих вершин, позволяет моделировать иерархические но не обязательно древовидные структуры понятий.

Замечание 3. В полных решетках простым вершинам соответствует элементы решеток неразложимые в объединение [Гуров, 2004].

Замечание 4. В развивающихся понятийных структурах простые вершины способны

превращаться в узловые. Так, на рисунке 4 приведен контекст K_4 , который отличается от контекста K_1 наличием только одного нового признака e для объекта 3, однако в диаграмме $H(K_4)$ бывшая простая вершина “23 // bd” стала узловой.

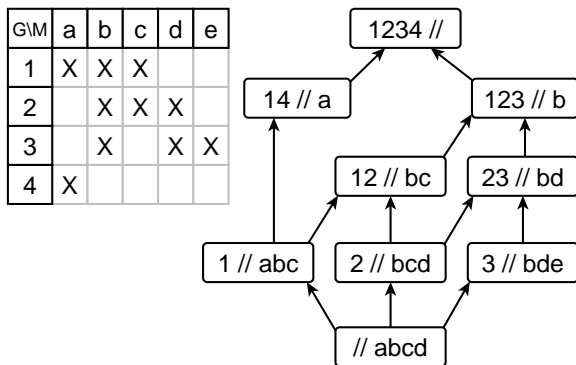


Рисунок 4 – Контекст K_4 и диаграмма Хассе $H(K_4)$

Замечание 5. В растущих пирамидальных сетях [Гладун, 2004], предназначенных для порождения и представления понятийных структур, простые вершины вообще не допускаются.

4. УТП как объект исследования

(УТП \Rightarrow УТП*) Каждая версия УТП содержит некоторое количество частично описанных понятий, присутствующих в терминологической сети, но не раскрытых посредством перечисления подвидов и отношений с другими понятиями. С одной стороны, такого рода “полупонятия” в УТП неизбежны, а с другой стороны, они способны серьезно повлиять на количественные показатели понятийной структуры. В связи с этим для исследования были отобраны только понятийные вершины УТП и родовидовые связи между ними. Усеченная таким образом часть УТП – обозначим ее УТП* – насчитывает 9’043 (из 53’477) вершин-понятий и 7’009 связей между ними.

(УТП* \Rightarrow УТП**) Из соотношения вершин и ребер следует, что УТП* не является связным графом. Как показывают расчеты, в составе УТП* насчитываются 1999 компонент, состоящих из изолированных понятий-вершин, и 261 компонента, каждая из которых состоит из двух вершин. Эти 2260 компонент также следует исключить из анализа по мотивам недостаточности описания. Таким образом, в окончательном варианте графа – обозначим его УТП** – имеется 5’522 вершин и 6’748 связей между ними.

Ориентированный граф УТП** состоит из 316 компонент связности, порождающих разбиение множества вершин УТП** на 316 подмножеств. Подавляющая часть – 4’483 из 5’522 (81%) вершин УТП** входят в одну самую крупную компоненту. Вторая по размеру компонента имеет 86 вершин, третья – 54.

5. Свойства УТП** как решетки

Обработка УТП** позволяет сформулировать ряд согласованных выводов о свойствах родовидовых связей терминологических сетей.

Первое. Как правило, в родовидовой структуре УТП циклы отсутствуют. Оговорка “как правило” здесь и далее означает, что обнаруженные в УТП дефекты представляют собой подлежащие устранению ошибки редактирования.

Второе. В ориентированном графе УТП** имеется 3’783 вершин без заходящих ребер и 617 вершин без исходящих ребер. Разнообразие связанных с этим вершинами понятий чрезвычайно широко: от “CGI-приложений” до “Уроков классического танца” и от “Денежных систем” до “Тушения пожаров”. Отсюда следует, что для УТП невозможно определить термины для наибольшего и наименьшего элементов, то есть элементы I и O могут существовать в УТП** только как абстрактные вершины, не связанные с определенными терминами. Вместе с тем, явно избыточное количество понятий верхнего уровня со всей очевидностью ставит вопрос о терминологическом представлении в УТП “нерасчлененного смыслового континуума” [Морковкин, 1970].

Третье. Как правило, в ориентированном графе УТП** не содержатся модельные диаграммы вида 0+m. Тем не менее отдельные диаграммы вида 0+3 и 0+4 вносятся в УТП вполне сознательно.

Четвертое. Ориентированный граф УТП** содержит 295 модельных диаграмм, из которых:

- 145 диаграмм (49%) вида 1+1;
- 88 диаграмм (30%) вида 1+2;
- 17 диаграмм (6%) вида 2+2;
- 15 диаграмм (5%) вида 1+3;
- 13 диаграмм (4%) вида 2+3.

В большинстве случаев выявленные диаграммы не выводят за пределы тематических кластеров. На рисунке 5 представлена типичная диаграмма, связывающая четыре понятия из тематического кластера “Театральное искусство”.



Рисунок 5 – Диаграмма для кластера “Театральное искусство”

В некоторых случаях в одну модельную диаграмму попадают понятия из родственных кластеров:

- “Горные породы” –и– “Полезные ископаемые”,
- “Ценные бумаги” –и– “Деньги”,
- “Судовождение” –и– “Суда”.

Фактически выявление в УТП модельных диаграмм

оказывается достаточно продуктивной эвристикой для алгоритма автоматической кластеризации понятий по родовидовым связям.

Незначительно количество модельных диаграмм позволяет выявить в УТП нетривиальные связи между понятиями. Пример такой диаграммы приводится на рисунке 6. Восемь понятий этой диаграммы принадлежат тематическим кластерам “Документы”, “Издания”, “Информация” и “Географические карты”.



Рисунок 6 – Модельная диаграмма 2+4

Модельные диаграммы, обнаруженные в УТП**, имеют общие вершины и ребра, что позволяет рассматривать модельные подграфы – суть – максимальные подграфы УТП**, целиком составленные из двух и более модельных диаграмм. По определению модельные подграфы не имеют общих вершин и ребер.

Пятое. В ориентированном графе УТП** обнаружены 43 модельных подграфа, самый крупный из которых имеет 128 вершин-понятий и 173 ребра, а самый мелкий – 6 вершин и 7 ребер. Семнадцать модельных подграфов (40%) решетками не являются, причем с увеличением размеров модельных подграфов вероятность “выпадения” из класса решеток возрастает, а все модельные подграфы, содержащие более 15 вершин и 20 ребер, гарантированно не являются решетками. При проверке свойств модельных графов допускалось отсутствие в решетках наибольшего и/или наименьшего элементов, предполагалось, что универсальные грани I и O можно достроить. На рисунке 7 представлена структура одного из модельных подграфов, который не является решеткой – у него не отсутствуют $\inf \{7, 11\}$, $\sup \{1, 6\}$ и др.

Шестое. В ориентированном графе УТП** имеется 783 общие понятийные вершины, 374 из которых являются наименьшими элементами модельных диаграмм. Таким образом, в УТП** для 409 общих вершин не нашлось явно сформулированных понятий, способных сыграть роль наибольших элементов соответствующих

модельных диаграмм.

$$(409 / 783) * 100\% = 52\%.$$

Седьмое. В ориентированном графе УТП** выявленные модельные диаграммы покрывают 1319 ребер из 6'748. Остальные 5'429 ребер не входят в диаграммы. Доля “неохваченных” ребер составляет

$$(5429 / 6748) * 100\% = 80\%.$$

Восьмое. В ориентированном графе УТП** выявленные модельные диаграммы покрывают 1'048 вершин из 5'522. Остальные 4'474 вершин не входят в диаграммы. Доля “неохваченных” вершин составляет

$$(4474 / 5522) * 100\% = 81\%.$$

Девятое. В ориентированном графе УТП** имеется 191 простая вершина, что составляет 3.5% от общего количества вершин. Как показывают дополнительные исследования незначительная доля простых вершин характерна для всех версий УТП.

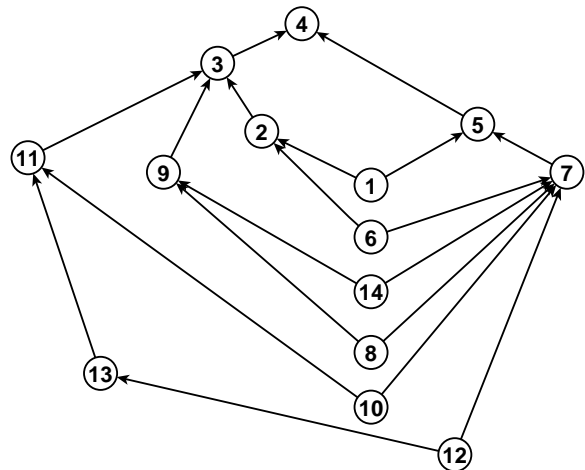


Рисунок 7 – Модельная диаграмма 2+4

В совокупности девять приведенных свойств образуют своеобразную систему косвенных свидетельств о наличии и характере связей между решеточно упорядоченными множествами и терминологическими сетями.

Анализ УТП на наличие модельных диаграмм с последующим построением и исследованием модельных подграфов

- позволяет обнаруживать некоторые дефекты УТП (свойства 1 и 3);
- предлагает подход к автоматической кластеризации понятий (свойство 4);
- открывает возможность data mining [Багсеян и др., 2004] в терминологических сетях (свойство 4).

ЗАКЛЮЧЕНИЕ

По результатам проведенных исследований соотношение между двумя методологиями конструирования родовидовых связей понятийных структур представляется достаточно сложным. С

одной стороны, слишком большое количество реально существующих отношений между понятиями не сводятся к полным решеткам. С другой стороны, полные решетки понятий вполне естественны для хорошо структурированных или отдельно взятых терминологических систем. Соотношения такого рода характерны для взаимодополняющих методик.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Барсегян и др., 2004] Методы и модели анализа данных: OLAP и Data Mining / А.А.Барсегян, М.С.Куприянов, В.В. Степаненко, И.И.Холод – СПб.: БХВ-Петербург, 2004.

[Биркгоф, 1984] Теория решеток / Г. Биркгоф – М.: Наука, 1984.

[Гладун, 2004] Гладун В.П. Растущие пирамидальные сети / В.П.Гладун // Новости искусственного интеллекта. – 2004, № 1. С.30-40.

[Гринев-Гриневиц, 2009] Введение в терминографию: Как просто и легко составить словарь / С.В.Гринев-Гриневиц – М.: ЛИБРИКОМ, 2009.

[Гринев-Гриневиц, 2008] Терминоведение / С.В.Гринев-Гриневиц – М.: Академия, 2008.

[Гуров, 2004] Упорядоченные множества и универсальная алгебра. Вводный курс / С.И.Гуров – М.: ВМК МГУ, 2004.

[Мальковский и др., 2012] Мальковский М.Г., Терминологические сети / М.Г.Мальковский, С.Ю.Соловьев // OSTIS-2012. Материалы конференции. С. 77-82

[Морковкин, 1970] Идеографические словари / В.В. Морковкин – М.: Изд-во МГУ, 1970.

[Кузнецов, 2004] Кузнецов С.О. Методы теории решеток и анализа формальных понятий в машинном обучении / С.О. Кузнецов // Новости искусственного интеллекта. – 2004, № 3. С.19-31.

[Шелов, 2003] Термин. Терминологичность. Терминологические определения / С.Д.Шелов – СПб.: Филологический факультет СПбГУ, 2003.

[Ganter, 1999] Formal Concept Analysis: Mathematical Foundations / V.Ganter, G.Stumme, R.Wille – Berlin: Springer, 1999.

HIERARCHIAL RELATIONS IN TERMINOLOGICAL NETWORK

Malkovsky M.G., Soloviev S.Y.

Lomonosov MSU CS department, Moscow, Russia

malk@cs.msu.su

soloviev@glossary.ru

We are considering the question of the application of formal concept analysis for the design and verification of terminological networks.

INTRODUCTION

In this paper we investigate the properties of the networks as a terminological methodology design conceptual structures. We have identified a set of structural and topological properties of diagrams formal concept lattices, allowing to characterize the hierarchial relations in terminological networks. We have argued and decided on a fragment of the terminological space as an object of study. We present the results of computer simulation to study the fragment. We justify a conclusion about the possibility of bringing the concepts of formal analysis techniques for the construction and verification of terminological networks

MAIN PART

Terminological network is a subclass of semantic networks built for definitions of terms. Terminological network builds editor using software tools. As an example, the network is considered a universal terminology terminological space (UTS). UTS uses two types of binary relations, of which the study involved only one. Technology features of the formation of UTS impose limitations on the structure of the terminological network.

The basic definitions of formal concept analysis are given in the article. We formulate a number of structural and topological characteristics of the Hasse diagram corresponding to complete lattices. We have proposed a method to identify the model diagrams to assess the similarity of arbitrary network complete lattice.

For arbitrary networks we consider classification of vertices used to draw conclusions about the structure of the network and the corresponding partial order. We consider the arguments that allow us to formulate the requirements for the classes of vertices.

We are considering various options for allocating UTS fragment, which is able to provide a valid comparison with the complete lattice.

Computer investigation UTS fragment revealed nine special properties of terminological networks. We found that the formal concepts analysis

- can detect certain defects UTS;
- provides an approach to the automatic clustering of concepts;
- opens the possibility of data mining in the terminological networks.

CONCLUSION

The ratio between the two design methodologies hierarchial relations conceptual structure is complex. On the one hand, too many real-life relationships between concepts can not be reduced to a complete lattice. On the other hand, the full array of concepts is quite natural for a well structured or individual terminology systems. Simultaneous analysis of the UTS for the presence of model diagrams can detect certain defects UTS, as well as an approach to the automatic clustering concepts. Relations of this kind are characteristic of complementary techniques.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ЛИНГВИСТИЧЕСКАЯ ОНТОЛОГИЯ – ТЕЗАУРУС РУТЕЗ

Алексеев А.А., Добров Б.В., Лукашевич Н.В.

*НИВЦ МГУ им. М.В. Ломоносова,
г. Москва, Российская Федерация*

a.a.alekseev@gmail.com

dobrov_bv@mail.ru

louk_nat@mail.ru

В статье рассматривается современное состояние тезауруса русского языка РуТез, сравнивается его структура с другими концептуальными компьютерными ресурсами, рассмотрены типы используемых в различных приложениях путей тезаурусных отношений и окрестностей понятий тезауруса. В качестве примера применения тезауруса РуТез в автоматической обработке текстов рассматривается технология выявления тематической структуры текста путем построения лексических цепочек на основе нескольких факторов близости между языковыми выражениями.

Ключевые слова: лингвистическая онтология; автоматическая обработка текстов; тезаурус русского языка; лексические цепочки

ВВЕДЕНИЕ

Одним из наиболее известных лексических ресурсов в сфере компьютерной лингвистики и автоматической обработки текстов является компьютерный тезаурус WordNet [Miller, 1998], который в последней версии (3.0) включает приблизительно 155 тысяч различных слов и словосочетаний, организованных в 117 тысяч понятий, или совокупностей синонимов (synset).

Появление этого тезауруса в свободном доступе в Интернет вызвало всплеск исследований по его использованию в различных компьютерных приложениях автоматической обработки текстов, породил большое число последователей в разных странах, создающих такие ворднеты для своих языков [Vossen, 1998], а также стал базой для многоплановых дискуссий и исследований того, на основе каких принципов должны строиться большие лингвистические ресурсы, пригодные для разнообразных приложений в области компьютерной лингвистики.

Разработчиков новых ворднетов можно разделить на две категории. Часть разработчиков считает, что важным делом является точное воспроизведение структуры и состава англоязычного WordNet (обычно называемого

Принстонский WordNet по месту работы его авторов), поскольку предполагается, что таким образом обеспечивается более тесная связь с англоязычным ресурсом и лексической системой английского языка. Другие разработчики полагают, что для создания качественного ресурса собственного языка необходимо учесть специфику его лексической системы, а также учесть критику и проблемы Принстонского WordNet. При таком подходе разработчики развивают собственную структуру синсетов, руководствуясь общими принципами построения ворднетов.

На текущий момент в России отмечен большой интерес к ресурсам типа WordNet. Однако начатые проекты разработки русских ворднетов [Азарова, Синопальникова, 2003; Сухоногов, Яблонский, 2005; Гельфенбейн и др., 2003] не достигли объема и/или качества, необходимых для их использования в приложениях.

В данной статье будут описаны структура и современное состояние тезауруса русского языка РуТез, который относится к тому же классу лингвистико-онтологических ресурсов, что и WordNet, но при этом обладает значительными отличиями [Лукашевич, 2011]. В ближайшее время мы планируем начать готовить тезаурус РуТез к публикации и, таким образом, данный ресурс может стать источником семантических знаний в

различных системах автоматической обработки текстов подобно ресурсам типа WordNet.

1. Принципы разработки тезауруса RuTез

Разработка тезауруса RuTез началась в 1998 году на базе Общественно-политического тезауруса [Лукашевич, 2011]. Основным назначением тезауруса является использование его как источника знаний о мире и языке в приложениях информационного поиска и автоматической обработки текстов.

По принципам своей разработки тезаурус RuTез объединяет три существующие традиции разработки компьютерных ресурсов:

- традиции разработки традиционных информационно-поисковых тезаурусов;
- традиции разработки лингвистических ресурсов типа WordNet (Принстонский университет);
- традиции создания формальных онтологий.

Так как предполагается использовать лингвистический ресурс в автоматическом режиме обработки текстов, то необходимо использовать методологию разработки лексических ресурсов типа WordNet, в которой важны следующие положения:

- единицы тезауруса RuTез создаются на основе значений реально существующих языковых выражений;
- многоступенчатое иерархическое построение лексико-терминологической системы понятий;
- принципы описания значений многозначных слов и выражений.

Вместе с тем единица тезауруса RuTез рассматривается не как совокупность синонимов – синсет, а как понятие, имеющее уникальные свойства и уникальное имя в иерархически организованной системе понятий, как это рассматривается в онтологических исследованиях. Кроме того, для описания отношений между понятиями используются отношения с формальными свойствами и аксиомами.

Наконец, практика принятия решений ввода понятий на основе значений многословных выражений близка к практике разработки традиционных информационно-поисковых тезаурусов.

Таким образом, тезаурус RuTез принадлежит к особому классу онтологий, так называемым лингвистическим онтологиям, поскольку введение понятий в значительной мере мотивируется значениями языковых единиц, относящихся к предметной области ресурса. В то же время он является тезаурусом, поскольку каждое понятие связано с набором языковых выражений (слов, терминов, словосочетаний), которыми это понятие может быть выражено в тексте, – такой набор текстовых входов понятий необходим для

использования онтологий для автоматической обработки текстов.

В настоящее время тезаурус RuTез включает в себя 53 тысячи понятий, более 210 тысяч отношений между понятиями, 155 тысяч слов и словосочетаний русского языка

2. Единицы тезауруса RuTез

Единицей описания в тезаурусе RuTез является не множество синонимичных слов или терминов, как в тезаурусе WordNet, а понятие, отражающее значимые классы сущностей, различаемых людьми в мире, в современной общественной жизни, в психической жизни людей. При этом предполагается, что значения слов и выражений, существующие в современных естественных языках, позволяют выделить главное, существенное для современной жизни людей.

При такой ориентации на значения реально существующих выражений естественного языка важным принципом ввода понятий в тезаурус является отличимость каждого вводимого понятия от других понятий в системе понятий. Такая цель иногда трудно достижима, но, на наш взгляд, она должна ставиться, поскольку использование в качестве единиц тезауруса таких отличимых понятий позволяет единым образом представлять лексические значения литературного языка и значения терминов предметной области, более последовательно описывать систему отношений между понятиями и тем самым облегчает формальный вывод на отношениях, позволяет отображать единым образом систему значений разных языков

В тезаурусе RuTез каждое понятие должно иметь однозначное имя, которое построено на базе его текстовых входов, и должно быть понятным носителю языка. Понятие может иметь комментарий, который пишется в случае необходимости и не является частью имени понятия.

Каждое вводимое понятие должно быть снабжено списком слов и словосочетаний, с помощью которых можно сослаться в тексте на вводимое понятие – текстовых входов. В качестве таких текстовых входов могут быть отдельные слова (существительные, прилагательные, глаголы, наречия), а также именные и глагольные группы. Текстовый вход может быть многозначным (иметь другие значения), тогда он должен быть помечен как многозначный. Для лучшего распознавания в тексте текстовые входы тезауруса RuTез снабжаются последовательностью нормализованных форм всех составляющих многослового выражения (мужской род, именительный падеж, единственное число).

Языковые выражения (слова, словосочетания, термины), которые были описаны как текстовые входы одного и того же понятия, становятся

неразличимыми с точки зрения тезауруса РуТез – онтологическими синонимами.

Понятия в тезаурусе РуТез могут иметь достаточно большие ряды онтологических синонимов. Приведем пример синонимического ряда, включающего несколько типов синонимов для понятия *ДУШЕВНОЕ СТРАДАНИЕ* (по алфавиту): *боль* (*м* – многозначное), *боль в душе*, *в душе наболело*, *душа болит*, *душа саднит*, *душевная боль*, *душевная рана*, *душевное страдание*, *душевный недуг*, *рана в душе*, *рана в сердце*, *рана души*, *саднить*, *саднить на душе*, *саднить на сердце*, *сердечная рана*

Как видно, синонимический ряд понятия может содержать значительно количество синтаксических вариантов словосочетаний, некоторые словосочетания образуются заменой слова-компонента на синоним. Установление соответствия таких текстовых входов понятию является наиболее простым способом обнаружения понятия в тексте.

В тезаурусе РуТез существуют два основных способа представления значений многозначных терминов.

Первым способом представления многозначности является задание одного и того же текстового входа разным понятиям тезауруса (М-многозначность). Например, текстовый вход *пилот* сопоставлен двум разным понятиям понятию *ЛЕТЧИК* и понятию *АВТОГОНЩИК*. Такое представление полностью соответствует представлению разных значений слов, принятому в тезаурусах типа WordNet.

Второй способ представления многозначности используется в тех случаях, когда слово представлено в тезаурусе в одном значении, но если известно, что оно может употребляться и в других значениях в целевых текстах, то ему ставится специальная пометка многозначности (А-многозначность).

3. Отношения в тезаурусе РуТез

Отношения между понятиями, описываемые в онтологическом ресурсе, предназначенном для автоматической обработки текстов в рамках информационно-поисковых приложений, должны выполнять разнообразные функции, включая расширение поискового запроса и вывод рубрики; разрешение лексической многозначности; тематического анализа текстов с учетом их лексической связности.

Для реализации любой из этих функций необходимо осуществление своеобразного логического вывода: встретив вхождение некоторого понятия в тексте, нужно делать многошаговые проходы по отношениям.

Для стабильной работы на разных типах текстов в тезаурусе РуТез используется небольшой набор

надежных отношений, т.е. отношений, которые не исчезают, не меняются в течение всего срока существования любого или подавляющего большинства экземпляров понятия.

В результате исследований и экспериментов мы пришли к набору отношений ресурса, предназначенного для эффективной автоматической работы в информационно-поисковых приложениях и приложениях автоматической обработки текстов.

В тезаурусе РуТез имеется четыре основных типа отношений. Первый тип отношений – родовидовое отношение *ниже-выше*, представляет собой отношение класс-подкласс, обладает свойствами транзитивности и наследования.

Второй тип отношений – отношение *часть-целое*. Используется не только для описания физических частей, но и для других внутренних сущностей понятия, таких как свойства или роли для ситуаций. Важным условием при установлении этого отношения является то, что понятия-части должны быть жестко связаны со своим целым, то есть каждый пример понятия-части должен в течение всего времени своего существования являться частью для понятия-целого, и не относиться к чему-либо другому.

В этих условиях удается выполнить свойство транзитивности введенного таким образом отношения *часть-целое*, что очень важно для автоматического вывода в процессе автоматической обработки текстов.

Еще один тип отношения, называемого *несимметричной ассоциацией* $асц2 - асц1$, связывает два понятия, которые не могут быть связаны выше рассмотренными отношениями, но когда одно из которых не существовало бы без существования другого. Например, понятие *КИПЕНИЕ* требует существования понятия *ЖИДКОСТЬ*. В онтологических исследованиях такое отношение называется отношением онтологической зависимости.

Последний тип отношений – *симметричная ассоциация* – связывает, например, понятия, очень близкие по смыслу, но которые разработчики не решились соединить в одно понятие. Таким примером до недавнего времени были понятия *ПОЛИЦИЯ* и *МИЛИЦИЯ*, отношение между ними было естественным образом симметричным. После переименования российской милиции было решено все-таки оставить только одно понятие *ПОЛИЦИЯ*, а слово *милиция* сделать текстовым входом к этому понятию.

Отношения *выше-ниже*, *часть-целое* и *несимметричная ассоциация* являются иерархическими отношениями. Таким образом, на основе свойств иерархичности, транзитивности и наследования для каждого понятия может быть определена совокупность понятий, которые являются для него нижестоящими понятиями по иерархии – так называемое нижняя

полуокрестность понятия (O^-), а также может быть определена совокупность понятий, которые являются для него вышестоящими по иерархии – так называемое верхняя полуокрестность понятия (O^+).

Наиболее часто используются следующие виды путей между понятиями:

- *Путь по иерархии вверх* $P_{up}(C_0, C_{00})$: От понятия C_0 к понятию C_{00} существует путь по иерархии вверх, если $C_{00} \in O^+(C_0)$;

- *Путь по иерархии вниз* $P_{down}(C_0, C_{00})$: От понятия C_0 к понятию C_{00} существует путь по иерархии вниз, если $C_{00} \in O^-(C_0)$;

Введенные типы концептуальных путей используются в процедурах автоматического разрешения лексической неоднозначности, расширения поискового запроса, тематического анализа текстов, вывода рубрик по тексту.

4. Тематический анализ текста на основе тезауруса РуТез

Существуют две основные проблемы применения лингвистических онтологий в автоматической обработке текстов. Во-первых, ни один ресурс не может быть исчерпывающе полон, поэтому для эффективной обработки текстов его применение должно сопровождаться извлечением недостающей информации из текстов.

Во-вторых, обычно требуется применение технологий, которые могут разрешать лексическую неоднозначность слов и выражений, значения которых представлены в лингвистической онтологии. Однако качество разрешения лексической неоднозначности может быть достаточно низким для некоторых типов общепотребительных слов и нестабильным для других типов слов.

В данном разделе мы опишем новую развиваемую нами технологию работы с текстами, которая преодолевает вышеуказанные проблемы и направлена на построение тематического представления текста в виде совокупности так называемых тематических узлов – близких по смыслу слов и выражений, соответствующих отдельному участнику ситуации, описываемой в тексте. Ранее мы строили тематическое представление на основе исключительно информации, описанной в тезаурусе предметной области. Новая технология учитывает в этой процедуре ряд разнородных факторов, только одним из которых является информация из тезауруса.

4.1. Тематическая структура текста и лексические цепочки

Как известно, тематическая структура текста представляет собой иерархическую структуру тем и подтем, причем темы рассматриваются как

предикаты над некоторым множеством участников [Алексеев, Лукашевич, 2011].

Темы и подтемы, находящиеся в иерархических отношениях друг с другом, связываются между собой общими участниками или группами участников, имеющих отношение друг к другу. В тексте эти группы участников, а также варианты их наименования выглядят как так называемые лексические цепочки [Hirst, St-Onge, 1998]. Так, в тексте про вывод американской авиабазы из Киргизии такой цепочкой может быть следующая: *Киргизия, Киргизстан, киргизский, кыргызский, Бишкек* и т.п. Поэтому качественное распознавание таких лексических цепочек в тексте выявляет совокупность взаимодействующих между собой участников описываемой ситуации, и, следовательно, эксплицирует структуру текстового содержания.

Особенностью нашего моделирования лексических цепочек является использование для их построения комбинации нескольких факторов и одного разделяющего фактора, а именно: два языковых выражения не могут быть включены в одну лексическую цепочку, если их частота употребления в одних и тех же предложениях текста более чем в два раза превышает их встречаемость в соседних предложениях (для обоснования см. [Алексеев, Лукашевич, 2011]). Действительно, если мы посредством лексических цепочек пытаемся выявить основных участников описываемой в тексте ситуации, то частая встречаемость отдельных слов (не входящих в устойчивое словосочетание) в предложении свидетельствует в пользу их отнесенности к различным, взаимодействующим между собой участникам.

В качестве единиц, из которых строятся лексические цепочки в нашем случае, выступают: отдельные слова знаменательных частей речи, словосочетания, представленные в тезаурусе РуТез, а также новые словосочетания, извлеченные из текста. Новое словосочетание, полученное из текста, должно иметь частоту в этом тексте не менее 2, и обладать некоторой устойчивостью употребления в данном тексте. Мы проверяем устойчивость словосочетания путем сопоставления совместной встречаемости слов в тексте друг за другом и их отдельной встречаемости в заданных фрагментах предложения.

Отметим, что в предлагаемом методе построения лексических цепочек мы не делаем предварительного разрешения многозначности неоднозначных слов, предполагая, что разные учитываемые нами факторы должны усиливать отношения между теми значениями слов, которые более подходят текущему текстовому контексту.

4.2. Факторы, используемые для построения лексических цепочек

Для построения лексических цепочек используется совокупность факторов, каждый из

которых дает вес близости между языковыми выражениями (словами или словосочетаниями). Эти факторы могут являться контекстно-независимыми и контекстно-зависимыми.

Первый фактор **TS** (Thesaurus Similarity) – это контекстно-независимый фактор, вычисляемый на основе тезаурусных отношений. Максимальный вес связи по тезаурусу (равный 1.0) придается паре выражений, который являются однозначными и указаны в тезаурусе как синонимы. Вес связи по тезаурусу снижается обратно пропорционально длине пути между понятиями, текстовыми входами к которым являются данные выражения. Многозначность слов учитывается следующим образом: если многозначное слово относится к некоторому понятию *C*, и в тексте не было ни одного однозначного текстового входа этого понятия, то вес тезаурусного отношения умножается на некоторый снижающий коэффициент (сейчас установлен = 0.9). Если оба выражения в оцениваемой паре являются многозначными, то снижающий коэффициент применяется второй раз.

Второй контекстно-независимый фактор **BS** (Beginning Similarity) оценивает формальное сходство выражений по написанию. Максимальный вес сходства составляет 1 для слов с одинаковым началом (5 букв: *киргиз* – *Киргизия*). Для многословных выражений, включающих схожие слова, за каждое различающееся слово в паре выражений берется штраф 0.1.

Используются два фактора сходства, вычисляемые на основе пословных контекстов внутри предложений. Фактор **SPS** (Scalar Product Similarity) оценивает сходство между выражениями на основе скалярного произведения пословных векторов их контекстов в предложении (максимум – 0.5). Фактор **SC** (Strong Context) оценивает количество одинаковых контекстов величиной 4 слова (2 слова влево и вправо), нормированное на максимум величины **SC**, достигнутому в данном тексте.

Важнейшим контекстным фактором является фактор **NSF** (Neighboring Sentence Feature), который представляет собой оценку превышения совместной встречаемости в соседних предложениях по сравнению с встречаемостью внутри одних и тех же предложений (см. [Алексеев, Лукашевич, 2011]).

Вычисляется этот фактор следующим образом:

$$NSF = \text{Min} \left[1, \frac{F_{ext} - F_{int}}{\text{Max} (F_{ext} - F_{int})} \right],$$

где *Fext* – это совместная встречаемость выражений в соседних предложениях текста, *Fint* – внутри одних и тех же предложений текста.

Этот же фактор служит ограничителем соединения выражений в одну лексическую цепочку. Если величина NSF < 0, то выражения не

будут объединены в одну лексическую цепочку при любой суммарной величине сходства между выражениями.

Для сравнения сходства более длинных лексических цепочек служит фактор **EO** (Embedded Objects), который носит булевский характер и указывает, что у цепочек есть общий элемент. Этот фактор позволяет быстрее объединять в единую цепочку фрагменты похожих цепочек, начавшие формироваться отдельно.

Веса всех факторов для каждой пары выражений суммируются, и такие пары могут быть упорядочены по мере снижения веса сходства. Примеры наиболее сходных пар выражений для текста об американской авиабазе в Киргизии показаны в Табл. 1.

Таблица 1. Наиболее сходные по нескольким факторам пары языковых выражений в тексте про американскую авиабазу в Киргизии

Выражение1	Выражение2	Вес сходства
<i>Авиабаза</i>	<i>Авиабаза Манас</i>	2.78
<i>Киргизия</i>	<i>Киргизстан</i>	2.75
<i>Арендная плата</i>	<i>Плата за аренду</i>	2.49
<i>Аль-Каида</i>	<i>Аль-Каеда</i>	2.47
<i>База</i>	<i>Военная база</i>	2.45
<i>Кыргызская республика</i>	<i>Кыргызстан</i>	2.45

4.3. Формирование лексических цепочек

После упорядочения пар выражений начинается итеративный процесс соединения выражений в лексические цепочки. Мы формируем лексические цепочки в форме тематических узлов, т.е. в цепочке выделяется главный элемент (центр), который принадлежит только этой цепочке (тематическому узлу). Обычно главный элемент тематического узла является наиболее частотным среди других элементов этого узла. Остальные элементы тематического узла могут входить в еще один тематический узел.

Процесс начинается с рассмотрения пары выражений с максимальным весом. Эта пара соединяется в один узел так, что более частотное выражение становится центром этого узла, а второе элементом. Центр тематического узла становится представителем элементов узла со всеми их вхождениями и контекстами.

Склеивание в единый тематический узел тематических узлов с несколькими элементами происходит похожим образом. Центр наиболее частотного тематического узла становится центром нового объединенного тематического узла.

Итеративный процесс останавливается при достижении между очередной парой выражений порогового веса сходства, равного 1.1. Эта

пороговая величина имеет прозрачный смысл и отражает высокую степень сходства по тезаурусу (однозначные синонимы) плюс небольшое подтверждение от других факторов. Данный вес сходства может быть достигнут и на основе других факторов, при отсутствии в тезаурусе какой-либо связи между двумя выражениями.

Примерами автоматически построенных лексических цепочек могут служить следующие цепочки:

(Киргизия): Киргизстан, Кыргызстан, Киргизский, столица Киргизии, Киргизская республика, Киргизская столица, киргиз, парламент Киргизии..

(Авиабазы): авиабаза Манас, база, база ВВС, авиационная база, база ВВС, база Манас, закрытие базы, военная база США.

В настоящее время мы тестируем возможности использования такого рода лексических цепочек для автоматического аннотирования отдельных текстов и новостных кластеров.

ЗАКЛЮЧЕНИЕ

В данной статье мы рассмотрели современное состояние тезауруса русского языка RuТез, сравнили его структуру с другими концептуальными компьютерными ресурсами. Мы также рассмотрели типы используемых в различных приложениях путей и окрестностей в тезаурусе.

В настоящее время мы развиваем технологии, которые позволяют применять тезаурус RuТез без применения процедуры разрешения многозначности, совместно с техниками извлечения недостающих знаний из самих текстов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Азарова и др., 2003] Азарова, И.В., Митрофанова, О.А., Синопальникова, А.А. Компьютерный тезаурус русского языка типа WordNet // Компьютерная лингвистика и интеллектуальные технологии. Труды Международной конференции Диалог'2003. М., С. 43-50

[Алексеев, Лукашевич, 2011] Алексеев А.А., Лукашевич Н.В. Автоматическое извлечение сущностей на основе структуры новостного кластера // Искусственный интеллект и принятие решений. 2011. N 4. С. 95-103

[Гельфенбейн и др., 2003] Гельфенбейн И.Г., А.В. Гончарук, В.П. Лехельт, А.А. Липатов, В.В. Шило. Автоматический перевод семантической сети WORDNET на русский язык // Компьютерная лингвистика и интеллектуальные технологии. Труды Международной конференции Диалог'2003. М., 2003.

[Лукашевич, 2011] Лукашевич Н.В. Тезаурусы в задачах информационного поиска. М.: Изд-во Московского университета, 2011.

[Сухоногов, Яблонский, 2005] Сухоногов А.М., Яблонский С.А. Автоматизация построения англо-русского WordNet. // Компьютерная лингвистика и интеллектуальные технологии: Труды Международного семинара Диалог'2005 / Под ред. А.С.Нариньяни. – М.: Наука, 2005.

[Hirst, St-Onge, 1998] Hirst G., St-Onge D. Lexical Chains as representation of context for the detection and correction malapropisms // WordNet: An electronic lexical database and some of its applications /C. Fellbaum, editor. Cambridge, MA: The MIT Press, 1998.

[Miller, 1998] Miller G. 1998. Nouns in WordNet // WordNet – An Electronic Lexical Database / Fellbaum, C (ed). – The MIT Press, pp. 23-47.

[Vossen, 1998] Vossen P. EuroWordNet: A multilingual Database with Lexical Semantic Network. – Dordrecht. 1998.

LINGUISTIC ONTOLOGY – RUTHES THESAURUS

Alekseev A.A., Dobrov B.V., Loukachevitch
N.V.

*НИВЦ МГУ им. М.В. Ломоносова,
г. Москва, Российская Федерация*

a.a.alekseev@gmail.com

dobrov_bv@mail.ru

louk_nat@mail.ru

INTRODUCTION

One of well-known lexical resources in the sphere of computational linguistics and natural language processing is WordNet thesaurus.

In this paper we describe the structure and the current state of thesaurus of Russian language RuThes, which belongs to the same class of conceptual linguistic resources (linguistic ontologies) as WordNet. At the same time RuThes has specific features of knowledge representation motivated by its initial purpose to be a resource for natural language processing in information retrieval applications. In the near future we plan to begin the preparation of RuThes thesaurus to publication and therefore this resource can become a useful instrument of Russian text analysis similar to Wordnet-like resources.

MAIN PART

In principles of its development RuThes thesaurus comprises three existing traditions of computer resources such as traditional information-retrieval thesauri, WordNet-like thesauri and formal ontologies.

A unit of RuThes thesaurus is a concept reflecting significant classes of entities in the world and the internal life of people (not synonym sets as in WordNet). The small set of conceptual relations in the thesaurus describes the most reliable relations of concepts. Formal properties of the relations provide a basis for formal logical inference in natural language processing.

As an example of application of RuThes thesaurus to natural text analysis we consider the thematic structure analysis of texts including lexical chain construction based on several factors.

CONCLUSION

In this paper we present the structure and the current state of Thesaurus of Russian Language RuThes and describe the example of its application to the thematic analysis of texts.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.912

ПРОГРАММНЫЙ ИНСТРУМЕНТАРИЙ РАЗРАБОТКИ ЛИНГВИСТИЧЕСКИХ РЕСУРСОВ

Загорулько М.Ю., Сидорова Е.А.

*Институт систем информатики им. А.П. Еришова Сибирского отделения
Российской академии наук, г. Новосибирск, Россия*

zagorulko_maxim@yahoo.com

lena@iis.nsk.su

В работе рассматривается комплекс программных средств, предназначенных для проведения лингвистических исследований и разработки лингвистических ресурсов: корпусов текстов и предметно-ориентированных словарей. Описаны словарные системы, предназначенные для создания терминологических, лексико-семантических и семантико-синтаксических словарей.

Ключевые слова: лингвистический ресурс; терминологический словарь; корпус текстов; конкорданс.

ВВЕДЕНИЕ

Разработка естественно-языковых интерфейсов для обеспечения пользователей информационных систем более удобными средствами доступа, поиска и представления информации требует решения разнообразных лингвистических задач. Сложность задачи зависит от лингвистических потребностей разработчиков информационной системы. Это может быть задача извлечения терминологии и построения терминологического словаря или задача обработки текста на основе лексикографического (словарного) ресурса или реализация полноценного семантического анализа текста и автоматического извлечения информации. Решение любой из перечисленных задач требует наличия определенных лингвистических ресурсов, создание которых (в случае их отсутствия) является отдельной задачей.

Совершенно необходимым ресурсом для решения практически любой задачи является морфологический словарь одного или нескольких языков. Остальные лингвистические ресурсы, как правило, предметно-ориентированы на ту сферу деятельности, в рамках которой создается информационная система. Можно выделить следующие типы лексикографических ресурсов.

- Терминологические словари, в состав которых входят как однословные, так и многословные термины [Большаков, 2002];

- Тезаурусы, содержащие знания о языке в проекции на конкретную сферу деятельности [Лукашевич, 2011];

- Лексико-семантические и семантико-синтаксические словари, основанные на шаблонных описаниях (например, словари моделей управления [Волкова и др., 1998]) и др.

Неотъемлемой частью процесса создания лингвистических ресурсов и разработки методов решения поставленных лингвистических задач, является проведение предварительного лингвистического исследования тех материалов, с которыми в дальнейшем будет работать система. Лингвистические исследования с помощью современных компьютерных технологий проводятся корпусными методами [Захаров и др., 2011] и их часто называют «объективными» или квантитативными. Результатом такого исследования может стать аннотированный корпус текстов, который в дальнейшем служит для автоматизированного создания лексикографических ресурсов. Лексикографическое направление является на данный момент основным в корпусной лингвистике, это связано с тем, что основным инструментом исследования корпуса, является построение конкордансов (т.е. совокупности контекстов слов), которые изначально, еще в «доцифровую» эпоху были основой для создания словарей [Богданова, 2010].

В работе рассматривается комплекс программных средств, предназначенных для проведения лингвистических исследований и разработки лингвистических ресурсов: корпусов

текстов и предметно-ориентированных словарей. Вопросы применения данных лингвистических ресурсов для автоматического анализа текста и, связанные с этим лингвистические модели, останутся за рамками этого доклада.

1. Лингвистические потребности информационных систем

В процессе разработки естественно-языковых модулей возникают различные лингвистические задачи, решение которых требует применения разнообразных лингвистических ресурсов.

1.1. Лингвистические задачи

Рассматривая общую задачу автоматической обработки текста можно выделить несколько подзадач, решение которых связано с созданием и использованием лингвистических ресурсов.

(1) Как уже было отмечено, проведение *лингвистических исследований текстов* необходимо как для выработки основных принципов последующей разработки лингвистических ресурсов, необходимых для решения поставленных задач, так и конкретного наполнения этих ресурсов.

(2) Решение задачи *извлечения терминологии* служит не только для дальнейшей поддержки словарного анализа текста, но и для предварительного экспертного исследования предметной области (ПО), выявления основных классов понятий и логических взаимосвязей между понятиями, которые соответствуют введенным терминам, создания модели ПО, формирования структуры БД системы.

(3) Одним из главных назначений компьютерных лингвистических ресурсов является поддержка одного из этапов *автоматической обработки текста*, целью которого является извлечение из текстов словарных концептов.

(4) В более общем виде задача обработки текста подразумевает *извлечение информации*. Для решения этой задачи используют дополнительные модели и знания о согласованиях и взаимосвязях языковых единиц и элементов ПО с учетом жанровых особенностей документов.

Для решения указанных лингвистических задач разрабатываются лингвистические ресурсы, а также методы и средства их автоматизированного создания и поддержки.

1.2. Лингвистические ресурсы

В нашей технологической цепочке мы рассматриваем следующие типы лингвистических ресурсов.

(1) Корпус текстов – подборка текстов определенного жанра, тематика которых соответствует заданной ПО. Корпус может содержать лингвистическую разметку, представляющую собой информацию, полученную

автоматически при анализе текстов, либо приписанную экспертом вручную. Основное назначение корпуса – автоматизация создания других лингвистических ресурсов.

(2) Универсальные и предметные словари, содержащие перечень минимальных единиц языка, терминов и устойчивых терминологических словосочетаний, используемых при описании значимой для разработчиков системы информации, а также жанровую лексику, описываемую лексическими шаблонами, для извлечения нестандартно представленной в тексте лексики. В рамках словарей определяется набор специфичных для данного подязыка лингвистических знаний: морфологические классы (определение лексемы для найденной в тексте словоформы), правила формирования многословных терминов (синтаксические шаблоны).

(3) Семантический словарь, формирующий семантические признаки и отношения на лексиконе. Семантический словарь включает целевые тезаурусы (например, справочно-информационный тезаурус, тезаурусы для анализа текста, для поддержки информационного поиска, для перевода и т.п.), а также семантико-синтаксические словари, которые ограничивают синтаксическую сочетаемость и проверяет согласованность грамматических и семантических признаков терминов (вершин синтаксических групп) в соответствии с правилами согласования и управления. Эти знания могут быть заданы с разной степенью подробности в зависимости от требований и возможностей разработчиков системы.

(4) Набор описаний жанровых структур текста в совокупности с логическим представлением текста образуют модели документов, соотнесенных с тем или иным типом или жанром текстовых ресурсов.

(5) Знания о согласовании имеющихся лингвистических знаний с предметными знаниями. С этой целью термины группируются в семантические группы, которые в свою очередь также согласуются с элементами онтологии либо непосредственно, либо в соответствии с определенной схемой (схемой факта).

Для решения поставленных разработчиками задач и создания необходимых для их решения ресурсов нами разработан набор программных средств.

Данные инструменты поддерживают как создание лингвистических ресурсов, так и дальнейшее их использование в задачах автоматической обработки текста.

1.3. Схема взаимодействия модулей программного инструментария

Вся технологическая цепочка объединяет три функциональных компонента, отвечающих за управление корпусом (корпус-менеджер), построение словарей, а также за извлечение

информации (рисунок 1). Каждый из компонентов состоит из отдельных программных модулей, имеющих пользовательский интерфейс и определенный формат обмена данными между другими модулями.

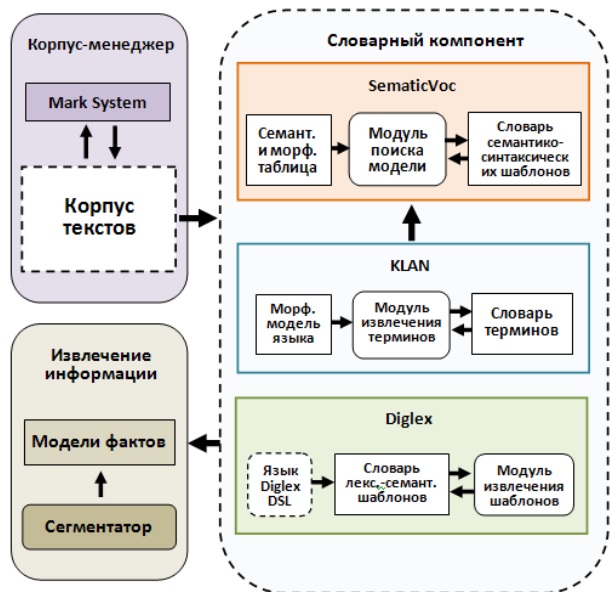


Рисунок 1 – Компонентная архитектура инструментария

Рассматривая весь процесс разработки лингвистических ресурсов и решения поставленных лингвистических задач, можно выделить следующие этапы.

- (1) *Обработка корпуса текстов.* Помимо развитых средств управления коллекцией текстовых документов корпус-менеджер предоставляет возможность многоуровневой разметки текста, на основе которой реализуется автоматизированное наполнение предметных словарей.
- (2) *Построение словарей.* Словарные модули поддерживают разработку словарей, предоставляют необходимые редакторы, реализуют, где это возможно, методы автоматического начального наполнения словарей на основе корпуса текстов, поддерживают дальнейшее использование словарей для обработки текста.
- (3) *Извлечение информации* [Загорюлько и др., 2009] предполагает наличие дополнительных моделей (модели документов, модели фактов, онтологии ПО) и специализированных модулей (в данной работе не рассматриваются).

2. Инструменты создания словарей

Наш инструментарий поддерживает создание нескольких типов словарей.

- Терминологические словари (система KLAN), содержащие предметную и необходимую универсальную лексику.
- Лексико-семантические словари (система DigLex), содержащие как нетерминологические

единицы, имеющие регулярную структуру (например, номер телефона, дата, инициалы) так и специфические термины предметной области, отсутствующие в базовом словаре и/или имеющие сложную структуру (например, полные названия организаций, событий).

- Семантико-синтаксические словари (система SemanticVoc), содержащие модели согласования терминов или классов терминов.

Все словарные системы снабжены пользовательским редактором ресурса, модулем тестирования, осуществляющим текстовый анализ выбранного текста и визуализацию результатов, а также программным API, позволяющим использовать разработанный лингвистом-экспертом ресурс автономно.

2.1. Система KLAN

Система KLAN предназначена для создания предметных словарей (рисунок 2) и позволяет включать в терминологический лексикон грамматическую и статистическую информацию, используемую как для наполнения словаря на основе обучающей выборки, так и для автоматического извлечения словарных терминов из текста.

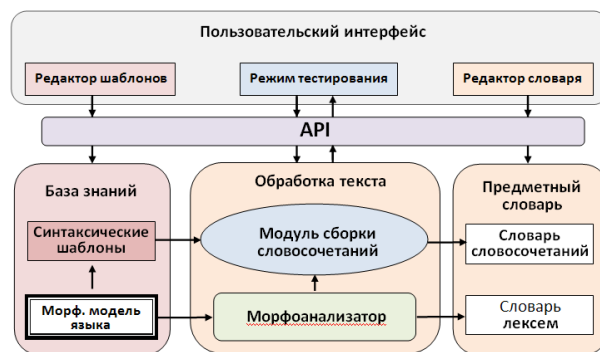


Рисунок 2 – Архитектура системы KLAN

В качестве термина может выступать слово или словосочетание, в общем случае, произвольной длины и состава [Лукашевич и др., 2008]. Для поиска новых словосочетаний используются синтаксические шаблоны, описываемые в терминах морфологических классов и характеристик [Большакова и др., 2010]. Синтаксический шаблон задает иерархическое представление синтаксической структуры словосочетания, где связь между элементами словосочетания осуществляется на основе морфологических характеристик. Система имеет пользовательский интерфейс для редактора шаблонов, позволяющего задать любую синтаксическую конструкцию.

Система поддерживает экспертную настройку морфологической таблицы (набора морфологических классов, атрибутов, типов парадигм), что обеспечивает мультязычность, а также возможность ее согласования с морфоанализаторами сторонних производителей.

2.2. Система Diglex

Система Diglex предназначена для описания и поиска в тексте шаблонных лексических конструкций. Язык описания шаблонов [Жигалов и др., 2002] позволяет определять произвольные символьные выражения, указывать альтернативные выражения, образуемые при использовании сокращений, аббревиатур, синонимов, пропусков и перестановок в лексическом составе конструкции, задавать условия, опциональные подвыражения, дистантный контекст и т.д. Язык Diglex DSL также позволяет описывать структуру слова за счет введения ограничений на его части (окончания, приставки и т.п.) – ограничение длины и задание списка допустимых значений.

Для реализации редактора языка Diglex DSL (рисунок 3) был выбран подход на основе концепции «проекционного редактора». Подход является симбиозом представления языка в виде текста и набора графических примитивов, и имеет ряд преимуществ по сравнению с текстовым представлением.

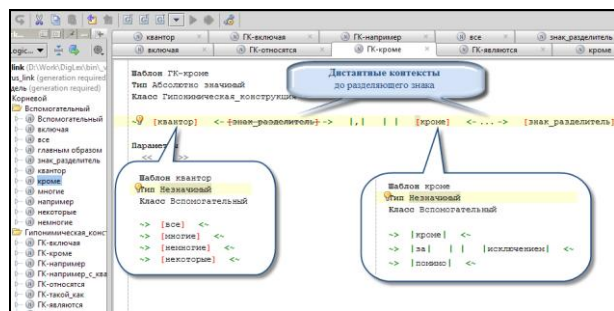


Рисунок 3– Проективный редактор словаря Diglex

Редактор обладает многими возможностями современных сред разработки: «автодополнение», интеграция с системами контроля версий, поиск и подсвечивание ошибок на этапе написания кода, расширенная навигация по коду, различные упрощающие действия и т.п.

Система снабжена модулем тестирования, который позволяет обработать фрагмент текста и визуализировать результат, сгруппированный по шаблонам и классам шаблонов.

2.3. Система SemanticVoc

Система SemanticVoc предназначена для создания семантико-синтаксических словарей, которые ограничивают синтаксическую сочетаемость и проверяют согласованность грамматических и семантических признаков терминов (вершин синтаксических групп) в соответствии с правилами согласования и управления. Словарь включает описание семантико-синтаксических моделей в виде древовидной иерархической структуры (рисунок 4), вершиной которой является лексическая метка (идентификатор модели), на следующем уровне перечисляются актанты, характеризующие соответствующую валентность, а каждый актант описывается набором семантических и грамматических характеристик,

которые являются ограничениями для зависимых слов. Каждая модель может быть приписана любому количеству словоформ, лексем или обобщенных лексем, т.е. лексем, описанных в терминах грамматических и семантических категорий без указания нормальной формы.

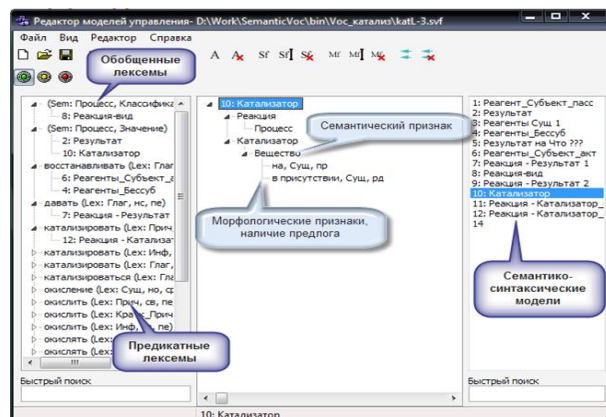


Рисунок 4 – Редактор семантико-синтаксических моделей

Предложенная структура семантико-синтаксических моделей предоставляет широкие возможности моделирования языковых связей в тексте. Так, модель может не содержать синтаксических ограничений и представлять собой онтологические отношения ПО, или модель может описываться без семантических характеристик и соответствовать классическим моделям управления. Обобщение моделей позволяет компактно определить многие языковые конструкции, варианты взаимосвязи слов в предложении и различные словарные группы.

Важным свойством системы, на наш взгляд, является редактируемая и настраиваемая система семантических признаков (в зависимости от ПО) и морфологическая модель языка, что, в частности, обеспечивает условную независимость от языка.

Созданная система является универсальным средством, реализующим словарь семантико-синтаксических шаблонов, и может использоваться в системах, обрабатывающих связный текст, для широкого круга задач. Ядро компонента представляет собой отдельную библиотеку, которая обеспечивает полный набор функций по работе со словарем, а также дополнительные сервисные функции поиска соответствующего актанта, проверки управления или согласования для двух входящих абстрактных терминов. Компонент позволяет создавать независимые xml-словари, а также согласовывать словарь с терминологическим автоматическим словарем, созданным с помощью словарного компонента KLAN.

3. ИНСТРУМЕНТ СОЗДАНИЯ КОРПУСОВ

Инструментарий, поддерживающий создание и изучение текстовых корпусов должен включать средства предварительной разметки текста, поиска данных в корпусе, получения статистической информации и предоставления результатов

пользователю в удобной форме. Данный функционал объединен в специализированной системе для работы с корпусами – корпус-менеджере [Захаров и др., 2011].

На текущий момент создан один из важнейших компонентов корпус-менеджера - модуль разметки текста MarkSystem. Система разметки текстов позволяет приписывать фрагментам текста различные лингвистические признаки. В качестве фрагмента может выступать слово, неразрывная цепочка слов (связный фрагмент) или множество неразрывных цепочек, не образующих связный фрагмент (разрывный фрагмент).

Помимо лингвистической разметки данная система позволяет производить семантическую разметку. Семантическая разметка предметно ориентирована, поскольку определяется онтологией ПО и делится на два типа:

- терминологическая разметка, которая в первую очередь предназначена для фиксации в тексте имен понятий ПО,
- разметка отношений (или ситуаций, представляющих собой многоместные отношения), в которых различные сущности выступают в определенных семантических ролях.

3.1. Конкорданс

Конкорданс – традиционный способ изучения корпуса текста. Создание конкордансов рассматривается как первый этап в работе по составлению словарей. Конкорданс, предоставляя (многочисленные) контексты употребления слов, позволяет выделить основные значения слова по его сочетаемости, месту в синтаксической структуре предложения и т.д. Чем больше параметров фильтрации поддерживает инструмент построения конкорданса (например, по лемме или словосочетанию, по индексу частотности или количеству словоупотреблений, по части речи или синтаксической/семантической структуре), чем разнообразнее анализируемые параметры, тем лучше такой конкорданс может быть использован для систематизации информации и формирования лингвистических ресурсов [Добрынина, 2012].

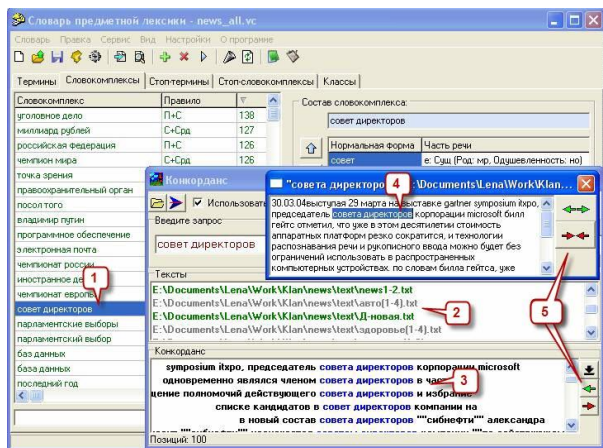


Рисунок 5 – Терминологический конкорданс

На рисунке 5 продемонстрирован пользовательский интерфейс, визуализирующий конкорданс для выбранного многословного термина словаря KLAN.

Для любого однословного или многословного термина (1) можно вызвать модуль построения конкорданса, который осуществит в реальном времени поиск в текстах (2), расположенных в заданной директории. Результатом будут все короткие контексты термина (3), обнаруженные в корпусе. При просмотре контекстов вхождения термина пользователь может самостоятельно определять длину просматриваемых фрагментов (5) (поддерживается пословное расширение контекста) или перейти к просмотру широкого контекста конкретного вхождения термина (4), который также может быть расширен (5).

Терминологический конкорданс дает полный индекс терминов в ближайших и расширенных контекстах. Таким образом, конкорданс осуществляет обратную связь словаря, словарных терминов с корпусом и обеспечивает своего рода лингвистическую разметку на морфологическом и поверхностно-синтаксическом уровне.

На данный момент модуль конкорданса взаимодействует только с системой KLAN. В дальнейшем, планируется внедрить данный модуль в корпус-менеджер для того, чтобы можно было формировать конкорданс по любому основанию, используемому при аннотировании текстов.

ЗАКЛЮЧЕНИЕ

В работе представлен комплексный программный инструмент для разработки лингвистических ресурсов. Каждый инструмент может использоваться независимо от других или в составе любой конфигурации из имеющихся модулей.

Дальнейшие исследования будут направлены на развитие методов и средств обучения и автоматического формирования лингвистических ресурсов (в первую очередь, семантико-синтаксических словарей) на основе обучающего корпуса текстов. Также будут развиваться средства поиска в корпусах, визуализации информации из различных текстов в виде совокупности контекстов фактов, которое будет служить основой для проведения исследований семантических свойств текста.

Работа выполняется при финансовой поддержке РФФИ (проект № 12-07-31216 «Разработка методов создания информационной системы, сочетающей семантическое и текстовое представление информации») и Президиума РАН (Интеграционный проект СО РАН № 15/10 «Математические и методологические аспекты интеллектуальных информационных систем»)

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Богданова, 2010] Богданова, С.Ю. Исследование слова и предложения компьютерными методами: цели и перспективы / С. Ю. Богданова // Слово в предложении: кол. монография / под ред. Л.М. Ковалевой (отв. ред.), С.Ю. Богдановой, Т.И. Семеновской. –Иркутск: ИГЛУ, 2010. – С.194-214.

[Большаков, 2002] Большаков, И.А. Какие словосочетания следует хранить в словарях? / И. А. Большаков // Труды международного семинара Диалог'2002 по компьютерной лингвистике и ее приложениям. Протвино: 2002. Т.2. С.61–69.

[Большакова и др., 2010] Большакова, Е.И. Система для поиска и выделения конструкций в тексте на естественном языке / Е. И. Большакова, А. А. Носков // Труды 12-й национальной конференции по искусственному интеллекту с международным участием – КИИ-2010. – Москва: Физматлит, 2010. Т.3. -С.137-145.

[Волкова и др., 1998] Волкова, И.А. Компьютерный словарь моделей управления русских глаголов (экспериментальный вариант) / И.А. Волкова, И.Г. Головин, О.Ф. Кривнова// Труды Международного семинара по компьютерной лингвистике и ее приложениям "Диалог'98" / под ред. А.С. Нариньяни. –Казань: Хэтер, 1998.

[Гаврилова и др., 2001] Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский // Учебник. СПб.:Питер, 2001.

[Добрынина, 2012] Добрынина, К.С. О методике работы над конкордансами / К.С. Добрынина // Филологические науки. Вопросы теории и практики. –Тамбов: Грамота, 2012. № 1 (12). С. 59-63.

[Жигалов и др., 2002] Жигалов, В.А. Система ALEX как средство для многоцелевой автоматизированной обработки текстов / В. А. Жигалов, Д. В. Жигалов, А. А. Жуков, И. С. Кононенко, Е. Г. Соколова, С. Ю. Толдова // Труды международного семинара Диалог'2002 по компьютерной лингвистике и ее приложениям. Т.2. –М.: Наука, 2002. –С.192–208.

[Загорулко и др., 2009] Загорулко, Ю.А. Технология анализа документов в информационных системах поддержки научной и производственной деятельности / Ю. А. Загорулко, Е. А. Сидорова // Автометрия, 2009. Т.45, №6. –С. 38-45.

[Захаров и др., 2011] Захаров, В.П. Корпусная лингвистика / В.П. Захаров, С.Ю. Богданова // Учебник для студентов гуманитарных вузов. – Иркутск: ИГЛУ, 2011. – 161 с.

[Лукашевич, 2011] Лукашевич, Н.В. Тезаурусы в задачах информационного поиска / Н.В. Лукашевич// –М.: Издательство Московского университета, 2011. –512 с.

[Лукашевич и др., 2008] Лукашевич, Н.В. Отбор словосочетаний для словаря системы автоматической обработки текстов / Н.В. Лукашевич, Б.В. Добров, Д.С. Чуйко // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог 2008». Вып. 7 (14). М.: РГГУ, 2008. С. 339–344.

TOOLS FOR LANGUAGE RESOURCES SOFTWARE DEVELOPMENT

Zagorulko M.Yu., Sidorova E.A.

*A.P. Ershov Institute of Informatics Systems
Siberian Branch of the Russian Academy of
Sciences, Novosibirsk, Russia*
zagorulko_maxim@yahoo.com

lena@iis.nsk.su

The paper presents software tools designed for linguistic research and development of linguistic resources: text corpora and domain-specific dictionaries. Technology of dictionaries creation is focused on subject terminology extraction, as well as lexical-semantic and syntactic and semantic structures are proposed.

INTRODUCTION

The development of natural language interfaces for users of information systems providing with more convenient means of access, search and text presentation call for solution of different linguistic tasks. The solution of these tasks requires certain linguistic resources, the creation of which (in their absence) is a separate task. Software tools designed for linguistic research and development of linguistic resources such as text corpora and domain-specific dictionaries is considered.

MAIN PART

During development of natural-language modules different linguistic tasks grow up and their solution requires the use of a variety of linguistic resources. In our process chain, we consider the following types of linguistic resources: text corpora, generic and subject dictionaries, semantic dictionary, document model, fact scheme.

Our tools support the creation of several types of dictionaries: terminological dictionary, lexical and semantic dictionaries, semantic and syntactic dictionaries. All systems are equipped with a custom dictionary editor of the resource, testing module and software API, which allows using the developed linguist expert resource offline.

Tools that support the creation and study of text corpora should include a means of preliminary marking of text, data retrieval, statistical information obtaining and providing the results to the user in a convenient form. This functionality is integrated in a specialized system to work with text corpora - the corpus-manager

CONCLUSION

This paper presents software tools for development of linguistic resources. Each tool can be used independently or as part of any configuration of the available modules. Further research will be aimed at methods and tools developing for learning and automatic creation of linguistic resources through the training corpus. Also we plan to develop different tools to study the semantic properties of the text.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89

ИНТЕЛЛЕКТУАЛЬНЫЕ ОБУЧАЮЩИЕ СИСТЕМЫ НА ОСНОВЕ АЛГЕБРАИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ВОПРОСНО-ОТВЕТНЫХ ТЕКСТОВ

Зуенко А.А. *, Кулик Б.А. **, Фридман А.Я. *

**Институт информатики и математического моделирования КНЦ РАН, Апатиты, Россия*

zuenko@iimm.kolasc.net.ru

fridman@iimm.kolasc.net.ru

***Институт проблем машиноведения РАН, Санкт-Петербург, Россия*

ba-kulik@yandex.ru

В статье описывается подход к разработке вопросно-ответных обучающих систем на основе контролируемых языков, а также алгебраических моделей представления и обработки вопросно-ответных текстов. Предлагается при построении индивидуальной траектории обучения использовать отношение частичного порядка “вопрос-подвопрос”. Модифицируемые рассуждения, моделирующие стратегию опроса учителя, формализуются с использованием разработанного авторами математического аппарата *QC*-структур.

Ключевые слова: интеллектуальная обучающая система, вопросно-ответный текст, *QC*-структуры, алгебра кортежей.

ВВЕДЕНИЕ

В интеллектуальных обучающих системах (ИОС) можно выделить четыре части: учебный материал, обучающий модуль, модуль контроля и модуль проверки [Таран, 2001]. Рассмотрим лишь две последние составляющие. Модуль контроля знаний проверяет усвоение учеником материала; модуль проверки тестирует знания и умения ученика и выставляет оценку. Отделение модуля контроля знаний от модуля проверки позволяет определить, каких именно знаний не хватает ученику. Однако, часто модуль контроля рассматривают не отдельно, а относят либо к модулю обучения, либо к модулю проверки.

В работах, где встречается термин «обучающие системы», в основном, можно найти идеи по построению проверяющих систем, которые в результате анализа решений определяют умения ученика (см., например, [Денисова, 2012]). Основной целью работы таких систем остается выставление оценки.

В отличие от подобных систем, преподаватель, получив неправильный ответ на задачу, пытается выявить “пробелы” в знаниях ученика, задавая наводящие вопросы и ставя дополнительные задачи. В результате, он либо “наводит” ученика на правильный ответ, либо определяет, какое именно

понятие, правило или теорему ученик не знает или не умеет применять.

По мнению авторов, назначение интеллектуальной системы контроля знаний, в отличие от системы проверки, заключается в том, чтобы, аналогично реальному преподавателю, не просто оценить уровень знаний ученика, а выявить плохо усвоенные знания и выработать рекомендации по дальнейшему закреплению и изучению материала.

Выявление “пробелов” в знаниях ученика осуществляется в процессе вопросно-ответного диалога, управляемого ИОС. Важную роль в таком диалоге играет блок, имитирующий стратегии опроса, применяемые преподавателем, и позволяющий пересматривать накопленные знания об уровне подготовки обучаемого.

Ситуация усложняется, когда вопросно-ответный диалог предполагает общение на естественном языке с возможностью задания ответа в “произвольной” форме. Следует уточнить, что термин “произвольная форма ответа” не вполне корректен, поскольку предполагается, что опрашиваемый знаком с учебным материалом, т.е. погружен в контекст и, в связи с этим, дает осмысленные ответы в терминологии, близкой к терминологии учебного материала.

В настоящее время наиболее перспективным при разработке диалоговых систем, допускающих свободную форму ответов пользователя, считается подход на основе концепции CNL (Controlled Natural Language). Контролируемый язык (упрощенный задачно-методологическим контекстом естественный язык) – это версия естественного языка, созданная для выполнения определенных задач [Pool, 2006]. Оригинальный вариант контролируемого языка вопросно-ответного диалога на основе концептуальных грамматик предложен в [Сулейманов, 2011]. Там же рассмотрена семантическая классификация вопросно-ответных текстов.

В статье предлагается подход к построению интеллектуальных обучающих систем на основе алгебраической интерпретации упомянутой модели вопросно-ответного диалога. В качестве математического аппарата для представления и анализа вопросно-ответных текстов выбраны разработанная авторами алгебра кортежей (АК) [Кулик и др., 2010], [Кулик, 1993], [Зуенко и др., 2010], [Kulik et al., 2010], а также развитие аппарата частично упорядоченных множеств – *QC*-структуры [Кулик, 2001].

Сначала рассмотрим более подробно принципы построения ИОС в рамках концепции контролируемых языков.

1. Организация вопросно-ответного диалога на основе концепции контролируемых языков

В условиях вопросно-ответного диалога, когда активной стороной является интеллектуальная обучающая система (ИОС), контекст достаточно четко определяет круг ожидаемых возможных ответов. Смысловая типизация вопросов и семантическая классификация ответных текстов дают возможность сопоставить каждому типу вопроса ограниченный набор допустимых, т.е. логически правильных, смысловых конструкций (ответных формул). Соответственно, содержание ответа, его лексикон, форма и отчасти объем предопределены, и пользователь с необходимостью отвечает на вопрос в определенных рамках [Сулейманов, 2011]. Действительно, если вопросы формируются на основе лекционных материалов, с которыми знаком обучаемый, то становится возможным задавать даже наборы ожидаемых лексических единиц при ответе на конкретный заданный вопрос. Этот подход делает возможным построение эффективной системы проверки правильности ответа на вопрос.

Система интерпретации ЕЯ-текстов в контексте, управляемом системой [Сулейманов, 2011], включает базу знаний, базу специфических грамматических конструкций, лексический процессор и семантический интерпретатор. Распознавание текста ответа включает перевод

текста в каноническое представление, выбор семантической схемы эталонного ответа, сопоставление ответа и соответствующей семантической схемы.

Результат формируется в виде некоторого вектора ситуации, представляющего собой набор показателей качества ответа, который используется для управления дальнейшим диалогом.

База знаний представляет собой совокупность моделей обучающего текста, множеств тестирующих вопросов и эталонных ответов. Модель вопросов и ответов строится в автоматизированном режиме специалистом по предметной области. Предложенная в работе [Сулейманов, 2000] классификация вопросов используется для построения шаблонов ответов.

Вопросы, относящиеся к первому типу, требуют явного задания в ответе ключевых параметров без учета отношений. Вопросы второго типа требуют раскрытия в ответе одного отношения, связанного с одним главным понятием. В вопросах третьего типа необходимо указать в ответе композицию фиксированного набора базовых отношений, связанных с главным понятием. Четвертый тип вопросов требует ответа в виде произвольной композиции отношений, связанных с одним понятием. В ответах на вопросы пятого типа допускается указывать несколько понятий, связанных произвольным набором отношений.

Разбиение текстов на семантические классы осуществляется на основе выявления главного (главных) понятия и отношений, связанных с этим главным понятием. Множество конкретных понятий и отношений по определенным признакам можно разбить на конечное число *типов понятий* и *типов отношений*. Согласно терминологии работы [Сулейманов, 2011], такие типы называются *семантическими единицами* или *концептулами*. Каждое осмысленное предложение предметной области можно перевести в текст, составленный из типов понятий и типов отношений, т.е. семантических единиц, без детального учета грамматических признаков лексем, соотнося каждое понятие или отношение с определенным типом. Представленным выше семантическим классам ответов соответствуют присущие им схемы сочетания концептул, передающие характерный (обобщенный) смысл ответов данного класса (значений вопросов). Схемы сочетания концептул, соответствующие правильной передаче ожидаемого смысла, названы *индивидуальными концептуальными грамматиками* (ИКГ).

В данной работе нас интересует, как осуществляется построение модели (шаблона) ответа.

Модель ответа строится на основе задаваемого вопроса и представляет собой пару $\langle F, G \rangle$. G обозначает ИКГ класса ответов, соответствующего

заданному вопросу. $F = \langle L, K \rangle$ представляет собой информационную структуру, содержащую лексемы L , отражающие понятия и отношения, а также их предполагаемые роли K в ответе.

В качестве учебного примера рассмотрим следующий обучающий текст (текст лекций):

"Компилятор – это программа, которая переводит исходный текст на ЯВУ в объектный текст на ЯМК и находится в оперативной памяти. Этап компиляции включает синтаксический, лексический анализы, оптимизацию и генерацию кодов и выполняется раньше этапа загрузки, которая из объектного модуля делает загрузочный и располагает в памяти. Редактор связей записывает загрузочный модуль на диск".

Пусть задан вопрос третьего типа: "Какую функцию выполняет компилятор?". Перечислим некоторые из возможных ответов на данный вопрос [Сулейманов, 2000]:

1. переводит исходный текст на языке высокого уровня в объектный текст в машинных кодах,
2. получает ЯМК из ЯВУ,
3. компилятор переводит ЯВУ в ЯМК.

Представленные ответы относятся к классу ФУНКЦИЯ. При описании грамматик ответов класса ФУНКЦИЯ используются следующие концептулы:

SS – концептула, отражающая главное понятие;
 SA – концептула, отражающая понятие-аргумент;
 SP – концептула, отражающая понятие-результат;
 GP_A – предлог перед SA ;
 GP_P – предлог перед SP ;
 RA – концептула, отражающая отношение SS к SA ;
 RP – концептула, отражающая отношение SS к SP .

Формализованное представление перечисленных ответов, соответственно, имеет вид:

- 1) $RA \rightarrow SA \rightarrow GP_P \rightarrow SP$
- 2) $RP \rightarrow SP \rightarrow GP_A \rightarrow SA$
- 3) $SS \rightarrow RA \rightarrow SA \rightarrow GP_P \rightarrow SP$

Здесь отношение "переводит" есть RA , отношение "получает" – RP , понятия "текст на языке высокого уровня", "ЯВУ" – SA , "текст в машинных кодах", "ЯМК" – SP , предлог "из" – GP_A ; предлог "в" – GP_P , понятие "компилятор" – SS . Стрелки отображают цепочку следования концептул в предложении.

Множество подобных формул, описывающих
Таблица 1.

ОТВЕТ: КЛАСС = ФУНКЦИЯ{

SS	RA	RP	SA	SP
{компилятор, транслятор}	{переводит, преобразует}	{получает}	{текст на языке высокого уровня, ЯВУ}	{текст в машинных кодах, программа на ЯМК, ЯМК}

// КОНЕЦ ОТВЕТ КЛАСС = ФУНКЦИЯ

порядок следования и роли лексем, образуют ИКГ. Информационная структура F представляется в следующем виде:

ОТВЕТ: КЛАСС = ФУНКЦИЯ

F : $SS =$ компилятор, транслятор; $RA =$ переводит, преобразует; $RP =$ получает; $SA =$ текст на языке высокого уровня, ЯВУ; $SP =$ текст в машинных кодах, программа на ЯМК, ЯМК.

Здесь перечислены лексемы (L), например "компилятор", и их семантические роли в предложении (K), например, концептула " SS ".

Далее для удобства представления подобные структуры будем записывать в виде таблиц (таб. 1).

В данном случае информационная структура F представима в виде декартова произведения множеств значений концептул. В более общем случае, когда требуется учесть еще и сочетаемость лексем, выступающих в качестве значений концептул, информационная структура F может быть записана в виде системы множественных отношений.

Для моделирования и обработки типовых структур знаний, используемых в вопросно-ответных компонентах ИОС, предлагается применить алгебру кортежей [Кулик и др., 2010], [Kulik et al., 2012]. В частности, шаблоны ответов компактно записываются в виде S -систем.

Далее приводится решение задачи автоматизации контроля знаний обучаемого.

2. Алгебраическая модель вопросно-ответного диалога

Система, осуществляющая контроль знаний, относится к открытым системам, способным пополнять свои знания и изменять построенные ранее выводы при изменении ситуации, реально отражая картину обучения. Для формализации рассуждений в системах такого типа применяют аргументацию.

Учитель, опрашивая ученика, обычно имеет некоторую последовательность вопросов, в случае неправильного ответа на один из них задаются подвопросы, помогающие выявить пробелы в знаниях обучаемого. Опрос длится до тех пор, пока учитель не получит достаточно аргументов для обоснования своего решения об уровне знаний обучаемого и/или выработки рекомендаций по дополнительному изучению материала.

Введем на множестве вопросов отношение подчинения S , смысл которого заключается в следующем: x подчинен y , если для правильного ответа на вопрос y требуется знание верного ответа на вопрос x . Допустим, имеются вопросы с введенным на них отношением подчинения:

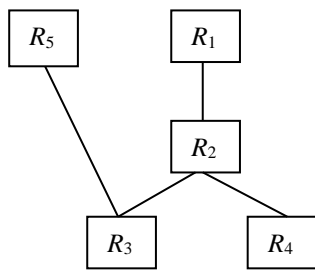


Рисунок 1 – Графическое представление множества вопросов

- $R1$. Что Вы знаете о компиляторе и процессе компиляции? (Вопрос 5 типа)
 $R2$. Дайте определение компилятора? (Вопрос 4 типа)
 $R3$. Где находится компилятор? (Вопрос 2 типа)
 $R4$. Какую функцию выполняет компилятор? (Вопрос 3 типа)
 $R5$. Для чего служит оперативная память ПК? (Вопрос 3 типа)

Из рисунка 1 видно, что вопросы, чей тип сложнее, доминируют над вопросами более простых типов. Например, вопрос R_2 четвертого типа доминирует над вопросом R_4 третьего типа. Другими словами, принятая нами классификация вопросно-ответных текстов позволяет естественным образом упорядочивать множество вопросов, предполагающих в ответе наличие общих понятий и отношений. Согласно [Сулейманов, 2011], каждому вопросу сопоставляется модель ответа, из которой нас далее будет интересовать информационная структура F . Модели ответов для подвопросов содержат лишь часть отношений, предусмотренных самими вопросами. Это можно увидеть, сравнивая модели ответов на вопросы R_2 и R_4 .

Таблица 2.

ОТВЕТ КЛАСС = ОПРЕДЕЛЕНИЕ

SS (главное понятие)	$Soп$ (более общее понятие, чем главное)
{компилятор, транслятор}	{программа}

КЛАСС = ФУНКЦИЯ

R_A	R_P	SA	SP
{переводит, преобразует}	{получает}	{текст на языке высокого уровня, ЯВУ}	{текст в машинных кодах, программа на ЯМК, ЯМК}

КЛАСС = ПРО

SO	R_{so}	R_{os}
{оперативная память}	{находится, содержится}	{содержит}

// КОНЕЦ ОТВЕТ КЛАСС = ОПРЕДЕЛЕНИЕ.

Модель ответа для вопроса R_4 мы уже рассмотрели в предыдущем разделе. Вопросу R_2 на основе приведенного выше текста лекций может быть сопоставлена такая модель ответа (таблица 2).

Соединение всех трех отношений, записанных в виде таблиц, и формирует АК-объект [Аюпов и др., 2012], [Зуенко и др., 2011], задающий модель ответа на вопрос R_2 .

Основными процедурами аргументации являются процедуры определения истинностных оценок утверждений по соответствующим им множествам аргументов. Множество высказываний и соответствующих аргументов задается как контекст $K=(V, A, I)$, где V – множество высказываний, A – множество аргументов, I – отношение на множестве $V \times A$, такое что pIa^σ , где $p \in V, a^\sigma \in A, \sigma \in \{+, -\}$, тогда и только тогда, когда a^σ есть аргумент высказывания p [Таран, 2001].

В качестве элементарных высказываний при построении логики аргументации рассматриваются утверждения вида: “учащийся знает ответ на данный вопрос”. Если правильные ответы трактовать как аргументы “за” подобные утверждения, а неправильные как аргументы “против”, то отношение S на множестве вопросов позволяет строить логику аргументации с упорядоченным множеством аргументов [Вагин и др., 2008].

Пусть с вопросом R_i связано высказывание p_i (“учащийся знает ответ на i -ый вопрос”), а Q_j вопросу сопоставлено высказывание q_j (“учащийся знает ответ j -ый вопрос”). Тогда при выполнении соотношения $A_i \subseteq_G A_j$ (согласно принятой сегментации вопросно-ответного текста, данное соотношение всегда выполняется, если для вопросов R_i и Q_j верно, что R_i доминирует Q_j) можно утверждать, что $V[(p_i \rightarrow q_j)] = 1$ (V – функция оценивания).

Если модель ответа включает не только образцы правильных ответов, но запрещает некоторые ответы или отдельные лексемы, то каждому утверждению p_i , помимо АК-объекта A_i , может быть сопоставлен АК-объект B_i , описывающий запрещенные ответы. В терминологии работы [Финн, 1991] АК-объект A_i для высказывания p_i задает область значений функции g^+ , а АК-объект B_i – область значений функции g^- .

Анализ множеств аргументов позволяет выявлять причинно-следственные связи предметной области. Эта возможность обусловлена определением импликации в логике аргументации. Достаточным условием истинности импликаций является выполнение следующих условий [Финн, 1991]:

Если

$$g^+(p) \subseteq g^+(q) \text{ и } g^-(q) \subseteq g^-(p), \text{ то } V[(p \rightarrow q)] = 1$$

$$g^+(p) \subseteq g^-(q) \text{ и } g^+(q) \subseteq g^-(p), \text{ то } V[(p \rightarrow \neg q)] = 1$$

На основе анализа множества аргументов можно получать два вида утверждений $p \rightarrow q$ и $p \rightarrow \neg q$. Выявление и устранение коллизий (понятие коллизия здесь используется как обобщение понятий атака аргумента, подрыв аргумента и т.п.) в системах рассуждений такого типа предлагается выполнять на основе разработанного авторами аппарата QC -структур [Кулик, 2001].

Рассмотрим один из возможных сценариев реализации процесса контроля с использованием упомянутых средств логического анализа, когда проверка начинается с самого сложного вопроса (см. рисунок 1). Предположим, ученик неправильно ответил на вопрос R_1 . В этом случае система задает ему вопрос R_2 . Если на него обучающийся ответил неправильно, то система задает вопрос R_3 . Если на вопрос R_4 получен правильный ответ, то ученику повторно задается вопрос R_3 . Если на вопрос R_3 получен правильный ответ, то предлагается снова ответить на вопрос R_2 , а также предлагается вопрос R_5 . В случае правильных ответов на данные вопросы предлагается снова ответить на вопрос R_1 .

ЗАКЛЮЧЕНИЕ

Авторами предложены алгебраические модели представления и обработки вопросно-ответных текстов, предназначенные для решения задачи контроля знаний обучаемого. Разработанные модели открыты для пополнения знаний. Они позволяют осуществлять контроль и формировать индивидуальную траекторию обучения с помощью процедур аргументации.

Работа выполнена при поддержке РФФИ (проекты № 12-07-006689-а и № 12-07-000550-а), Президиума РАН (проект 4.3 Программы № 15), ОНИТ РАН (проект 2.3 в рамках текущей Программы фундаментальных научных исследований).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Аюпов и др., 2012] Аюпов, М.М. Подход к построению вопросно-ответных обучающих систем на базе сетей множественных отношений / М.М. Аюпов, Б.А. Кулик, О.А. Невзорова, Д.Ш. Сулейманов, А.Я. Фридман // Тринадцатая национальная конференция по искусственному интеллекту с международным участием (16-20 октября 2012г., г. Белгород, Россия): Труды конференции. Т. 2.- Белгород: Изд-во БГТУ, 2012. – С.152-160
- [Вагин и др., 2008] Достоверный и правдоподобный вывод в интеллектуальных системах / Вагин В.Н. [и др.]; – М.: ФИЗМАТЛИТ, 2008.
- [Денисова, 2012] Денисова, И.Ю. Математические модели онтологии базы данных информационной обучающей системы / И.Ю. Денисова, П.П. Макарычев // Онтология проектирования. – 2012. – № 3. – С.62-78.
- [Зуенко и др., 2010] Зуенко, А.А. Унификация обработки данных и знаний на основе общей теории множественных отношений / А.А. Зуенко, Б.А. Кулик, А.Я. Фридман // Искусственный интеллект и принятие решений. – 2010. – № 3. – С.52-62.
- [Зуенко и др., 2011] Зуенко, А.А. Интеграция баз данных и знаний интеллектуальных систем на основе алгебраического подхода / А.А. Зуенко, Б.А. Кулик, А.Я. Фридман // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2011): материалы Междунар. научн.-техн. конф. (Минск, 10-12 февраля 2011 г.) – Минск: БГУИР, 2011. – С.59-70.
- [Кулик, 1993] Кулик, Б.А. Система логического программирования на основе алгебры кортежей / Б.А. Кулик // Известия РАН. Техническая кибернетика. – 1993. – № 3. – С. 226-239.
- [Кулик, 2001] Кулик, Б.А. Логика естественных рассуждений / Б.А. Кулик // – СПб.: Политехника. 1997.
- [Кулик и др., 2010] Кулик, Б.А. Алгебраический подход к интеллектуальной обработке данных и знаний / Б.А. Кулик, А.А. Зуенко, А.Я. Фридман // – СПб.: Изд-во Политехнического ун-та, 2010. 235 с.
- [Сулейманов, 2000] Сулейманов Д.Ш. Системы и информационные технологии обработки естественно-языковых текстов на основе прагматически-ориентированных лингвистических моделей. Автореферат диссертации на соискание ученой степени доктора технических наук. Казань, 2000.
- [Сулейманов, 2011] Сулейманов, Д.Ш. Двухуровневый лингвистический процессор ответных текстов на естественном языке / Д.Ш. Сулейманов // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2011): материалы Междунар. научн.-техн. конф. (Минск, 10-12 февраля 2011 г.) – Минск: БГУИР, 2011. – С. 311-322.
- [Таран, 2001] Таран, Т.А. Аргументационная система контроля знаний / Т.А. Таран, А.И. Ривкинд // Новости искусственного интеллекта. 2001. № 5-6, с. 12-18.
- [Финн, 1991] Финн, В.К. Об одном варианте логики аргументации // В.К. Финн/ НТИ: Сер.2. – 1996 – № 5-6 – С. 3-19.
- [Kulik et al., 2010] Kulik, B. Logical Analysis of Intelligence Systems by Algebraic Method / Boris Kulik, Alexander Fridman, Alexander Zuenko // Cybernetics and Systems 2010: Proceedings of Twentieth European Meeting on Cybernetics and Systems Research (EMCSR 2010), Vienna, Austria, 2010. – pp. 198-203.
- [Kulik et al., 2012] Boris Kulik, Alexander Fridman, Alexander Zuenko. Logical Inference and Defeasible Reasoning in N-tuple Algebra. In: “Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems”, IGI Global, P 102-128.
- [Pool, 2006] Jonathan Pool. Can controlled languages scale to the Web? // In Proceedings of the 5th International Workshop on Controlled Language Applications (CLAW 2006), 2006.

INTELLIGENT TEACHING SYSTEMS BASED ON ALGEBRAIC REPRESENTATION OF QUESTION-AND-ANSWER TEXTS

Zuenko A.A.*, Kulik B.A.***, Fridman A.Ya.*

**Institute for Informatics and Mathematical Modelling, Kola Science Centre of RAS*

zuenko@iimm.kolasc.net.ru

fridman@iimm.kolasc.net.ru

***Institute of Problems in Machine Science of RAS*

ba-kulik@yandex.ru

The paper introduces an approach to development of question-and-answer teaching systems using controlled natural languages and algebraic models to represent and process question-and-answer texts. We propose to build a personal teaching trajectory by means of a partial order relation "question-subquestion". A mathematical apparatus developed by the authors and called *QC*-structures allows to formalize defeasible reasoning that models examination strategy of a teacher.

INTRODUCTION

Unlike a testing system, an intelligent system checking somebody's knowledge is to work as a real teacher who does not simply check the knowledge level of a student, but reveal poorly digested knowledge and recommend how to acquire and consolidate it.

Lately, approaches based on the concept of CNL (Controlled Natural Language) are considered most popular ones for development of interactive systems admitting plain-text form of answers. A CNL that is a natural language simplified by using a problem methodology context is a version of the natural language composed to solve certain problems. An original modification of CNLs as well as a classification of question-and-answer texts we use was proposed by D.Sh. Suleymanov.

In this paper, we discuss constructing of teaching systems based on an algebraic interpretation of the above-mentioned model of a question-and-answer dialog. The interpretation uses our earlier developed *n*-tuple algebra (NTA) and *QC*-structures, which are a modification of partially ordered sets, as a mathematical ground for our research.

MAIN PART

Knowledge checking systems relate to open systems capable to replenish it's knowledge and change some earlier inferred conclusions to correspond with a real situation in learning when it changes. To formalize reasoning in such systems, logic of argumentation is applicable. Argumentation-based proof procedures have to model examination strategy of a teacher.

A teacher examining a student usually uses a certain sequence of questions. If a question is answered wrong, the teacher asks some subquestions to reveal deficiencies in the student's knowledge. To model this, we define a owner-member relationship *S* on the set of questions. xSy if you need to know the right answer to the question *x* in order to give the right answer to the question *y*.

The proposed classification of question-and-answer texts allows to naturally order a set of questions implying answers with common concepts and relations. Every question R_i corresponds to a pattern of answer that describes a set of right answers and can be formalized as a structure A_i within our NTA. If this pattern comprises not only samples of right answers but forbids certain answers and/or separate tokens, then we can connect any question R_i not only with an NTA object A_i , but also with another NTA object B_i describing forbidden answers. To build an logic of argumentation, we use the statements like "a student knows the right answer to the given question" as elementary propositions. If we consider right answers as arguments "pro" and wrong answers as arguments "contra", the relation *S* provides building an logic of argumentation with an ordered set of arguments.

We propose to construct reasoning itself, namely analysis of hypotheses, detection and elimination of collisions, which are generalizations of concepts like "rebutting," "argument undercutting," "counter-evidence (attack) etc., on the basis of *QC*-structures.

CONCLUSION

To check knowledge of a student, we develop algebraic models for representation and processing of question-and-answer texts. The models are open for replenishing of included knowledge. They allow for checking and forming a personal teaching trajectory by means of argumentation-based proof procedures.

The authors would like to thank the Russian Foundation for Basic Research (grants 12-07-00302, 11-08-00641, 12-07-00550, 12-07-689), the Department for Nanotechnologies and Information Technologies of RAS (project 2.11 of the current Programme of Basic Scientific Researches), and the Chair of RAS (project 4.3 of the Program # 16) for their help in partial funding of this research.



УДК 004.822:514

УНИФИЦИРОВАННОЕ ПРЕДСТАВЛЕНИЕ И ИНТЕГРАЦИЯ ЗНАНИЙ

Ивашенко В.П.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

ivashenko@bsuir.by

Рассматриваются составляющие и применение средств технологии компонентного проектирования баз знаний в виде однородных семантических сети с теоретико-множественной семантикой для решения задач отладки и интеграции баз знаний.

Ключевые слова: база знаний, онтология, семантическая сеть, интеграция.

ВВЕДЕНИЕ

Рассматриваются задачи интеграции знаний в рамках задачи разработки базы знаний для интеллектуальной системы [Ивашенко, 2009a].

Интеграция знаний заключается в погружении новых знаний в состав знаний, уже известных интеллектуальной системе, и является необходимым этапом понимания информации, которая либо поступает в интеллектуальную систему извне, либо генерируется (порождается) в процессе обработки информации. Процесс понимания в интеллектуальной системе заключается не только и не столько в помещении добавляемых новых знаний в память, но – в выявлении синонимов, связывании и отождествлении поступающих фрагментов с уже известными. Таким образом, интеграция знаний является основой функционирования любой интеллектуальной системы и во многом определяет качество интеллектуальной системы, являясь последним и основным этапом понимания поступающих (приобретаемых) знаний.

Разработка баз знаний наталкивается на следующие трудности:

- отсутствие развитых технологий разработки или их ограничения:
 - ограниченные возможности верификации и отладки баз знаний (отсутствие средств верификации количества синонимичных знаков, неполное выявление противоречий);
 - плохая отчуждаемость и переносимость базы знаний в силу привязки к инструментарию или конкретной оболочке (CLIPS) [CLIPS, 1991],

ограниченность языков и моделей представления знаний

- немногочисленность инженеров баз знаний (из-за высоких стартовых требований к разработчику) – от разработчика требуется владение специальными знаниями по моделям и языкам представления знаний):
 - необходимость выбора среди нескольких моделей знаний, между которыми нет однозначного предпочтения [Martin Ph., 2002];
 - неоднородность моделей представления знаний, приводящая к тому, что в базе появляются синонимичные структуры разных типов, требующие от пользователя их согласования;
 - сложность языков представления знаний
- не полностью решён вопрос интеграции баз знаний:
 - ограничения на расширение базы знаний;
 - ограниченные иерархической таксономической структурой онтологии возможности интеграции баз знаний или отсутствие таких возможностей, необходимость выбора средств интеграции пользователем [Gangemi et al., 1996];
 - отсутствие общих стандартов совместимости разработанных фрагментов баз знаний;
 - ограниченность средств поиска и каталогизации разработанных фрагментов баз знаний;
 - ограниченность или отсутствие простых средств взаимодействия с внешней средой.

Для разработки баз знаний используются:

- языки представления и обработки баз знаний;
- средства создания и отладки баз знаний;
- средства интеграции баз знаний.

К языкам представления знаний в разных системах на сегодняшний день относятся: Conceptual Graph [Sowa et al., 2008], Frame-logic [Michael Kifer et al., 1995], Knowledge Interchange Format [Genesereth et al., 1992], Integrated Definition for Ontology Description Capture Method (IDEF5) [IDEF5, 1994], Common Algebraic Specification Language [CoFI:CASL-Summary, 2004], Concept maps/UML [Novak et al., 2008] [ISO24707], RDF/RDFS [W3C:RDFS, 2004], DARPA Agent Markup Language [DAML, 2006], CycL [CycL, 2002], Ontology Inference Layer [W3C:DAML+OIL, 2001], RDF/OWL-Lite, RDF/OWL-DL, RDF/OWL Full [W3C:OWL, 2004], Topic Maps [ISO13250], XTM/LTM/CML/GML [XTM, 2001], Common Logic [ISO24707], Developing Ontology-Grounded Methods and Applications [Mustafa Jarrar et al., 2008], Formal English [Martin Ph., 2002], Gellish [Van Renssen, 2005], RDF/Rule Interchange Format [W3C:RIF, 2010], Open Biomedical Ontologies [Smith et al., 2007], RDF/OWL2 [W3C:OWL2, 2009], RDF/OWL2 EL, RDF/OWL2 RL и др.

Эти языки могут быть классифицированы по различным признакам:

- синтаксические признаки
 - URI-ориентированные
 - линейные языки
 - мультиплетные языки
 - триплетные языки
 - квинтиплетные языки
 - иерархически структурированные языки
 - ЛИСП-подобные языки
 - нелинейные языки
 - графовые языки
 - гиперграфовые языки
- семантические признаки
 - фреймовые языки
 - правила
 - семантическая сеть
 - семантический гиперграф
 - логические языки
 - дескрипционной логики
 - логики предикатов первого порядка
 - логики предикатов высших порядков
 - модальной логики
 - темпоральной логики (линейной, ...)
 - взаимодействующих процессов Хоара
 - объектно-ориентированные языки
 - процедурные языки
 - естественный язык

Кроме очевидных, дополнительно к недостаткам [Martin Ph., 2002] вышеперечисленных языков представления знаний можно отнести:

- отсутствие разделения понятий и терминов (исключения – DOGMA и Gellish),
- отсутствие поддержки монотонного расширения базы знаний (исключения – OWL, OWL2 и языки, построенные на основе классических логических моделей),
- отсутствие у некоторых из вышеперечисленных языков возможности семантического расширения языка (замкнутость языка).

Для поиска в базах знаний и онтологиях [Хорошевский, 2008] используются такие языки, как: RDQL, squish, SPARQL [W3C:SPARQL, 2008], KQML, DMX, Datalog, TSQL, ERROL, RuleML, RQL, OQL, TQL, VERSA, DQL и др.

Среди средств, которые могут рассматриваться в качестве основы для разработки баз знаний, можно выделить: оболочки экспертных систем (CLIPS (FuzzyCLIPS, DYNACLIPS, WxCLIPS) [CLIPS, 1991], SOAR, OPS83, RT-EXPERT, MIKE, BABYLON, WindExS, ES; ACQUARE, Easy Reasoner, ECLIPSE, EXSYS Professional, SIMER+MIR, AT ТЕХНОЛОГИЯ, CAKE v2.0) [Гаврилова и др., 2000]; инструментальные пакеты для разработки экспертных систем (G2, ART, KEE, Knowledge KRAFT); системы, ориентированные на обработку онтологий [Sowa et al., 2008] – Protégé, WebOnto, OntoEdit, WebODE, OilEd, OntoLingua.

Достоинствами приведённых средств являются: поддержка представления знаний различного вида различными моделями представления знаний в рамках одной системы; наличие средств визуального проектирования баз знаний; наличие средств верификации базы знаний, включая проверку на непротиворечивость; возможность монотонного расширения базы знаний, наличие средств интеграции баз знаний; наличие средств поддержки обмена данными с внешней средой, включая средства обмена данными в реальном времени.

Для преодоления трудностей семантической интеграции [Doan and Halevy, 2005], [Кудрявцев, 2008] (отображения онтологий (ontology mapping) и интеграции знаний (knowledge integration) в базах знаний используются следующие подходы:

- сравнение и выравнивание онтологий (ontology matching & alignment),
- интеграция онтологий (ontology merging),
- семантическое сравнение (semantic matching),
- семантическая унификация (semantic unification).

Для интеграции онтологий можно выделить методы:

- структурно-синтаксические:
 - анализ внутренней структуры
 - экстенционально-статистические
 - анализ внешней структуры (метаструктурный анализ)
 - терминологические (лексические)
- логико-семантические

К наиболее развитым подходам и методам интеграции можно отнести ONION (ONtology composition [Mitra, 2001]), формальный концептуальный анализ (FCA [Ganter, 1999], [Stumme, 2001]) и варианты его развития для нечётких (FOGA [IEEE, 2006]) и неопределённых множеств, методы использующие элементы семантического анализа – СТХМАТЧН [Bouquet, 2003] и S-match. Однако все эти методы, несмотря на использование некоторыми из них нечётких и неопределённых множеств, плохо приспособлены или не приспособлены к интеграции знаний в условиях наличия НЕ-факторов [Нариньяни, 2000]. Перечисленные методы, и методы, основанные на мерах близости (CUPID [Maldavan, 2001] и т.п.), не обеспечивают достаточной формальной строгости и непротиворечивости онтологий или баз знаний, получаемых в результате.

К существующим на настоящий момент средствам интеграции онтологий можно отнести: Optima, Prompt, Ontolingua, Chimaera [McGuinness et al., 2000], ONION [Gangemi et al., 1996], COMA++ [Aumueller et al., 2005] и др.

Таки образом, все эти средства и методы имеют ограничения и не преодолевают в полной мере вышеперечисленные трудности.

1. Модель унифицированного нелинейного представления знаний

Семантическая модель интеграции использует унифицированное представление знаний и обеспечивает интеграцию sc-моделей баз знаний и их фрагментов [Ивашенко, 2012]. Унифицированное представление знаний обеспечивается моделью унифицированного нелинейного представления знаний, которая является частным случаем такой модели представления знаний, как семантические сети и задаётся семейством совместимых sc-языков, использующих унифицированный способ семантического кодирования Semantic Computer code (SC-код) [Голенков и др, 2001] и поддерживающих представление знаний различного вида [Ивашенко, 2003], [Ивашенко, 2004], [Ивашенко, 2011a]. Особенности SC-кода являются: простой алфавит, содержащий узлы и дуги, простой синтаксис, базовая теоретико-множественная интерпретация. Семантика sc-языка задаётся на основе модели ситуативных множеств, являющейся развитием моделей L-нечётких множеств и неоднородных нечётких множеств, предложенных Дж. Гогеном и А. Кофманом [Кофман, 1982].

Модель ситуативных (событийных, нестационарных [Ивашенко, 2012]) множеств может быть задана следующей шестёркой компонентов:

$$\langle Universe, [0;1], Events, r, h, sM \rangle \quad (1)$$

где $Universe$ – универсальное множество объектов предметной области, $Events$ – множество

элементарных событий, $r \subseteq Events \times Events$ – отношение доступности (следования во времени) событий, $h \in (2^{Events})^{Universe}$ – функция, задающая множество событий существования каждого элемента универсального множества, sM – семейство пар множества событий существования ситуативного множества и функций (нечёткой) ситуативной принадлежности элементов универсального множества ситуативному множеству, отображающих элементы ситуативных множеств, множества событий и соответствующие им наборы степеней нечёткой принадлежности высших порядков на множество степеней нечёткой принадлежности $[0;1]$.

Для вычисления степеней нечётких принадлежностей введены параметризованные выражения для верхних и нижних треугольных норм и выразимых через них операций, задающих соответственно значения степеней принадлежности для операций пересечения, объединения и разности ситуативных множеств.

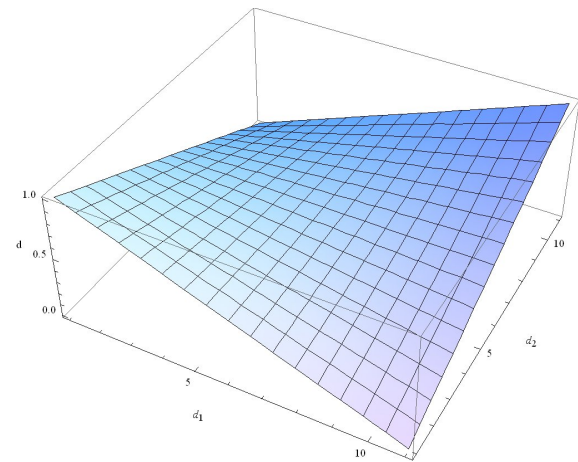


Рисунок 1– график функции нечёткой принадлежности элемента универсального нечёткого множества третьего порядка (d_1 и d_2 – аргументы, компоненты вектора нечётких степеней принадлежности, d – значение).

Выражение для верхней треугольной нормы: x и y – основные аргументы, значения принадлежностей исходных множеств, u – параметр, значение принадлежности универсального нечёткого множества (рис. 1).

$$tnormu(\langle x, y, u \rangle) = \max(\{x\} \cup \{y\} \cup \{u\}) + \min(\{x\} \cup \{y\} \cup \{u\}) - u \quad (2)$$

Нижняя s-норма.

$$snormn(\langle x, y, u \rangle) = x + y - tnormu(\langle x, y, u \rangle) \quad (3)$$

Верхняя s-норма.

$$snormu(\langle x, y, u \rangle) = snormn(\langle x + y + u - 1, u, (x + y) / 2 \rangle) \quad (4)$$

Верхняя разностная форма.

$$dformu(\langle x, y, u \rangle) = tnormu(\langle x, 1 - y, u \rangle) \quad (5)$$

Нижняя разностная форма.

$$dformn(\langle x, y, u \rangle) = x + 1 - u - tnormu(\langle x, y, u \rangle) \quad (6)$$

Нижняя треугольная норма.

$$tnormn(\langle x, y, u \rangle) = dformn(\langle x, 1 - y, u \rangle) \quad (7)$$

При переходе от ситуативных множеств низших порядков (нечёткости) к множествам высшим порядков используются выражения, позволяющие переычислять значения степени нечёткой принадлежности $degree(\langle b, V \rangle)$, исходя из модели равновероятных исходов в конечной темпоральной модели ветвящегося времени. Для этого используется базовое значение вероятности b , которое позволяет рассчитать степени принадлежности высших порядков по рекуррентному выражению для $f(V)$.

$$degree(\langle b, V \rangle) = b + (1 - 2 * b) * f(V) \quad (8)$$

$$f(V) = \min(\{1\} \cup \{V_1 * f(tail(V)) | \dim(V) > 0\}) + \max(\{0\} \cup \{(1 - V_1) * f(inverse(V)) | \dim(V) > 1\}) \quad (9)$$

$$inverse(V) \in \times_{i=2}^{\dim(V)} \{1 - V_i - \text{sgn}(i - 2) * (1 + 2 * V_i)\} \quad (10)$$

где $tail(V)$ – вектор всех компонентов вектора V , начиная со второго компонента, а $\dim(V)$ – размерность вектора V .

В соответствии с введённой моделью ситуативных множеств для описания динамических предметных областей, используются понятия нестационарной принадлежности и непринадлежности.

Модель унифицированного нелинейного представления знаний задаётся следующими компонентами.

$$\langle SClanguages, R_{sc}, F_{sc} \rangle \quad (11)$$

$$F_{sc} = A_{sc} \cup I_{sc} \cup N_{sc} \cup S_{sc} \cup E_{sc} \cup K_{sc}$$

$SClanguage$ – множество sc-языков, R_{sc} – отношения на множестве sc-языков (sc-подъязыка и трансляции), F_{sc} – функции sc-языков, A_{sc} – алфавитные функции sc-языков, I_{sc} – функции отношений инцидентности, N_{sc} – синтаксические предикаты sc-языков, S_{sc} – семантика sc-языков, E_{sc} – ключевые элементы sc-языков, K_{sc} – спецификация sc-языков и их ключевых элементов (отображение на множество онтологий sc-языков).

Semantic Code ориентирован на универсальность, т.е. на представление любых видов знаний. Semantic Code является средством унификации представляемых знаний. На множестве sc-языков (языков, представленных в

SC-коде) определены отношения sc-подъязыка и трансляции. Подъязык, являющийся пересечением выделенного семейства совместимых специализированных sc-языков, рассматривается как интегрированный sc-язык представления знаний (SCK). Основным принципом построения sc-языков является представление понятий, соответствующих основным классам объектов, описываемых sc-языком, и отношений между этими объектами ключевыми узлами такого sc-языка: каждому sc-языку однозначно сопоставляется конечное множество ключевых узлов (элементов) этого языка. Каждый ключевой узел задаёт ограничения на собственную семантическую окрестность в информационных конструкциях (текстах) этого языка. Множество ключевых элементов интегрированного sc-языка представления знаний является объединением множеств ключевых элементов остальных sc-языков выделенного семейства. Характеристиками sc-языка являются: мощность множества ключевых узлов sc-языка; семейство множеств собственных семантических окрестностей ключевых элементов языка; наличие функциональных зависимостей между собственными окрестностями множеств ключевых элементов sc-языка; соотношение алгоритмических сложностей поиска или вычисления элементов собственных семантических окрестностей на основании существующих зависимостей. Построена семантическая онтология существующих языков и моделей представления знаний, в которой указано место унифицированной модели представления знаний и её соотношение с существующими моделями представления знаний.

Введенные в sc-языки ключевые элементы поддерживают соответствующее семантически эквивалентное представление концептов OWL 2 QL и OWL 2 EL [W3C:OWL2, 2009], а также поддерживают представление множеств, мультимножеств, ситуативных множеств, отношений, включая ролевые отношения – строгие подмножества отношения принадлежности, простых, целых и рациональных чисел, логических формул. Остальные концепты могут быть определены с помощью логического sc-языка.

Модель унифицированного нелинейного представления знаний поддерживает представление знаний в условиях влияния различных НЕ-факторов: неполноты, неопределённости, нечёткости и пр. Далее приведены примеры представления нечётких и изменяющихся во времени знаний.

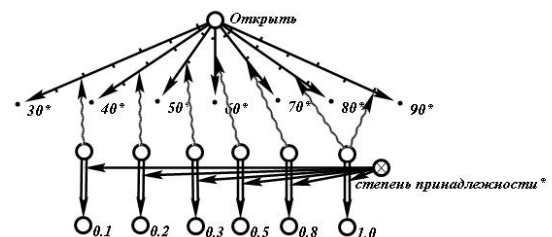


Рисунок 2– Пример представления нечёткого множества

Следующее нечёткое множество [Тэрano, 1993] описывает понятие открытия вентиля (рис. 2).

$$\text{Открыть} = 0,1/30^* + 0,2/40^* + 0,3/50^* + 0,5/60^* + 0,8/70^* + 1,0/80^* + 1,0/90^* \quad (12)$$

Степень нечёткой принадлежности указывается с помощью ключевого узла *степень принадлежности**, обозначающего бинарное отношение, связывающее sc-множество нечётких sc-дуг с числом, являющимся степенью принадлежности sc-элемента, в который входит нечёткая sc-дуга, sc-множеству, из которого она выходит. Следует отметить, что принадлежность нечётких sc-дуг классам с одинаковой степенью нечёткой принадлежности, в общем случае имеет нестационарную природу, что выражено соответствующими sc-дугами нестационарной принадлежности.

При представлении изменяющихся неточных и неопределённых знаний используются следующие языковые средства [Allen, 1991], [Allen, 1991] – временные отношения. В sc-языке временные отношения определяются на множестве sc-элементов, обозначающих нестационарные принадлежности и непринадлежности или их sc-множества, и выражают временные соотношения значений этих обозначений (знаков). Чтобы выразить временные отношения на самих обозначениях используется специальные ключевые узлы, связывающие значения одних обозначений, с обозначениями, временные отношения на которых необходимо выразить: такими узлами являются «полностью представленное множество», «sc-множество с полностью представленным экстенциональным замыканием» и другие.

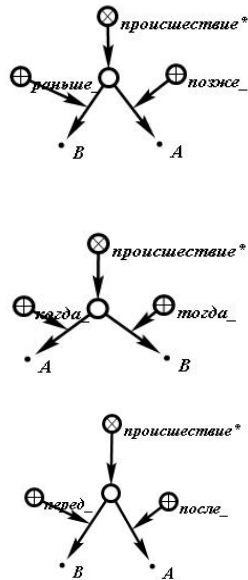


Рисунок 3– Примеры связей временных отношений.

Временное отношение, являющееся объединением всех неролевых временных отношений, обозначается ключевым узлом *происшествие**. Элементы связей этого отношения

отмечаются с помощью ролевых временных отношений (рис. 3). Временные отношения определяются в соответствии с отношением следования событий модели ситуативных множеств. Временное отношение, обозначаемое ключевым узлом *после_*, выражает непосредственное следование всех событий одного множества за каждым событиями другого, к которым относится соответствующая принадлежность или непринадлежность или каждая из их sc-множества. Отношение *после_* является иррефлексивным. Отношение, обозначенное ключевым sc-узлом *перед_*, является двойственным и обратным отношению *после_*. Отношения *позже_* и *раньше_* соответственно выражают транзитивные замыкания отношений *после_* и *перед_*. Отношение *когда_* выражает включение всех соответственных событий первого множества во второе множество событий или (в слабом случае) существование во втором множестве простых целей событий, заданных отношением следования таких, что все события первого множества лежат на простых цепях, соединяющих начало какой-либо из этих цепей с её концом. Временное отношение *тогда_* является двойственным отношением *когда_*.



Рисунок 4– Пример представления временного отношения на нестационарных принадлежностях и заданных ими знаках множеств.

Рассмотрим представление знания (рис. 3) о том, что когда множество слева (обозначенное узлом с точкой) полностью представлено (в памяти системы представлены все обозначения его элементов и принадлежностей), то представлено и множество (обозначенное узлом с точкой), находящееся справа.

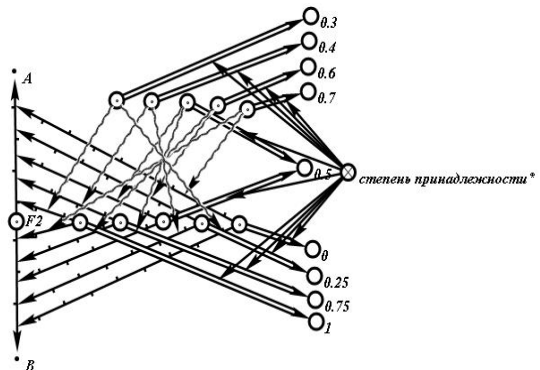


Рисунок 5– Пример представления нечёткого множества с нечётко заданными степенями принадлежности.

Для указания степеней нечёткой принадлежности элементов нечёткого множества второго порядка (2-типа) $F2$ (рис. 5), использованы пять дополнительных нечётких множеств первого порядка.

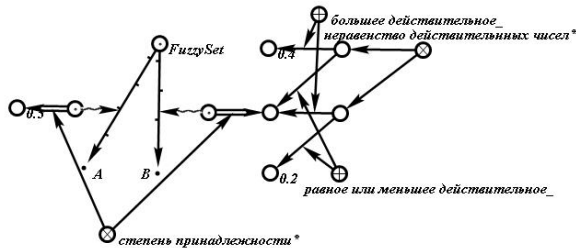


Рисунок 6– Пример представления нечёткого множества с неопределённой степенью принадлежности.

В случае, когда степень принадлежности точно указать нельзя, но можно оценить её значения сверху или снизу, для представления используются конструкции sc-языка арифметики. В нечётком множестве $FuzzySet$ (рис. 6), степень принадлежности которому элемента A известна, а степень принадлежности элемента B задана двумя числовыми рациональными оценками сверху и снизу.

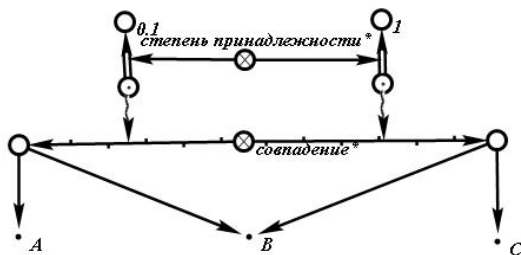


Рисунок 7– Пример представления нечёткого отношения.

Нечётко заданное отношение совпадения может использоваться при интеграции знаний в процессе выявления синонимов (рис. 7). Степень (нечёткой принадлежности связки отношения) совпадения sc-элемента A с sc-элементом B оценена в 0,1, а степень совпадения B с C – в единицу. Нечёткие sc-дуги принадлежности, имеющей степень равную единице, преобразуются в sc-дуги принадлежности, а те – которые имеют степень принадлежности равную нулю – в sc-дуги непринадлежности. Результат преобразования (рис. 8) есть текст, в котором обозначение принадлежности отмечено не как sc-дуга нечёткой принадлежности, а как sc-дуга принадлежности.

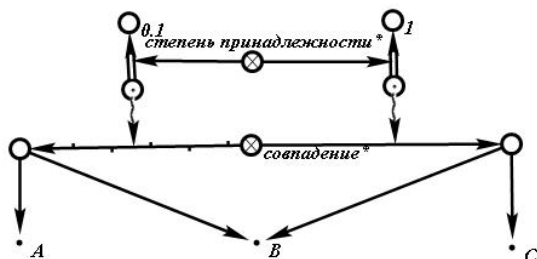


Рисунок 8– Пример перехода от нечёткого представления к чёткому.

В процессе рассуждений в интеллектуальной системе, нечёткая информация может уточняться. По мере того как происходят события исходная, оцененная в настоящий момент вероятность, сменяется условной по отношению к ней вероятностью уже в новый момент настоящего (рис. 9).

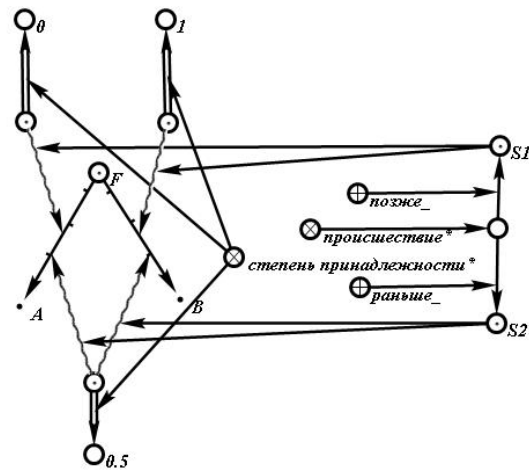


Рисунок 9– Пример представления изменяющегося во времени нечёткого множества.

В момент события, соответствующего узлу $S2$, степень принадлежности множеству F обоих sc-элементов A и B была оценена в 0,5, когда же наступило событие, соответствующее узлу $S1$, степень sc-элемента A стала равной нулю, а элемента B – единице.

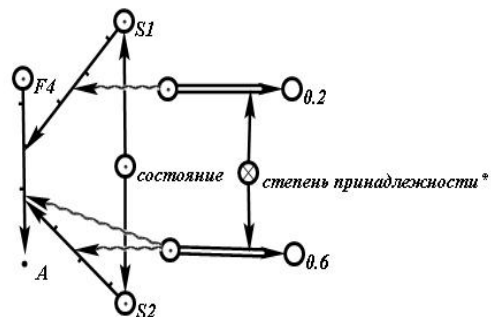


Рисунок 100– Пример представления множества с нечётко заданной ситуативностью.

Следует отметить, что сама информация о событиях, в которые рассматривается принадлежность или непринадлежность sc-элементов может быть нечёткой, как для нечёткого множества $F4$ (рис. 10). Элемент A принадлежит нестационарно множеству $F3$ (рис. 11), однако, то, что нечёткая оценка этой принадлежности соответствует событию, заданному узлом $S1$, равна 0,2, а – событию $S2$ – 0,6.

При интеграции знаний ситуативного (нестационарного) и гипотетического рода, каждое утверждение, которое содержится в исходных интегрируемых фрагментах, преобразуется по следующим схемам.

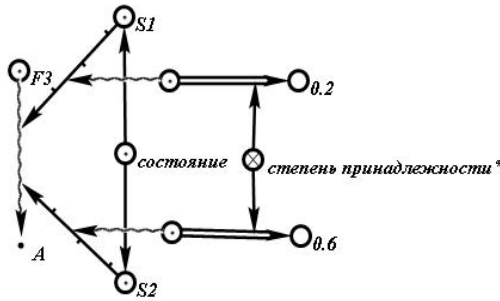


Рисунок 11– Пример представления нечёткого множества с нечётко заданной ситуативностью.

2. Интеграция знаний

Унифицированная модель баз знаний описывает базы знаний специального вида, в которых знания представлены с использованием унифицированной модели представления знаний [Ивашенко, 2009а]. Между базами знаний определены отношения содержательного и структурного включения одной базы знаний в другую. На множестве баз знаний заданы операции содержательной проекции и темпорализации [Ивашенко, 2009а]. Соответствие интеграции каждой паре баз знаний сопоставляет однозначное отображение множества всех знаков, формирующих эти базы знаний, на множество всех знаков результирующей базы знаний. Путём интеграции осуществляется переход от некоторой исходной базы знаний к требуемой оптимизированной базе знаний, которая имеет более высокое качество [Ивашенко, 2011а].

Решение задачи интеграции требуется в основных трёх случаях:

- интеграция разработанных и отлаженных баз знаний;
- интеграция отлаженных компонентов в базу знаний;
- добавление знаний при редактировании базы знаний.

В задаче интеграции двух баз знаний или их фрагментов в качестве исходных данных используются две БЗ (или фрагмента) и дополнительная метаинформация о свойствах знаков, принадлежащих этим базам знаний. Результатом решения этой задачи является нахождение интегрированной базы знаний, такой, что каждый знак из исходных интегрируемых баз знаний, имеет единственное представление в виде соответствующего знака в интегрированной базе [Ивашенко, 2009а]. Будем говорить, что осуществляется слияние двух знаков исходных БЗ тогда и только тогда, когда каждому из обоих соответствует в интегрированной базе знаний единственной знак. При интеграции в качестве дополнительной метаинформации может использоваться информация о внешних обозначениях (идентификаторах) понятий или информация, заданная базовой или ключевой рефлексивной семантикой знаков из этих БЗ. В

условиях неполноты информации можно выделить два типа стратегий слияния знаков интегрируемых баз знаний: безопасные (выполняемые однозначно и непротиворечиво) и небезопасные. Безопасная стратегия включается в любую стратегию слияния.

Задача интеграции баз знаний может быть разделена на две задачи – интеграция содержания баз знаний и интеграция структуры баз знаний (обеспечение целостности базы знаний).

Без потери общности для решения задачи интеграции содержания баз знаний рассмотрим задачу интеграции двух фрагментов баз знаний, каждый из которых представляет связную информационную конструкцию sc-языка.

При интеграции двух баз знаний между ними выявляется некоторое множество пар потенциально синонимичных элементов. Методика точного решения этой задачи включает как решение этой задачи вручную, путём получения соответствующего ответа от разработчика, так и автоматизированное решение на основе известной базовой теоретико-множественной семантики элементов и ключевых узлов sc-языка. Исходя только из базовой теоретико-множественной интерпретации, уже в некоторых фрагментах базы знаний можно легко установить факт наличия или отсутствия синонимии. Зная множество таких пар и множество всех возможных пар всех элементов в заданном фрагменте базы знаний, легко вычислить множество и число пар потенциально синонимичных элементов такой базы знаний.

Имея информацию о потенциально синонимичных элементах, можно оценить множество и количество всевозможных структурно различимых вариантов слияния потенциально синонимичных элементов в элементы результирующей базы знаний, являющееся подмножеством результатов соответствия интеграции. Это множество обозначим величиной $I(G)$, где G – симметричный ориентированный граф, множество рёбер $E(G)$ которого является множеством всех пар потенциально синонимичных элементов исходных баз знаний, а множество вершин $V(G)$ – множеством всех элементов исходных баз знаний

$$E(G) \subseteq (V(G))^2. \quad (13)$$

Для того, чтобы определить множество $I(G)$, используем специальную операцию

$$A \overset{\text{def}}{\cup} B = \bigcup_{(P,Q) \in A \times B} \{P \cup Q\}, \quad (14)$$

которая является ассоциативной и коммутативной.

Чтобы задать множество $I(G)$, введём семейство всевозможных множеств рёбер ориентированных графов $C(G)$ на множестве вершин графа G , каждая компонента связности которых является полным подграфом [Ивашенко, 2009b], тогда

$$I(G) = \left(\bigcup_{e_{ij} \in E(G)} \{\emptyset, \{e_{ij}\}\} \right) \cap C(G). \quad (15)$$

Число Q элементов множества $I(G)$ можно рассматривать в виде критерия качества: чем меньше это число, тем выше качество и наоборот. Однако, уже для небольших фрагментов баз знаний, это число может оказаться большим и трудным для расчёта, поэтому на практике более целесообразно использовать или логарифм этого числа или логарифм его оценок, или связанные с ними величины (число пар потенциальных синонимов).

$$Q = |I(G)|. \quad (16)$$

В работе [Ивашенко, 2009b] для числа Q установлены соотношения для верхней и нижней границы.

Интеграция онтологий и баз знаний осуществляется через слияние знаков. Рассмотрим m объектов области значений семантической интерпретации некоторого знака. Тогда, на множестве этих объектов, число неоднозначных и однозначных постоянных семантических интерпретаций (семантик) знака равно $2^{2^m} - 1$.

Чтобы осуществить слияние знаков (воплощений знака) необходимо, чтобы пересечение областей значения их семантик было непустым. В частности – теоретико-множественных семантик.

Если пересечение областей значения семантик знаков является пустым, то слияние таких знаков невозможно и такие знаки различны.

Тогда как для выявления различных знаков достаточно одного контр-примера, для выявления подлежащих слиянию знаков необходимо не только установить, что пересечение областей значения их семантик не является пустым, но и доказать, что эти знаки совпадают. Для этой цели используются различные утверждения о совпадении в виде аксиом и теорем.

Опишем свойства этого отношения совпадения. Если два знака совпадают, то не существует множества, которому одновременно принадлежит один из них и не принадлежит другой. Если два знака множеств совпадают, то множества равны. Выявить неравенство множеств просто: достаточно найти элемент, который принадлежит одному множеству и не принадлежит другому. Процедура выявления равенства множеств зависит от их свойств. Если множества имеют конечное, не очень большое число принадлежностей или непринадлежностей элементов, что характерно для понятий, то перечислив все принадлежности или все непринадлежности, число которых совпадает соответственно с его мощностью или количеством непринадлежностей, и убедившись, что среди них нет непринадлежностей элементов, принадлежащих другому множеству, можно заключить, что множества равны. Если же множества бесконечны или число принадлежностей, равно как и непринадлежностей у них, очень большое, то

равенство множеств можно доказать только через утверждения об их свойствах.

Прежде, чем проводить слияние знаков двух баз знаний, следует провести отображение баз знаний (по аналогии с отображением онтологий).

Для отображения в некоторых работах [Maltese et al., 2010] используются следующие отношения между онтологическими понятиями $\{\equiv, \supseteq, \sqsubseteq, \perp, \sqcap\}$, соответственно – совпадения, обобщения, частности, строгого исключения и строгого пересечения. Например, эти отношения в проекции на теоретико-множественные соответствуют отношениям равенства множеств ($=$), надмножества (\supseteq), подмножества (\sqsubseteq), пустого пересечения множеств ($S1 \cap S2 = \emptyset$) и непустого пересечения множеств ($S1 \cap S2 \neq \emptyset$). Однако, как можно показать, эти отношения трудно или невозможно установить, когда множества имеют большую или бесконечную мощность, или если исходить из того, что существуют равные, но несовпадающие множества.

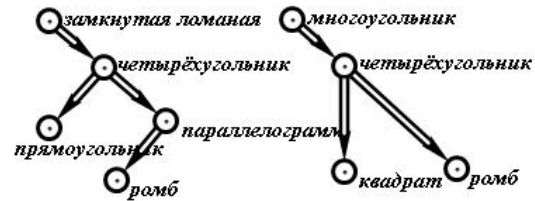


Рисунок 12– Пример, приводящий к противоречивому отображению

Для приведённого примера (рис. 12), при условии известности, что понятия ромба и квадрата не пересекаются, будет получено противоречивое «сильное» отображение (многоугольник (S1), многоугольник \wedge четырёхугольник (S2), многоугольник \wedge четырёхугольник \wedge ромб (S3), многоугольник \wedge четырёхугольник \wedge квадрат (S4), замкнутая ломаная (S5), замкнутая ломаная \wedge четырёхугольник (S6), замкнутая ломаная \wedge четырёхугольник \wedge прямоугольник (S7), замкнутая ломаная \wedge четырёхугольник (S8), замкнутая ломаная \wedge четырёхугольник \wedge параллелограмм \wedge ромб (S9)).

Таблица 1 – Противоречивое отображение

	S1	S2	S3	S4
S5	\equiv	\supseteq	\supseteq	\supseteq
S6	\sqsubseteq	\equiv	\supseteq	\supseteq
S7	\sqsubseteq	\sqsubseteq	\equiv	\equiv
S8	\sqsubseteq	\sqsubseteq	\equiv	\supseteq
S9	\sqsubseteq	\sqsubseteq	\equiv	\perp

Кроме этого предложенные [Maltese et al., 2010] методы отображения применимы к так называемым «легко-взвешенным» онтологиям [Giunchiglia et al., 2006] и не применимы к другим, аналогичные трудности встречаются и в других подходах [Aumueller et al., 2005], [Jean-Mary et al., 2007], [Nagy et al., 2010].

Исходя из этого, автором предложен следующий набор отношений {не уточнены*, связность*, различие*, исключение*, исключение пересекающихся* (исключаемое пересекающееся, исключяющее пересекающееся), симметричное исключение*, строгое пересечение*, совпадение*}; отношение потенциальной синонимии можно выразить, как $возможная\ синонимия^* = (не\ уточнены^* \cup связность^*) / (исключение^* \cup исключение\ пересекающихся^* \cup различие^*)$.

Следующее правило (рис. 13) позволяет в рамках унифицированной модели представления знаний единообразно, непротиворечиво и строго, что было одной из трудностей в [Maltese et al., 2010], задать правила отображения и слияния знаков баз знаний при использовании методов лексико-терминологического анализа (анализа идентификаторов).

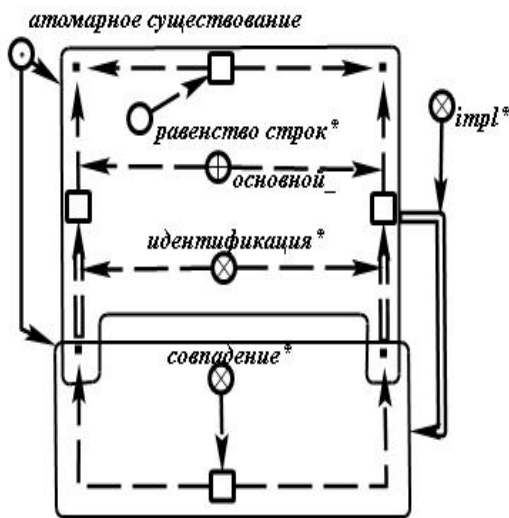


Рисунок 13 – Пример, приводящий к противоречивому отображению

Стратегии слияния при использовании методов лексико-терминологического анализа подробно рассмотрены в [Ивашенко, 2011b].

Следующие правила (рис. 14-19) описывают установление отображающих отношений на множестве знаков интегрируемых фрагментов баз знаний, после проведённых слияний в результате лексико-терминологического анализа.

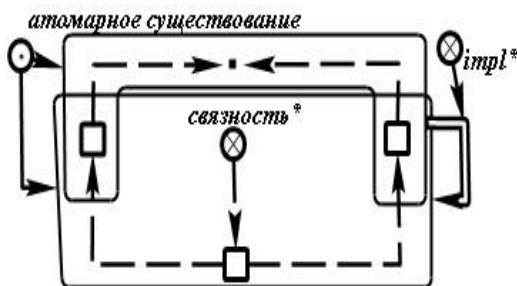


Рисунок 14 – Свойство отношения связности («два множества связны, если имеют общий элемент»)

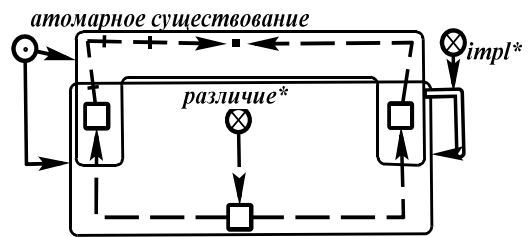


Рисунок 15 – Свойство отношения различия («два элемента различны, если один принадлежит множеству, а другой – нет»)

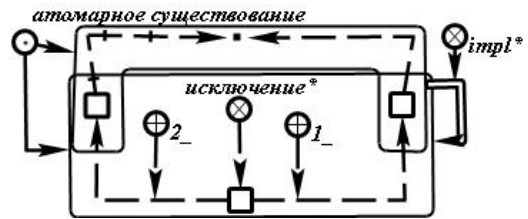


Рисунок 16 – Свойство отношения исключения (исключаемое, исключяющее)

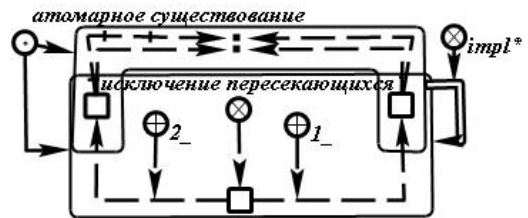


Рисунок 17 – Свойство отношения исключения пересекающихся

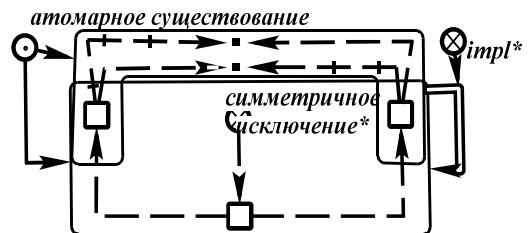


Рисунок 18 – Свойство отношения симметричного исключения

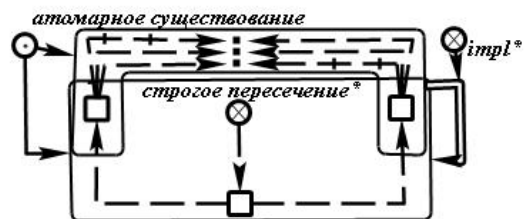


Рисунок 19 – Свойство отношения строгого пересечения

Приведём некоторые правила, описывающие взаимные свойства отношений (рис. 20-24).

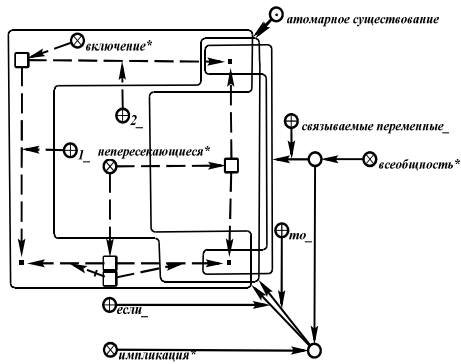


Рисунок 20 – Свойство отношения непересечения

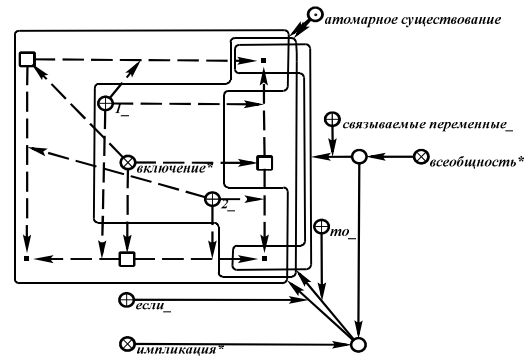


Рисунок 23 – Свойство транзитивности отношения включения

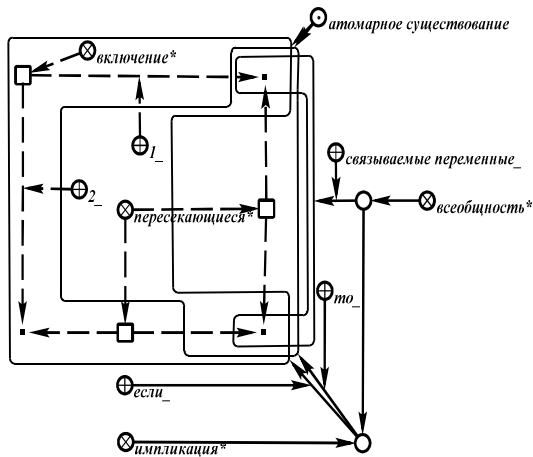


Рисунок 21 – Свойство отношения пересечения

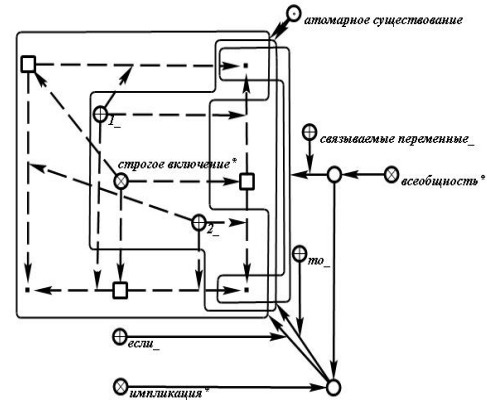


Рисунок 24 – Свойство транзитивности отношения строгого включения

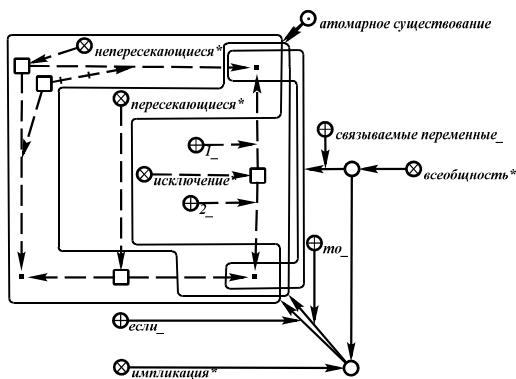


Рисунок 22 – Свойства отношения исключения

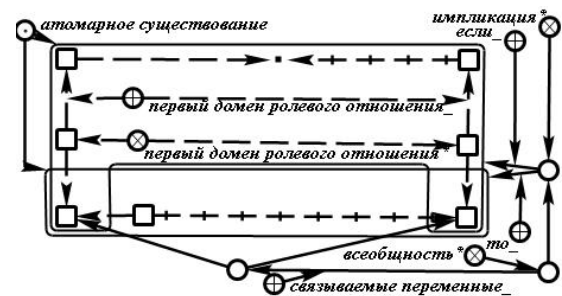
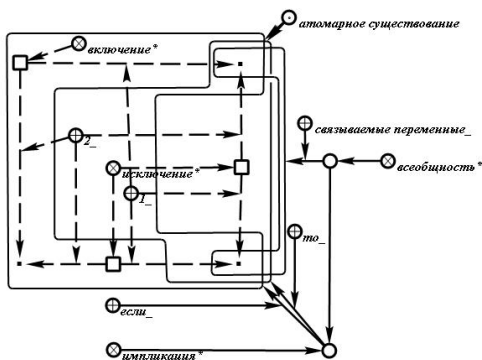


Рисунок 25 – Свойство ролевых отношений с различными первыми доменами

Для учёта теоретико-множественных отношений объединения и пересечения и (относительного) дополнения (как это предлагается в S-match) в данной задаче достаточно, например, добавить такие отношения как относительное дополнение (разность) и семейство непересекающихся множеств (двух и более) и задать их соответствующие свойства по отношению к ранее введённым отношениям.

При интеграции реляционных моделей можно придерживаться следующего порядка действий:

- интеграция ролевых отношений и их связок (рис. 25, 26),
- интеграция неролевых отношений (рис. 27, 28).

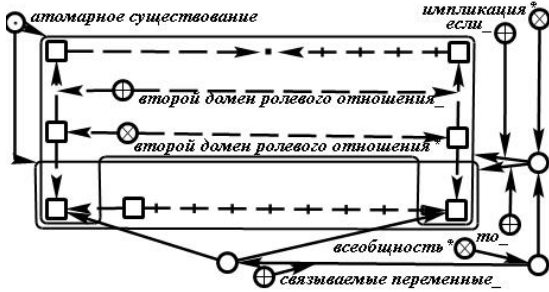


Рисунок 26 – Свойство ролевых отношений с различными вторыми доменами

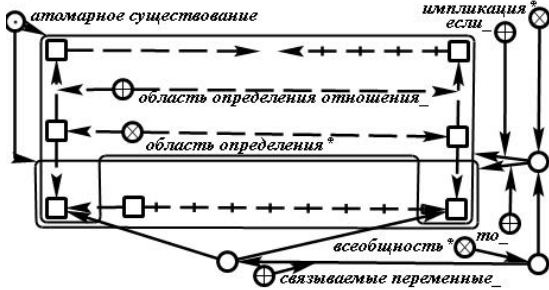


Рисунок 27 – Свойство отношений с различными областями определения

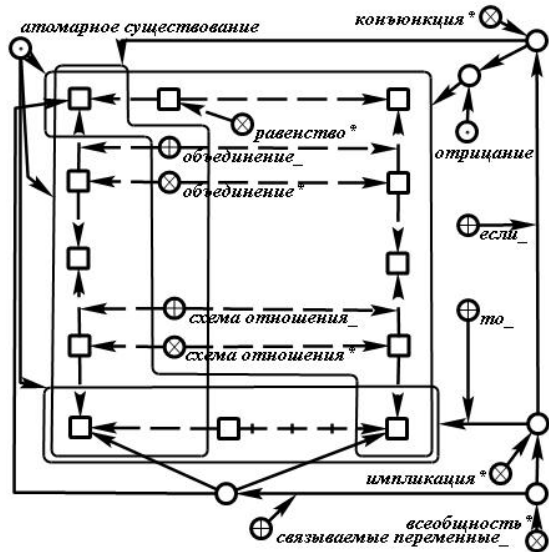


Рисунок 28 – Свойство отношения с различными объединениями схем

Важным признаком совпадения элементов является их принадлежность одной связке под однозначным ролевым отношением или наличие принадлежности им общего элемента под обратным однозначным ролевым отношением.

Все эти отношения можно однозначно и непротиворечиво установить в результате соответствующего базовой теоретико-множественной семантики структурного анализа, включая экстенциональный, за

полиномиальное время. Более глубокий логико-семантический анализ может потребовать больших временных затрат.

Для анализа на логико-семантическом уровне важно выделять утверждения, которые приводят к совпадению знаков интегрируемых фрагментов, в результате чего можно произвести их слияние. Такими высказываниями являются высказывания о свойствах логических высказываний о единственности, высказывания о свойствах множеств без кратных вхождений элементов (канторовских множеств), высказывания об отношениях без кратных связей. Ниже приведён пример (рис. 29), описывающий свойство логических утверждений о единственности и сводящий задачу анализа таких утверждений к задаче выявления связей отношений совпадения. Таким образом, при наличии средств логического вывода для решения задачи интеграции логико-семантическими методами достаточно описать свойства соответствующих ключевых узлов, выражающих количественные ограничения (например, единственность) или описывающих отсутствие кратных связей, через понятия совпадения sc-элементов (знаков текстов sc-языка).

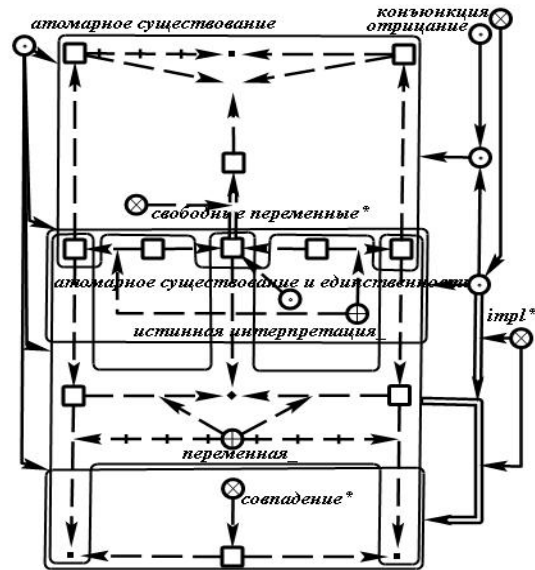


Рисунок 29 – Высказывание о слиянии значений переменных в утверждениях единственности

Для интеграции фрагмента базы знаний требуются следующие sc-операции, решающие следующие задачи:

- поиск и выборка элементов заданного интегрируемого фрагмента;
- проверка и выборка элементов, имеющих строковое содержимое и хранящих внешние обозначения (идентификаторы);
- проверка и выборка элементов, имеющих основной идентификатор;
- проверка и выборка элементов, имеющих неосновной идентификатор;

- добавление в словарь элемента с идентификатором;
- поиск и выборка элементов из словаря по идентификатору;
- формирование для заданных идентификаторов факта одинаковых идентификаторов;
- слияние совпадающих элементов;
- выявление и формирование фактов потенциально синонимичных элементов;
- выявление и формирование отображения знаков онтологий;
- формирование исходного варианта слияния элементов заданных разделов;
- применение к заданным элементам утверждений о совпадении элементов, в частности элементов, имеющих одинаковый основной идентификатор, утверждений о различии (несовпадении) и о возможном совпадении элементов;
- выбор и формирование варианта слияния заданных элементов для установленного факта возможного совпадения и заданного варианта слияния;
- выбор варианта слияния элементов базы знаний, исключение невыбранных вариантов слияния.

Обобщённый алгоритм интеграции фрагментов двух баз знаний приведён в [Ивашенко, 2011b]. Ниже (рис. 30) приведён алгоритм выявления пар потенциальных синонимов, использующий вышеописанные правила экстенционального метода.

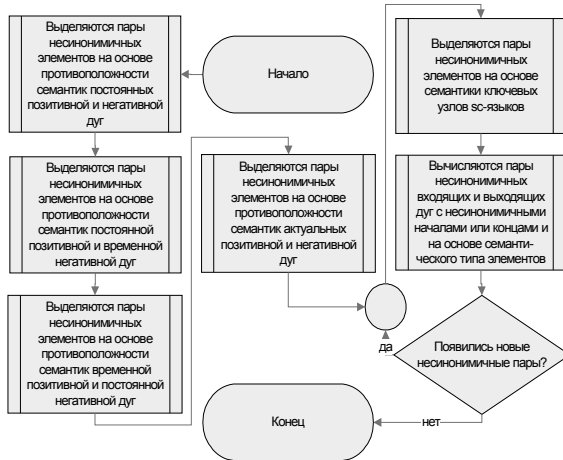


Рисунок 30 – Алгоритм вычисления множества пар различающихся (несинонимичных) элементов

Интеграция структуры баз знаний может быть проведена на исходном уровне или на метуровне. В первом случае, если в базе знаний после содержательной интеграции появились разделы, слияние которых не произошло, то формируется новый раздел базы знаний, который декомпозируется на основные (недекомпозируемые) разделы интегрируемых баз знаний. Этот вариант допустим, если декомпозиция разделов сохраняет древовидную структуру. Вторым вариантом предполагает формирование мета-разделов, которые описывают основные разделы исходных баз

знаний и их структуру и на которые декомпозируется основной раздел интегрированной базы знаний. Вторым вариантом интеграции структуры баз знаний может иметь смысл, когда содержание исходных баз знаний является слабо связным.

При интеграции тезаурусов необходимо построить связную иерархию определяемых понятий. Для каждого понятия необходимо указать является оно определяемым или нет. Для каждого понятия необходимо указать множества понятий, на основе которых оно может быть однозначно определено, а на основе каких – нет. Для построения такой иерархии применим алгоритм, подобный обобщённому алгоритму интеграции. При поиске выполнимых интерпретаций схем обход возможных предикативных выражений осуществляется по всевозможным предварённым конъюнктивным нормальным формам, атомарные подформулы которых соответствуют алфавитным меткам, связкам инцидентности и равенству значения переменной константе и задают всевозможные размеченные привязанные с помощью констант подграфы в графе онтологии. Для сокращения перебираемых вариантов используются эвристики.

Интеграция тезаурусов (баз знаний, содержащих определения) может быть осуществлена на основе отсеивания наборов понятий, которые не могут относиться как определяющие и определяемые, в результате анализа по следующим схемам (фильтрация неоднозначно определяющих множеств понятий для каждого понятия).

Схема n-значности ($|v| = n$) свойства α задана следующим выражением.

$$\lambda(\alpha, x, v, z) = (\alpha(\langle x, v, z \rangle) \wedge \forall w (\alpha(\langle x, w, z \rangle) \sim (w \in v))) \quad (17)$$

Схема необходимости.

$$\neg \exists U \exists V \exists Y (\neg \beta(\langle X, U, Y \rangle) \wedge \gamma(\langle Y, V, Z \rangle)) \quad (18)$$

Схема достаточности.

$$\neg \exists U \exists V \exists Y (\beta(\langle X, U, Y \rangle) \wedge \neg \gamma(\langle Y, V, Z \rangle)) \quad (19)$$

Схема прямой однозначности.

$$\exists U \exists V \exists Y (\lambda(\langle \beta, X, \{U\}, Y \rangle) \wedge \neg \lambda(\langle \gamma, Y, \{V\}, Z \rangle)) \quad (20)$$

Схема обратной однозначности.

$$\exists U \exists V \exists Y (\lambda(\langle \gamma, Y, \{V\}, Z \rangle) \wedge \neg \lambda(\langle \beta, X, \{U\}, Y \rangle)) \quad (21)$$

Здесь U, V, X, Y, Z являются множествами элементов: X – определяющий набор понятий; Z – определяемый набор понятий; U, V, Y – элементы окрестности X и Z ; α, β и γ – предикативные выражения определяющих и определяемых свойств.

Рассмотрим пример интеграции двух фрагментов баз знаний (см. рис. 31).

Для решения приведённой в примере задачи интеграции требуется следующая дополнительная информация:

- информация о равенстве основных идентификаторов;
- информация о равенстве неосновных идентификаторов;
- информация о том, какие множества являются множествами без кратных вхождений элементов (канторовские множества);
- информация об отсутствии кратных связей в геометрических отношениях;
- утверждение об однозначном задании биссектрисы заданного треугольника, проходящей через заданную вершину;
- утверждение об однозначном задании треугольника заданными тремя вершинами
- утверждение об однозначном задании вписанной окружности заданным треугольником;
- утверждение о совпадении центра вписанной в треугольник окружности и точки пересечения биссектрис треугольника.

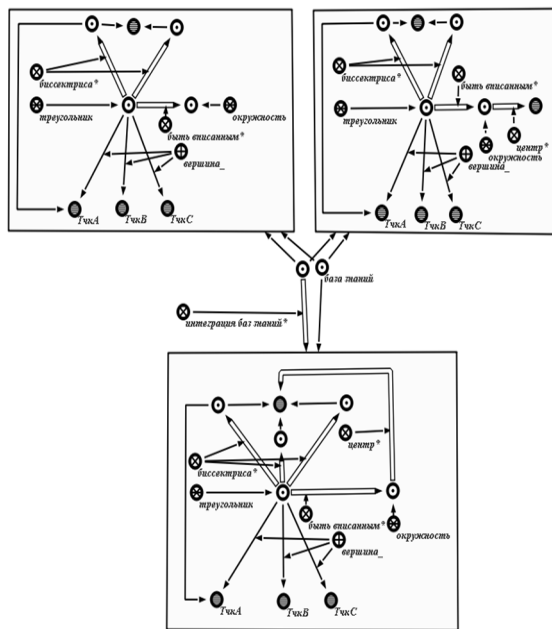


Рисунок 31 – Пример интеграции двух фрагментов баз знаний

ЗАКЛЮЧЕНИЕ

Рассмотренные модели и средства в рамках семантической технологии проектирования баз знаний [OSTIS, 2010] обеспечивают разработку баз знаний путём добавления и интеграции [Гулякина и др., 2004] в разрабатываемую базу знаний многократно используемых компонентов баз знаний, что позволяет сократить сроки её разработки.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Гаврилова и др., 2000] Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2000.
- [Голенков и др., 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков[и др.] – Мн. : БГУИР, 2001.
- [Гулякина и др., 2004] Гулякина Н.А. Ивашенко В.П. Интеграция знаний в информационных системах. / Н.А. Гулякина, В.П. Ивашенко // Доклады БГУИР. – 2004. – №6. – С. 113-119.
- [Ивашенко, 2003] Представление нейронных сетей и систем продукции в однородных семантических сетях. / В.П. Ивашенко // Известия Белорусской инженерной академии. – 2003. – №1(15)/1. – с.184-188.
- [Ивашенко, 2004] Применение однородных семантических сетей для представления знаний о нестационарных предметных областях. / В.П. Ивашенко // Известия Белорусской инженерной академии. – 2004. – №1(17)/3. – с.77-80.
- [Ивашенко, 2009a] Ивашенко В.П. Семантические модели баз знаний / В.П. Ивашенко Информационные системы и технологии (IST'2009): материалы V Междунар. конф.-форума в 2-х ч. Ч.2 – Минск: А.Н.Вараксин, 2009.- с.125-128.
- [Ивашенко, 2009b] Ивашенко В.П. Алгоритмы верификации и интеграции баз знаний. Вестник Брестского государственного технического университета, БрГТУ, 2009, №5.
- [Ивашенко, 2011a] Ивашенко В.П. Семантическая технология компонентного проектирования баз знаний. Материалы Международной науч.-техн. Конференции OSTIS,2011:Минск, Республика Беларусь, БГУИР 10-12 февраля 2011.
- [Ивашенко, 2011b] Ивашенко В.П. Алгоритмы операций отладки и интеграции баз знаний. Дистанционное обучение – образовательная среда XXI века: материалы VII Международной науч.-метод. конференции (Минск, 1-2 декабря 2011г.). – Минск: БГУИР, 2011, сс.227-229.
- [Ивашенко, 2012] Ивашенко В.П. Семантические модели интеграции и отладки баз знаний. Материалы Международной науч.-техн. Конференции OSTIS,2012:Минск, Республика Беларусь, БГУИР 16-18 февраля 2012.
- [Колб, 2012] Колб, Д. Г. Web-ориентированная реализация семантических моделей интеллектуальных систем / Д. Г. Колб // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012
- [Кофман, 1982] Кофман А. Введение в теорию нечетких множеств. М.: Радио и связь, 1982. — 432 с.
- [Кудрявцев, 2008] Кудрявцев Д. В. Практические методы отображения и интеграции онтологий. Семинар Знания и онтологии *Elsewhere*. КИИ-2008, Дубна, 2008.
- [Нариньяни, 2000] Нариньяни А. С. НЕ-факторы: неточность и недоопределенность — различие и взаимосвязь // Изв. РАН. Сер. Теория и системы управления. 2000. № 5. С. 44-56.
- [Тэрано, 1993] Прикладные нечеткие системы: Пер. с япон. / К. Асаи, Д. Ватада, С. Иваи и др.; Под ред. Т. Тэрано, К. Асаи, М. Сугэно. – М.: Мир, 1993.
- [Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В. Ф. Хорошевский // Искусственный интеллект и принятие решений. - 2008. - № 1. - С.80-97.
- [Allen, 1991] Allen, J.F.: Time and time again: the many ways to represent time. International Journal of Intelligent Systems 6 (1991) 341-355.
- [Allen, 1994] Allen, J. F. Ferguson, G. Actions and Events in Interval Temporal Logic. J. Log. Comput., 1994, 4, 531-579.
- [Aumuller et al., 2005] D. Aumuller, H. Do, S. Massmann, E. Rahm. 2005. Schema and ontology matching with COMA++. Proc. of the 2005 International Conference on Management of Data, pp. 906-908.
- [Bouquet, 2003] P. Bouquet, L.Serafini and S. Zanobini, M. Benerecetti. An algorithm for semantic coordination. Proc. of Semantic Integration Workshop, colocated with the 2nd Int.l Semantic Web Conference (ISWC2003), 20-23 October 2003, Sanibel Island, Florida, US.
- [DAML, 2006] DAML.org: The DARPA Agent Markup Language Homepage, Электронный ресурс. Режим доступа: <http://www.daml.org/>.
- [Doan and Halevy, 2005] A. H. Doan, A. Y. Halevy. Semantic integration research in the database community: A brief survey. AI magazine, 26(1), 2005

- [**CLIPS, 1991**] CLIPS Version 5.1 User's Guide, NASA Lyndon B. Johnson Space Center, Software Technology Branch, Houston, TX, 1991.
- [**CoFI:CASL-Summary, 2004**] CoFI Language Design Group. Casl summary. In Casl Reference Manual, LNCS 2960 (IFIP Series), part I. Springer, 2004.
- [**CycL, 2002**] The Syntax of CycL. Электронный ресурс. Режим доступа: <http://www.cyc.com/cycdoc/ref/cycl-syntax.html>.
- [**Ganter, 1999**] B. Ganter, R. Wille: Formal Concept Analysis – Mathematical Foundations. Springer, Heidelberg 1999.
- [**Gangemi et al., 1996**] A. Gangemi, G. Steve and F. Giacomelli, 1996. "ONIONS: An Ontological Methodology for Taxonomic Knowledge Integration", ECAI-96 Workshop on Ontological Engineering, Budapest, August 13th.
- [**Genesereth and Fikes, 1992**] Michael R. Genesereth and Richard E. Fikes. Knowledge interchange format version 3.0 reference manual. Logic Group First Draft January 1992 Report Logic-92-1 Current Version June 1992.
- [**Giunchiglia et al., 2006**] F. Giunchiglia, M. Marchese and I. Zaihayeu (2006). Encoding classifications into lightweight ontologies. University of Trento Technical Report # DIT-06-016, March 2006.
- [**Jean-Mary et al., 2007**] Jean-Mary Y., Kabuka, M. ASMOV: Ontology Alignment with Semantic Validation. Joint SWDB-ODDBIS Workshop, September 2007, Vienna, Austria, 15-20.
- [**IEEE, 2006**] IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, Automatic Fuzzy Ontology Generation for Semantic Web, VOL. 18, NO. 6, JUNE 2006.
- [**IDEF5, 1994**] Information Integration for Concurrent Engineering (ICE). IDEF5 Method Report. - Knowledge Based Systems, Inc. , 1408 University Drive East College Station, Texas, USA. September 21. 1994.
- [**ISO13250**] ISO/IEC 13250:2003 Topic maps – representation and interchange of knowledge, with an emphasis on the findability of information.
- [**ISO24707**] ISO/IEC 24707:2007 – Information technology – Common Logic (CL): a framework for a family of logic-based languages.
- [**Maldavan, 2001**] J. Madhavan, P. A. Bernstein, E. Rahm. Generic Schema Matching with Cupid. In. Proc. of the 27th Conference on Very Large Databases, 2001.
- [**Maltese et al., 2010**] V. Maltese, F. Giunchiglia, A. Autayeu: Save up to 99% of your time in mapping validation In Proceedings of ODBASE, 2010.
- [**Martin Ph., 2002**] Martin Ph. (2002). Knowledge representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English. Proceedings of ICCS 2002, 10th International Conference on Conceptual Structures (Springer, LNAI 2393, pp. 77-91), Borovets, Bulgaria, July 15-19, 2002.
- [**McGuinness et al., 2000**] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. The chimaera ontology environment. In Proceedings of AAAI, pages 1123-1124, 2000.
- [**Mitra, 2001**] Prasenjit Mitra, Gio Wiederhold and Stefan Decker: A Scalable Framework for Interoperation of Information Sources. 1st International Semantic Web Working Symposium (SWWS '01), Stanford University, Stanford, CA, July 29-Aug 1, 2001, Jul.
- [**Michael Kifer et al., 1995**] Michael Kifer, Georg Lausen, James Wu: Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of ACM 42(4): 741-843 May (1995).
- [**Mustafa Jarrar et al., 2008**] Mustafa Jarrar and Robert Meersman: "Ontology Engineering -The DOGMA Approach". Book Chapter (Chapter 3). In Advances in Web Semantics I. Volume LNCS 4891, Springer, 2008.
- [**Nagy et al., 2010**] Miklos Nagy and Maria Vargas-Vera. Towards an Automatic Semantic Data Integration: Multi-Agent Framework Approach. Sematic Web. Gang Wu (ed), Chapter 7, pp. 107-134; In-Tech Education and Publishing KG; 2010, ISBN 978-953-7619-54-1.
- [**Novak et al., 2008**] Joseph D. Novak & Alberto J. Canas, The Theory Underlying Concept Maps and How to Construct and Use Them. Florida Institute for Human and Machine Cognition. Pensacola Fl, 32502. Technical Report IHMC CmapTools 2006-01 Rev 2008-01.
- [**OSTIS, 2010**] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2010. - Режим доступа: <http://ostis.net>. – Дата доступа: 2.11.2010
- [**Smith et al., 2007**] Smith, B.; Ashburner, M.; Rosse, C.; Bard, J.; Bug, W.; Ceusters, W.; Goldberg, L. J.; Eilbeck, K. et al. (2007). "The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration". Nature Biotechnology 25 (11): 1251-1255. doi:10.1038/nbt1346.
- [**Sowa et al., 2008**] Sowa, J. Conceptual Graphs/ John F. Sowa, F. van Harmelen, V. Lifschitz, B. Porter// eds., Handbook of Knowledge Representation, Elsevier, 2008, pp. 213-237.
- [**Stumme, 2001**] Gerd Stumme and Alexander Maedche. FCA-merge: bottom-up merging of ontologies. In In Proceedings of 17th IJCAI, pages 225{230, Seattle (WA), USA, 2001.
- [**W3C:DAML+OIL, 2001**] DAML+OIL (March 2001) Reference Description: W3C Note 18 December 2001. Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein and Lucent Technologies, Inc, Электронный ресурс. Режим доступа: <http://www.w3.org/TR/daml+oil-reference>.
- [**W3C:RDFS, 2004**] RDF Vocabulary Description Language 1.0: RDF Schema: W3C Recommendation 10 February 2004 Электронный ресурс. / eds. Dan Brickley, R.V. Guha. Режим доступа: <http://www.w3.org/TR/rdf-schema/>, свободный.
- [**W3C:RIF, 2010**] RIF Overview: W3C Working Group Note 22 June 2010. Электронный ресурс. / eds. Michael Kifer, Harold Boley. Режим доступа: <http://www.w3.org/TR/2010/NOTE-rif-overview-20100622/>.
- [**W3C:OWL, 2004**] OWL Web Ontology Language. Overview: W3C Recommendation 10 February 2004 Электронный ресурс. / eds. Deborah L. McGuinness, Frank van Harmelen. Режим доступа: <http://www.w3.org/TR/owl-features/>.
- [**W3C:OWL2, 2009**] OWL 2 Web Ontology Language Document Overview: W3C Recommendation 27 October 2009. Электронный ресурс. / eds. W3C OWL Working Group. Режим доступа: <http://www.w3.org/TR/owl2-overview/>.
- [**W3C:SPARQL, 2008**] SPARQL Query Language for RDF: W3C Recommendation 15 January 2008. Электронный ресурс. / eds. Eric Prud'hommeaux, Andy Seaborne. Режим доступа: <http://www.w3.org/TR/rdf-sparql-query/>.
- [**XTM, 2001**] XML Topic Maps (XTM) 1.0: TopicMaps.Org Specification. Members of the TopicMaps.Org Authoring Group. Электронный ресурс. / eds. Steve Pepper, Graham Moore. Режим доступа: <http://www.topicmaps.org/xtm/>.
- [**Van Renssen, 2005**] Van Renssen, Andries (2005). Gellish: A Generic Extensible Ontological Language. Delft University Press. ISBN90-407-2597-4. <http://repository.tudelft.nl/view/ir/uuid%3Ade26132b-6f03-41b9-b882-c74b7e34a07d/>.

UNIFIED KNOWLEDGE REPRESENTATION AND KNOWLEDGE INTEGRATION

Ivashenko V.P.

*Belarusian State University of Informatics and
Radioelectronics, Minsk, Republic of Belarus*

ivashenko@bsuir.by

The work concerns models for unified knowledge representation and knowledge integration in the component knowledge base design technology (OSTIS). Several schemas and rules for algorithms of knowledge and ontology integration are considered. These rules and schemas provide ontology integration in conditions of dynamically updating knowledge and knowledge base incompleteness.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ТЕХНОЛОГИЯ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ НА ОСНОВЕ УНИФИЦИРОВАННЫХ СЕМАНТИЧЕСКИХ СЕТЕЙ

Давыденко И.Т.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

ir.davydenko@gmail.com

В работе рассматривается состав семантической технологии проектирования баз знаний интеллектуальных систем. Данная технология ориентирована на семантическое представление знаний, расширение контингента разработчиков баз знаний и сокращение сроков проектирования.

Ключевые слова: база знаний, интеллектуальная система, онтология, технология проектирования.

ВВЕДЕНИЕ

В настоящее время все более актуальной становится задача эффективного информационного обеспечения научной и производственной деятельности, связанная с бурным ростом объемов информации в различных отраслях знаний. Данная задача, как правило, рассматривается в контексте создания хранилищ знаний и их систематизации и структуризации с целью облегчения их обработки.

База знаний является одним из ключевых компонентов интеллектуальных систем различного назначения [Гаврилова и др., 2001], [Гаврилова, 2008], [Хорошевский, 2008]. Разработка этого компонента является трудоемким и продолжительным процессом.

При разработке баз знаний важно обеспечить не только возможность хранения знаний и навигации по ней, но и возможность работы над созданием и изменением базы знаний распределенным коллективом разработчиков.

На сегодняшний день существует ряд проблем, в области формирования баз знаний [Ефименко, 2011]:

- доступность семантического контента;
- доступность баз знаний и средств их разработки;
- эволюция баз знаний;
- масштабируемость баз знаний;
- мультязычность баз знаний;
- стабильность баз знаний;
- визуализация баз знаний.

Эти проблемы обусловлены следующими недостатками существующих технологий проектирования баз знаний:

- сложность языков представления знаний;
- неоднородность и ограниченность моделей представления знаний;
- ограниченность или отсутствие средств верификации и отладки баз знаний;
- плохая отчуждаемость баз знаний в силу привязки к инструментарию;
- ограниченность или отсутствие возможности интеграции баз знаний, что влечет за собой ограничения на расширение базы знаний, отсутствие общих стандартов совместимости уже разработанных фрагментов баз знаний [Ивашенко, 2012].

Все эти недостатки порождают еще одну проблему – недостаточное количество квалифицированных инженеров баз знаний, которая возникает из-за высоких требований к разработчикам.

В качестве решения вышеуказанных проблем предлагается технология компонентного проектирования баз знаний, основанная на унифицированных семантических сетях с базовой теоретико-множественной интерпретацией. Данная технология представляет собой комплекс моделей, инструментальных средств и методов проектирования баз знаний.

В основе предлагаемой технологии лежат следующие основные принципы массовой семантической технологии проектирования интеллектуальных систем OSTIS (Open Semantic Technology for Intelligent Systems) [OSTIS, 2012]:

- поэтапное эволюционное проектирование баз знаний на основе быстрого прототипирования;
- ориентация на коллективное проектирование баз знаний в рамках Open Source проекта;
- ориентация на семантическое представление знаний;
- унификация моделей баз знаний интеллектуальных систем;
- модульное проектирование на основе библиотек типовых многократно используемых компонентов.

Технология проектирования баз знаний представляет собой комплекс моделей, инструментальных средств и методов проектирования баз знаний.

Предлагаемая технология имеет следующую структуру:

- унифицированная семантическая модель представления знаний
- семантическая модель базы знаний;
- библиотека многократно используемых компонентов баз знаний;
- средства проектирования семантических моделей баз знаний;
- методика проектирования семантических моделей баз знаний.

С формальной точки зрения, технология проектирования материализуется в виде интеллектуальной метасистемы для поддержки проектирования баз знаний.

1. Унифицированная семантическая модель представления знаний

В качестве формальной основы проектируемых логико-семантических моделей баз знаний интеллектуальных систем используются графодинамические модели специального вида – семантические модели представления и обработки знаний, в основе которых лежат унифицированные семантические сети с базовой теоретико-множественной интерпретацией. Основным способом кодирования информации для таких сетей является SC-код (Semantic Code) [Голенков, 2012].

Все элементы (атомарные фрагменты) семантической сети являются знаками различных сущностей. Такими сущностями могут быть всевозможные внешние описываемые объекты, а также различные множества, состоящие их элементов (атомарных фрагментов) этой же семантической сети.

База знаний интеллектуальной системы, представленная в виде корректно построенной семантической сети, полностью исключает дублирование информации в рамках такой базы знаний.

Представление знаний в виде унифицированных семантических сетей позволяет существенно упростить процедуру ассоциативного доступа к различным видам фрагментов хранимой базы

знаний, а также существенно расширить типологию запросов к базе знаний [Голенков, 2012].

2. Семантическая модель базы знаний

В основе разработки баз знаний с помощью технологии OSTIS лежит четкое разделение процесса проектирования формального описания семантической модели разрабатываемой базы знаний от процесса реализации (интерпретации) этой модели на той или иной платформе [Голенков, 2012]. Данный факт позволяет обеспечить кросс-платформенную разработку интеллектуальных систем.

Всю семантическую сеть (максимальную семантическую сеть), хранимую в семантической памяти абстрактной логико-семантической модели интеллектуальной системы, будем называть **абстрактной семантической моделью базы знаний** этой интеллектуальной системы.

Семантическая модель базы знаний интеллектуальной системы представляет собой формальную трактовку семантического пространства, которое известно интеллектуальной системе в текущий момент времени.

Эффективность интеллектуальной системы в первую очередь определяется объемом и качеством содержащихся в них формализованных экспертных знаний, как декларативных (теоретических), так и процедурных (практических навыков).

База знаний должна содержать в себе всю информацию, необходимую агентам, работающим над семантической памятью, для организации коллективной деятельности по решению задач, с которыми должна справляться интеллектуальная система.

Для расширения разнообразия видов знаний, хранимых в базе, необходимым этапом в разработке семантической модели базы знаний является ее структуризация.

Понятие базы знаний тесно связано с понятием предметной области. Соотношение между базой знаний и описываемой ею предметной областью задает семантику базы знаний интеллектуальной системы.

Рассмотрение структуры базы знаний во взаимосвязи с предметной областью позволяет рассматривать исследуемые объекты на разных уровнях детализации. Детализацию рассмотрения исследуемых объектов можно осуществлять как в рамках исходной предметной области, так и в системе самостоятельных, но связанных между собой предметных областей.

При переходе от предметной области к ее модели, представленной в виде семантической сети, выполняются следующие условия:

- каждому элементу предметной области взаимно однозначно соответствует обозначающий его элемент семантической сети;

- каждому сигнатурному элементу предметной области взаимно однозначно соответствует либо обозначающий его ключевой узел семантической сети, либо обозначающий элемент алфавита семантической сети.

Первым и важнейшим этапом проектирования семантической модели базы знаний является уточнение структуры описываемой предметной области или нескольких взаимосвязанных предметных областей. Под уточнением структуры предметной области понимается явное выделение *класса исследуемых объектов, класса вторичных объектов*, построенных на основе исследуемых, *класса вспомогательных объектов*, через связи с которыми описываются некоторые характеристики исследуемых объектов, *отношения, связи которых связывают только исследуемые объекты между собой*, а также *отношения, связи которых связывают исследуемые объекты со вспомогательными*.

Рассмотрение базы знаний с позиции ее соотношения с предметной областью позволяет рассматривать исследуемые объекты на различных уровнях детализации:

- классификация класса исследуемых объектов по различным признакам;
- классификация самих исследуемых объектов, т.е. рассмотрение структур взаимосвязанных частей этих объектов;
- рассмотрение связей исследуемых объектов со вспомогательными объектами, не входящими в класс исследуемых объектов.

В зависимости от исследуемых объектов можно говорить о достаточно богатой типологии предметных областей. Можно выделить следующие классы предметных областей:

- *предметная область, описывающая теоретико-множественные характеристики и связи заданного семейства объектов*. Такие предметные области, в частности, могут быть онтологиями других предметных областей;
- *терминологическая онтология* – это класс предметных областей, для каждой из которых объектами исследования являются термины (словосочетания), соответствующие различным элементам описываемой предметной области;
- *предметная область, являющаяся логическим описанием некоторой предметной области* – это класс предметных областей, для каждой из которых объектами исследования являются всевозможные логические формулы, описывающие причинно-следственные закономерности, интерпретируемые на некоторой описываемой предметной области;
- *логическая система понятий, описываемых в заданной формальной теории*. Эта предметная метаобласть выделяет класс понятий, не определяемых в заданной формальной теории, и связывает каждое определяемое понятие с теми понятиями, на основе которых оно определяется;

- *логическая система утверждений заданной формальной теории*. Эта предметная метаобласть выделяет класс аксиом для заданной формальной теории, каждой теореме ставит в соответствие одно из её доказательств (основное доказательство) и связывает каждую теорему со всеми теми утверждениями и определениями, которые используются в основном доказательстве этой теоремы;

- *предметная область вопросов и информационных задач* – это класс предметных областей, для каждой из которых объектами исследования являются вопросы, информационные задачи, обобщенные вопросы и обобщенные информационные задачи, задаваемые по отношению к некоторой описываемой предметной области, а также соответствующие им способы решения информационных задач (то есть различные программы).

Семантическая структура базы знаний интеллектуальной системы трактуется в рамках технологии проектирования баз знаний интеллектуальных систем *как иерархическая система взаимосвязанных между собой предметных областей*, которые представляются в базе знаний.

На множестве предметных областей могут быть заданы следующие отношения: включение, объединение, пересечение, декомпозиция, гомоморфизм, изоморфизм, теоретико-множественная онтология, логическое описание, логическая онтология. Исходя из этого, мы можем рассматривать некую *метаобласть*, объектами исследования которой являются всевозможные предметные области.

Таким образом, семантическая структура базы знаний представляет собой иерархическую систему описываемых ею предметных областей, надстраиваемых над заданной основной предметной областью.

Построение семантической структуры базы знаний требует не только явного представления спецификации каждой описываемой предметной области в виде формального текста, но и явного описания всевозможных связей между этими предметными областями.

3. Библиотека многократно используемых компонентов баз знаний

В целях сокращения времени процесса проектирования семантических моделей баз знаний интеллектуальных систем необходимо создать библиотеку многократно используемых семантически совместимых компонентов баз знаний. На основе этой библиотеки разработана методика компонентного проектирования баз знаний.

К основным типам компонентов баз знаний, хранящихся в библиотеке относятся:

- онтологии различных предметных областей, которые могут быть самыми различными по содержанию, однако должны быть семантически совместимыми;

- базовые фрагменты теорий, соответствующие различным уровням знания пользователя, начиная от базового школьного до профессионального;

- семантические окрестности различных объектов;

- спецификации формальных языков описания различных предметных областей.

Для обеспечения семантической совместимости таких компонентов баз знаний, которые являются унифицированными семантическими моделями, необходимо

- согласовать семантику всех используемых ключевых узлов;

- согласовать глобальные идентификаторы ключевых узлов, используемых в разных компонентах. После этого интеграция всех компонентов, входящих в состав библиотеки, и в любых комбинациях осуществляется автоматически, без вмешательства разработчика.

Для включения компонента в библиотеку необходимо его специфицировать по следующим критериям:

- предметная область, описание которой содержится в компоненте;

- класс (тип) компонента базы знаний;

- состав базы знаний;

- количественные характеристики ключевых узлов базы знаний;

- информация о разработчиках базы знаний;

- дата создания базы знаний;

- информация о верификации базы знаний;

- версия компонента базы знаний;

- условия распространения компонента базы знаний;

- сопровождающая информация.

4. Средства проектирования семантических моделей баз знаний

При коллективном инжиниринге знаний любая база знаний рассматривается как результат консенсуса группы специалистов о модели некоторой области знаний, что должно быть обеспечено соответствующими средствами проектирования [Ефименко, 2011].

Для эффективной организации проектирования баз знаний интеллектуальных справочных систем необходимо включать в состав основной системы в качестве подсистем следующие компоненты:

- интеллектуальную подсистему поддержки проектирования и сопровождения системы;

- help-систему информационного обслуживания разработчиков баз знаний;

- интеллектуальную систему автоматизации проектирования;

- интеллектуальную подсистему управления проектированием и сопровождением системы.

К средствам проектирования также относятся:

- средства верификации баз знаний, включающие пополняемую библиотеку команд и операций верификации баз знаний;

- средства анализа качества баз знаний, позволяющие определить такие характеристики баз знаний как полнота, связность, информативность;

- средства редактирования баз знаний, решающие проблему синхронизации редактирования семантической модели базы знаний и соответствующего фрагмента ее исходного текста несколькими разработчиками;

- средства поддержки коллективной разработки баз знаний.

5. Методика проектирования семантических моделей баз знаний

В основе комплексной методики проектирования семантических моделей интеллектуальных систем лежат два аспекта: эволюционное проектирование и коллективное проектирование интеллектуальных справочных систем.

В рамках семантической технологии проектирования баз знаний выделяют три основных метода разработки баз знаний.

1. Тест-ориентированный метод разработки баз знаний. Данный метод основывается на реализации стартового диалога с конечным пользователем системы. На первом этапе проектирования базы знаний исследуются запросы пользователей к проектируемой базе знаний и составляется тестовый сборник вопросов, охватывающий вопросы пользователей по рассматриваемой предметной области, что предполагает выделение семантически полного набора вопросов, ответы на которые должны содержаться в первой версии базы знаний. На следующем этапе все вопросы систематизируются и объединяются в классы. Далее на все вопросы, входящие в тестовый сборник записываются ответы, затем они переводятся на формальный язык. В процессе формализации ответов на вопросы выделяются ключевые узлы описываемой предметной области. Таким образом формируется первая версия базы знаний. Затем сборник вопросов расширяется и база знаний развивается эволюционно.

Такой метод проектирования баз знаний достаточно удобен при разработке баз знаний систем справочного назначения. К достоинствам данного подхода можно отнести эволюционное развитие базы знаний, которое позволяет быстро получить первый прототип.

2. Метод поэтапной формализации, основанный на семантической структуризации исходных документов. Данный подход к разработке баз знаний

основывается на использовании в качестве источника информации для формализации некоторого документа (книги, справочника, учебника, энциклопедии и т.п.).

В качестве подходов к декомпозиции процесса проектирования можно выделить следующие:

- выделение ветвей проектирования, соответствующих содержательной декомпозиции материала исходного документа (разделение на главы, параграфы, части и т.п.) вплоть до атомарных разделов. В данном случае каждый атомарный раздел документа соответствует атомарному разделу проектируемой базы знаний.
- проектирование семантической модели базы знаний заключается в детальной структуризации базы знаний, т.е. рассматривать структуру базы знаний, как иерархическую систему взаимосвязанных друг с другом предметных областей, представляемых в базе знаний. При таком рассмотрении процесса проектирования модели базы знаний можно выделить отдельные направления работ, соответствующие различным видам структуризации базы знаний, таких как построение теоретико-множественной онтологии описываемой предметной области, построение логической онтологии предметной области, построение онтологии системы высказываний предметной области, построение терминологической онтологии предметной области, и т.д. (направления структуризации знаний, хранящихся в базе знаний рассматривались выше).

3. Метод компонентного проектирования баз знаний, основанный на модели глобального семантического пространства человеческих знаний.

Суть данного подхода заключается в использовании библиотеки многократно используемых компонентов баз знаний, пополняемой авторами баз знаний. Основной интеграции для всех компонентов служит унифицированная модель представления знаний с использованием SC-кода [Голенков, 2011]. Достоинством данного подхода является максимальное сокращение сроков разработки базы знаний. Фактически процесс разработки базы знаний сводится к выбору инженером необходимых компонентов баз знаний из библиотеки, а затем их интеграции в единое семантическое пространство (принципы и подходы к интеграции знаний рассмотрены в работе [Ивашенко, 2013]). Недостающие фрагменты базы знаний разрабатываются одним из приведенных выше методов.

ЗАКЛЮЧЕНИЕ

В данной работе рассмотрена технология проектирования баз знаний интеллектуальных систем, основанная на семантических сетях. К особенностям приведенной технологии можно отнести:

- разнообразие видов хранимых знаний;

- наличие унифицированных средств структуризации базы знаний;
- наличие средств навигации по базе знаний;
- обеспечение коллективного проектирования баз знаний соответствующими средствами;
- обеспечение компонентной разработки баз знаний;
- наличие средств анализа качества баз знаний;
- мощные средства редактирования баз знаний.

Результаты, приведенные в работе, апробируются в рамках открытого проекта OSTIS [OSTIS, 2012].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Башмаков, 2003] Башмаков, А. И. Разработка компьютерных учебников и обучающих систем. / Башмаков А. И., Башмаков И. А., М.: Информационно-издательский дом "Филинь", 2003. — 616 с.
- [Башмаков, 2004] Башмаков, А.И. Интеллектуализация как средство повышения доступности технологий разработки компьютерных средств обучения / А.И. Башмаков // Образовательная среда сегодня и завтра: материалы Всероссийской научно-практической конференции [Электронный ресурс]. — 2004. - Режим доступа: <http://www.ict.edu.ru/vconf/index.php>. — Дата доступа: 10.10.2011
- [Бениаминов, 2008] Бениаминов, Е.М. О построении Web-сервера в стиле Semantic Wiki с открытым контекстным языком представления и запросов/Е. М. Бениаминов// КИИ-2008. Труды конференции. Т 2, С. 15-21
- [Голенков, 2011] Голенков, В.В., Гулякина Н.А. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем. — В кн Междунар. научн.-техн. конф. . «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2011). Материалы конф. [Минск, 10-12 февр. 2011 г.]. — Минск: БГУИР, 2011, с. 21-59.
- [Голенков, 2012] Голенков, В.В., Гулякина Н.А. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования. — В кн Междунар. научн.-техн. конф. . «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2012). Материалы конф. [Минск, 16-18 февр. 2012 г.]. — Минск: БГУИР, 2012.
- [Гаврилова и др., 2001] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. Учебник / Гаврилова Т.А. [и др.]; — СПб.: Изд-во «Питер», 2001.
- [Гаврилова, 2008] Гаврилова, Т. А. Визуальные методы работы со знаниями: попытка обзора / Т. А. Гаврилова, Н. А. Гулякина // Искусственный интеллект и принятие решений, 2008, № 1, С. 15-21
- [Грибова, 2011] Грибова, В. В. Облачная платформа для разработки и управления интеллектуальными системами / В. В. Грибова, А. С. Клещев, Д. А. Крылов, Ф. М. Москаленко, С. В. Смагин, В. А. Тимченко, М. Б. Тютюнник, Е. А. Шалфеева //Материалы международной научно-технической конференции OSTIS-2011, 5-14 стр. Минск БГУИР
- [Давыденко, 2012] Давыденко И. Т. Комплексная методика проектирования семантических моделей интеллектуальных справочных систем Интеллектуальная справочная система по геометрии / И. Т. Давыденко //Материалы международной научно-технической конференции OSTIS-2012, Минск БГУИР
- [Ефименко, 2011] Ефименко, И. В., Хорошевский, В. Ф. Онтологическое моделирование экономики предприятий и отраслей современной России: Часть 1. Онтологическое моделирование: подходы, модели, методы, средства, решения : препринт WP7/2011/08 (ч. 1) / И. В. Ефименко, В. Ф. Хорошевский ; Нац. исслед. ун-т «Высшая школа экономики». — М. : Изд. дом Высшей школы экономики, 2011. — 76 с.

[Загорулько и др., 2008] Загорулько, Ю.А., Боровикова, О.И. Подход к построению порталов научных знаний / Ю.А. Загорулько [и др.] // Автометрия. – 2008 – № 1, Т. 44, – С. 100–110.

[Клещев и др., 2001а] Клещев А.С., Артемьева И.Л. Математические модели онтологий предметных областей. Часть 1. Существующие подходы к определению понятия "онтология" // НТИ. Серия 2 "Информационные процессы и системы", 2001, № 2

[Клещев и др., 2001б] Клещев А.С. Артемьева И.Л. Математические модели онтологий предметных областей. Часть 2. Компоненты модели // НТИ. Серия 2 "Информационные процессы и системы", 2001, № 3

[Клещев и др., 2001с] Клещев А.С. Артемьева И.Л. Математические модели онтологий предметных областей. Часть 3. Сравнение разных классов моделей онтологий // НТИ. Серия 2 "Информационные процессы и системы", 2001, № 4

[Ивашенко, 2012] Ивашенко, В. П. Семантические модели и средства интеграции и отладки баз знаний / В. П. Ивашенко // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012. – С. 193-204

[Ивашенко, 2013] Ивашенко В.П. Унифицированное представление и интеграция знаний/ В. П. Ивашенко // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2013.

[Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В. Ф. Хорошевский // Искусственный интеллект и принятие решений. - 2008. - № 1. - С.80-97.

[ЭЗОП, 2007] Web-сервер онтологий системы ЭЗОП [Электронный ресурс]. – Режим доступа: <http://ezop-project.ru/drupal5/>. – Дата доступа: 14.11.2012.

[OSTIS, 2012] Проект OSTIS [Электронный ресурс]. Минск, 2012. – Режим доступа: <http://ostis.net/>. – Дата доступа: 11.12.2012.

COMPONENT TECHNOLOGY OF DESIGN OF KNOWLEDGE BASES BASED ON UNIFIED SEMANTIC NETWORKS

Davydenko I.T.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

ir.davydenko@gmail.com

INTRODUCTION

This paper considers the composition of semantic technology knowledge base design of intelligent systems. This technology is focused on the semantic representation of knowledge, the expansion contingent development of knowledge bases and reducing the time of design.

MAIN PART

The technology of design knowledge bases is a set of models, tools, and methods for knowledge bases designing.

The proposed technology has the following structure: unified semantic model of knowledge representation; semantic model of the knowledge base; library of reusable components of knowledge bases; design tools of semantic models of knowledge bases; the methods of designing semantic models of knowledge bases.

As a formal framework designed logical and semantic models of knowledge bases of intelligent systems using a semantic model of representation and processing of knowledge, which are based on semantic networks with a basic set-theoretic interpretation.

The unified semantic model of knowledge representation is in fact a formal way of describing the semantics of different kinds of knowledge. Semantic model of a knowledge base of intelligent system is a formal interpretation of the semantic space, which is known for the intelligent system at the current time.

To enhance the diversity of the knowledge stored in the knowledge base, a necessary step in the development of the semantic model knowledge base is its structuring.

The semantic structure of a knowledge base of intelligent system is treated in the technology knowledge base design of intelligent systems as a hierarchy of inter-related subject areas, which are submitted to the knowledge base.

In order, to reduce the time of the design process of semantic models of knowledge bases of intelligent systems, it's needed to create a library of reusable components is semantically compatible knowledge bases.

Designing tools are also include: tool of verification of knowledge bases, including the growing library of commands and operations verification of knowledge bases; tools for analysis of quality of knowledge bases for determining properties such as completeness of knowledge bases, connectivity, information; editing knowledge bases, the crucial problem of synchronization editing semantic model knowledge base and the corresponding fragment of the source code by multiple developers; tools to support the collaborative development of knowledge bases.

As a part of the semantic technology design knowledge bases there are three main methods of development of knowledge bases:

1. Test-driven development method knowledge bases.
2. The method of gradual formalization, based on the semantic structuring of the source documents.
3. The method of the component database design knowledge based on the idea of a global semantic space of human knowledge.

CONCLUSION

In the given paper short description of component technology of design of knowledge bases based on unified semantic are present. The article describes the main parts of this technology: the model of representation of knowledge, the model of knowledge base, tools and methods of designing of knowledge bases based on the unified semantic networks.

Given results are tested in an open project OSTIS [OSTIS, 2012].



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 618.3:50

ПАКЕТЫ ПРОГРАММ КАК ФОРМАЛЬНЫЕ ОНТОЛОГИИ: ПОСТРОЕНИЕ И ИСПОЛЬЗОВАНИЕ

Смирнов С.В.

*Институт проблем управления сложными системами Российской академии наук,
г. Самара, Россия*

smirnov@iccs.ru

В статье излагается онтологический подход к построению и использованию пакетов прикладных программ. Обоснованием подхода служит признание общности и определенности процессов отражения и моделирования, сводимое к построению и целенаправленной трансформации семантически гетерогенных объектных пространств, описывающих моделируемую действительность. Рассматриваемая технология имеет все признаки компонентного проектирования баз знаний, пакетов программ и приложений, ориентирована на интеграцию знаний и характеризуется высокой степенью унификации.

Ключевые слова: пакет прикладных программ; формальная онтология; интеграция знаний.

ВВЕДЕНИЕ

Всемирно известный российский специалист в области информатики академик А.П. Ершов неоднократно формулировал мысль о том, что качественный пакет прикладных программ (ППП) – это, прежде всего, методология решения прикладных задач [Ершов и др., 1982]. Это означает, что ценность ППП состоит, во-первых, в предоставлении пользователю развитой концепции для описания ситуаций в некоторой предметной области (ПрО), и, во-вторых, в наборе методов корректной трансформации ситуаций в нужном направлении, интерпретируемой как решение задач.

Блестящей иллюстрацией такой концепции являются библиотеки классов объектно-ориентированных систем автоматизации программирования, начиная с хрестоматийного класса *simset* Симулы [Андрианов и др., 1985]. Идею первичности представления знаний о ПрО (вернее, одновременно о нескольких ПрО, составляющих различные проблемные аспекты) при создании программных систем последовательно реализуют и все современные CASE-средства.

В этом контексте ППП видится как система, основанная на знаниях, фундаментальная функция которой – формирование для специалиста предметно-ориентированной среды моделирования (ПОСМ) с арсеналом методов конструирования и преобразования образов ПрО. И в этом смысле представление о ППП сближается со сложившимся пониманием формальной онтологии [Guarino, 1995].

В данной статье анализируются различные аспекты проектирования ППП и использующих их приложений на основе формальных онтологий.

1. Объектная модель предметной области: онтология и ее «денотат»

Несмотря на различные варианты толкования онтологии, когда обычно ей отводится место посредника между тезаурусом и той или иной формальной моделью ПрО, практический опыт дает основание считать, что онтология и есть модель ПрО [Нариньяни, 2001]. Но уже на философском уровне известный тезис об онтологической относительности [Куайн, 1996] ставит под сомнение саму возможность построения формальной онтологической спецификации. Эффективное решение этой проблемы может быть найдено в прагматической интерпретации формальной онтологии как объектно-ориентированной модели знаний о ПрО [Смирнов, 2008].

Обоснование этого подхода состоит в том, что формулировке знаний, составляющих содержание научных теорий и инженерных дисциплин, фундаментальную роль играют два когнитивных суждения о мире: возможность различения дискретных объектов и существование отношений между ними. Множество объектов, рассматриваемых в контексте всякой задачи, образует ее ПрО. На этом множестве задаются отношения: унарные интерпретируются как имманентные агрегируемые свойства объектов (функции объектов); произвольной арности, или

собственно отношения, описывают различные ассоциации объектов. Мощност моделирования, присущая бинарным отношениям, позволяет описывать любые ассоциативные отношения в ПрО с помощью пиринговых связей между объектами. Картина ПрО в виде сети связанных объектов в теории представления знаний известна как семантическая сеть, наглядным изображением которой служит раскрашенный оргграф. Вершины сети, представляющие объекты ПрО, распадаются на две категории: экземпляры и классы, - которые находятся в отношении экземплификации. Подсеть, состоящая из вершин-классов, соединяющих эти вершины дуг сорта «является видом», а также вершин-свойств, связанных с вершинами-классами дугами «является частью», описывает понятийную структуру ПрО, т.е. задает онтологию ПрО. Подсеть вершин-экземпляров являет собой «денотат» онтологии - денотативную объектную модель ПрО, в которой типы вершин и дуг определяются онтологией ПрО.

При решении задач онтология представляет теорию, а денотативная модель конкретизирует эту теорию применительно к актуальной ситуации в моделируемой ПрО. Множественность таких ситуаций делает целесообразным:

- иметь отдельные спецификации для онтологической и денотативной компонент объектной модели ПрО;
- реализовывать экземплификацию «внемодельными», технологическими методами;
- использовать при решении задач понятие контекста моделирования (КМ) как актуальной пары (онтология ПрО, денотативная модель ПрО).

Онтологические модели предметных областей - онтологии и денотативные модели - однородны в том смысле, что каждая является денотативной объектной моделью своей особенной ПрО. Для денотативной модели это ПрО, описываемая онтологией этой ПрО. Для онтологий это ПрО, которой принадлежат объемы понятий «класс объектов» и «свойство», а онтология, где эти понятия описаны (т.е. «онтология онтологий», или метаонтология), должна быть семантически замкнута, описывая саму себя. Это и есть «технологическая реакция» на онтологическую относительность описания реального мира.

2. Построение и использование ППП

Мы отождествляем создание и эксплуатацию ППП с формированием ПОСМ, представление о которой является естественным развитием концепции построения и использования предметно-ориентированных пакетов программ и систем моделирования, обобщая эволюцию известных схем автоматизации программирования [Поспелов, 1988].

Состав и назначение элементов ПОСМ, а также порядок их порождения и использования определяется единством и логической определенностью осуществляемого любой

отраслью науки познавательного процесса. Это означает, что процесс решения задачи с помощью компьютера включает ряд узловых этапов, определяющих совместно с продуктами их выполнения общую схему обработки информации при создании и использовании ППП (рисунок 1).

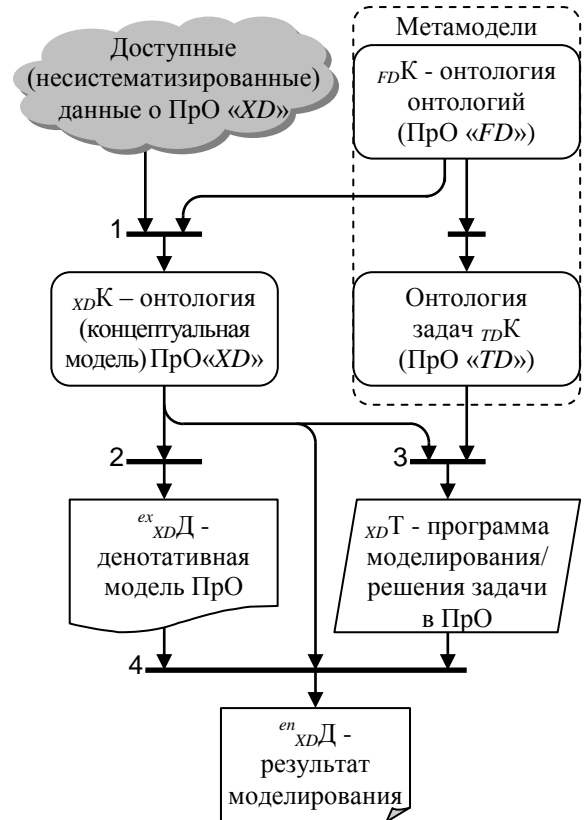


Рисунок 1 – Схема обработки информации при создании и использовании ППП

Продуктом онтологического анализа (или концептуализации; см. этап 1 на рисунке 1) целевой ПрО « $\langle XD \rangle$ » является концептуальная (К-) модель $\langle XD \rangle$ – онтология, которая представляет собой таксономию понятий ПрО, где каждое понятие с помощью механизма «атрибут-значение» описывает свойства-величины каждого объекта своего объема, способности этих объектов вступать в связи с другими элементами (валентности – см. [Смирнов, 2008]), методы работы с этими объектами, различные ассоциированные с понятием ограничения. Именно этот этап резонно сопоставить с разработкой ППП а, следовательно, ППП отождествить с онтологией. С прагматических позиций важно то, что в таком варианте одновременно с декларативным описанием целевой ПрО онтология непосредственно задает совокупность абстрактных типов данных, необходимых при разработке алгоритмов и программ, и схему объектно-ориентированной базы данных для размещения денотативных моделей системы объектов ПрО.

Остальные этапы обработки информации в схеме на рисунке 1 имеют характер синтеза:

- результатом этапа 2 - денотативного (Д-) моделирования системы объектов целевой ПрО,

является ее экзогенная модель $ex_{xD}D$. Этот акт выполняется на «языке» онтологии целевой ПрО;

- на этапе 3 конструируется денотативная модель специальной «технологической» ПрО «ТД» - трансформационная (Т-) модель xDT , реализующая знания о допустимых «вычислениях», связанных с Д-моделями целевой ПрО. В привычном смысле именно этому этапу соответствует разработка приложения, использующего ППП, т.е. спецификация сценария решения задачи, описания воздействий на экзогенную объектную модель, в результате которых она должна приобрести некоторые искомые пользователем свойства. Операционным базисом подобных трансформаций, аппаратом решения служит «исчисление», основу которого составляют функциональные части К-модели целевой ПрО. Потенциальная множественность Т-моделей закономерно отражает как множество различных задач в каждой ПрО, так и множество способов их решения;

- на этапе 4 – фазе вычислений, или вычислительного эксперимента, или выполнения программного приложения, использующего ППП, - знания, зафиксированные в xDT , реализуются (посредством интерпретации на ЭВМ) применительно к экзогенной модели $ex_{xD}D$ путем ее трансформации в эндогенную денотативную модель целевой ПрО $en_{xD}D$, которая с общих позиций и есть результат работы приложения.

Перечисленные модели (т.е. специфическая форма ППП, исходная модель ПрО, использующее ППП приложение и получаемый результат) составляют систему знаний и одновременно систему программно-технологических единиц ПОСМ. Однородность всех перечисленных моделей обеспечивают две априори заданные метамодели ПОСМ (см. рисунок 1): метаонтология для конструирования онтологий целевых ПрО и онтология задач (приложений), описывающая их на уровне, абстрагированном от конкретного содержания решаемых вопросов.

3. Интеграция знаний

Интерпретация ППП как формальных онтологий позволяет указать несколько эффективных подходов к решению проблемы интеграции разнородных знаний при разработке ППП и использующих их приложений.

На этапе построения ППП наиболее значимыми являются следующие приемы:

- конструирование понятийных структур для ППП на основе экспериментальных данных о ПрО расширенным методом анализа формальных понятий [Смирнов, 2011];
- формирование новых понятийных структур путем специализации и/или композиции имеющихся.

Другие способы интеграции знаний могут быть применены при разработке использующих ППП приложений (т.е. в Т-моделях) и реализованы при выполнении вычислений:

- одновременное манипулирование в ходе решения задачи несколькими объектными моделями ПрО, представленной одной онтологией;
- совмещение нескольких взглядов на целевую ПрО, которые выражены в приложении объектными моделями, построенными согласно различным онтологиям этой ПрО;
- совместное использование объектных моделей из разных ПрО, включая модели, конкретизирующие метод решения задачи (речь идет о том, что в методическом плане целесообразно различать предметно- и проблемно-ориентированные ППП);
- возможность построения приложений, автоматизирующих программирование ППП и использующих их приложений, на основе соответствующих метамodelей;
- организация взаимосвязанных вычислительных экспериментов для реализации альтернативных и эволюционных исследований ПрО.

Все эти методы интеграции обеспечиваются двумя «контурами» управления моделями различных типов на этапе вычислений (рисунок 2):

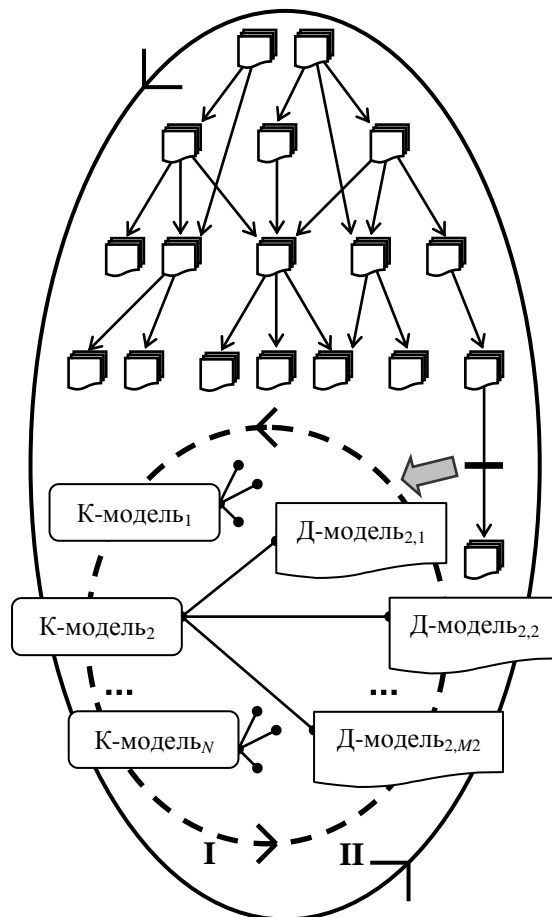


Рисунок 2 Управление на этапе вычислений:
I – задачный и II – проектный «контуры» управления

В первом, или задачном (поскольку его обеспечивают методы онтологии задач; см. I на рисунке 2), контуре путем переключения КМ - надлежащего определения «текущей» пары (К-модель, Д-модель) - обеспечивается работа с

несколькими Д-моделями из разных, вообще говоря, ПрО. При этом должен контролироваться состав допустимых ПрО (Т-модель должна быть компетентна в каждой из них) и корректность контекстов моделирования (К- и Д-модели в КМ должны представлять одну и ту же ПрО);

Второй, проектный, контур (П на рисунке 2) не является обязательным и связан с макроуправлением вычислениями. Для его поддержки необходимо специальный пакет программ – супервизор вычислений, который организует под управлением пользователя взаимосвязанные серии прогонов программ, структурируя сведения о выполненных вычислениях в форме растущего ациклического графа. Такая структура, по сути, определяет отдельную ПрО в области обработки информации, и указанный супервизор, контролируя «правильный» рост этой структуры, способен обеспечить транзакционные свойства приложений ППП.

4. Практические применения

В ИПУСС РАН в течение ряда лет развивается программный инструментарий - *gB*-система, реализующий разработку информационных моделей ПрО, разработку ППП и использующих их приложений на основе онтологического подхода. Сама *gB*-система строится в соответствии с изложенными выше принципами и имеет открытую архитектуру.

Инструментарий используется в научных исследованиях лаборатории анализа и моделирования сложных систем ИПУСС РАН, с его помощью создан ряд экспериментальных и промышленных программных комплексов (включая и инструментальные средства, и специализированные прикладные интеллектуальные системы) в машиностроении, экологии, в задачах организационного управления.

ЗАКЛЮЧЕНИЕ

При создании и эксплуатации ППП разработчик и пользователь оперируют онтологиями как основной разновидностью баз знаний и соответствующих им программно-технологических единиц. Разработчик строит ППП как онтологию ПрО, а пользователю предлагается расширенная трактовка приложения, использующего ППП, как многомодельной предметно-ориентированной среды моделирования. Конструирование подобных ПОСМ допускает включение в их состав средств автоматизации программирования и управления вычислениями, эффективно мобилизующих когнитивные и интеллектуальные ресурсы человека.

В целом онтологический подход дает достаточно ясный взгляд на состав, назначение и структуру программно-технологических компонентов и методов управления ими в ходе построения и использования ППП. При этом охватывается

большинство содержательных проблем разработки прикладного программного обеспечения: организации системы знаний об актуальной ПрО и о способах решения задач в этой ПрО, планирования решения задач и управления вычислениями, методологии отчуждения ППП от разработчика. Решения всех этих проблем оказываются в высокой степени унифицированными, а используемые программно-технологические составляющие пакетов программ и использующих их приложений становятся однородными.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Андрянов и др., 1985] Андрянов, А.И. Программирование на языке Симула-67 / А.И. Андрянов, С.П. Бычков, А.И. Хорошилов // М.: Наука, 1985. – 288 с.
- [Ершов и др., 1982] Ершов, А.П. Пакеты прикладных программ как методология решения прикладных задач / А.П. Ершов, В.П. Ильин // Пакеты прикладных программ. Проблемы и перспективы. – М.: Наука, 1982. – С. 1.
- [Куайн, 1996] Куайн, В. Онтологическая относительность / В. Куайн // Современная философия науки: знание, рациональность, ценности в трудах мыслителей Запада: хрестоматия. 2-е изд., перераб. и доп. – М.: Логос, 1996.- 400 с.
- [Нариньяни, 2001] Нариньяни, А.С. Кентавр по имени ТЕОН: Тезаурус + Онтология / А.С. Нариньяни // Труды международного семинара Диалог'2001 по компьютерной лингвистике и ее приложениям. Т. 1, С. 184-188.
- [Поспелов, 1988] Поспелов, Г.С. Искусственный интеллект – основа новой информационной технологии / Г.С. Поспелов // М.: Наука, 1988. – 280 с.
- [Смирнов, 2008] Смирнов, С.В. Прагматика онтологий: объектно-ориентированная модель знаний о предметной области // КИИ-2008. Труды конференции. Т. 3, С. 208-216.
- [Смирнов, 2011] Смирнов, С.В. Построение онтологий предметных областей со структурными отношениями на основе анализа формальных понятий // Знания – Онтология – Теория: Материалы Всероссийской конф. с международным участием Т. 2. – Новосибирск: Ин-т математики СО РАН, 2011. – С. 103-112.
- [Guarino, 1995] Guarino, N. Formal ontology, conceptual analysis and knowledge representation / N. Guarino // Int. J. of Human Computer Studies. - 1995. - V.43. №5/6. - P. 625-640.

PROGRAM PACKAGES AS A FORMAL ONTOLOGIES: CONSTRUCTION AND APPLICATION

Smirnov S.V.

*Institute for the Control of Complex System
of the Russian Academy of Sciences,
Samara, Russia
smirnov@iccs.ru*

The paper presents an ontological approach to the construction and application of program packages. The rationale behind the approach is a determinacy process's of reflection and modeling, which consists in the construction and transformation of semantic heterogeneous object-oriented frameworks that simulate reality. The technology in question has all the characteristics of componential design's of knowledge bases, program packages and applications, it is focused on the knowledge integration and a high degree of uniformity.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 681.3:519.6

ПРИНЦИПЫ ПОСТРОЕНИЯ ТЕХНОЛОГИИ ГРАФОСИМВОЛИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Коварцев А.Н., Жидченко В.В., Попова-Коварцева Д.А., Аболмасов П.В.

Самарский государственный аэрокосмический университет им. академика С.П. Королева (национальный исследовательский университет), г. Самара, Российская Федерация

kovr_ssau@mail.ru

В статье рассматривается технология графосимволического программирования (ГСП), реализующая визуальный стиль программирования. Визуальное программирование повышает наглядность представления моделей алгоритмов программ, существенно уменьшает число ошибок, допускаемых на этапе проектирования и кодирования программ, и тем самым повышает надежность разрабатываемых программ. В данной работе обсуждаются наиболее важные аспекты технологии ГСП.

Ключевые слова: технология графосимволического программирования, проектирование сложных программ, автоматизация программирования, визуальное программирование, предметная область, онтологии, автоматизация тестирования, творческие задачи, искусственный интеллект

ВВЕДЕНИЕ

Настоящая статья подготовлена по материалам научных исследований, проводимых в течение нескольких лет в Самарском государственном аэрокосмическом университете имени С.П. Королева (СГАУ) в области технологий автоматизации программирования и визуального моделирования вычислительных процессов. На кафедре программных систем СГАУ это научное направление представлено технологией графосимволического программирования (ГСП) ([Коварцев, 1999]).

Широкое внедрение средств вычислительной техники в различные сферы жизни и деятельности человека стимулировало развитие автоматизированных методов и инструментальных средств, применяемых для создания прикладного программного обеспечения (ПО). Производство современного ПО происходит на фоне высоких требований, предъявляемых к качеству создаваемых программ при значительной сложности выполняемых ими функций. “Идеальной” технологией программирования можно считать такую технологию, которая по некоторому достаточно неформальному описанию объекта программирования автоматически генерирует текст синтаксически и семантически корректной программы.

В 80-х годах XX века сформировалась идея о том, что целью программирования является не столько порождение программы как таковой, а

создание технологических условий, когда разрабатываемое программное обеспечение легко адаптируется к новым обстоятельствам и новому пониманию решаемой задачи. Р. Хемминг так формулирует этот тезис: “Здоровая вычислительная практика требует постоянного исследования изучаемой задачи не только перед организацией вычислений, но также в процессе его развития и, особенно на той стадии, когда полученные числа переводятся обратно и истолковываются на языке первоначальной задачи”. Перечисленные выше обстоятельства привели к осознанию необходимости реализации интегрированного окружения поддержки всего жизненного цикла ПО, что обусловило появление инструментальных средств автоматизации проектирования программных систем (CASE-технологий).

В настоящее время наблюдается возрастание роли визуализации, используемой как в технологиях программирования, так и представлении результатов работы программ. Новый графический подход к решению проблемы автоматизации разработки ПО, основанный на идее привлечения визуальных форм представления программ, в большей степени соответствует образному способу мышления человека. Применение графических методов обещает кардинально повысить производительность труда программиста. Визуальное программирование, бесспорно, обладает достоинством наглядного представления информации и гораздо лучше соответствует природе человеческого восприятия, чем методы традиционного, текстового программирования.

Кроме того, графическая форма записи по сравнению с текстовым представлением программ обеспечивает более высокий уровень их структуризации, соблюдение технологической культуры программирования, предлагает более надежный стиль программирования. За счет использования графических моделей удастся не только сократить время разработки параллельных вычислительных процессов, но и повысить их надежность, т.к. графическая нотация допускает формальное математическое описание модели, по которому может быть проведена ее автоматическая верификация и оптимизация.

Интеллектуальная сложность процессов разработки современного ПО настоятельно требует создания инструментальных средств представления, как абстракций, так и программных объектов на всех этапах их разработки. Предлагаемая технология ГСП в какой-то мере ориентирована на решение перечисленных проблем.

1. Концептуальная модель технологии ГСП

Технологию ГСП определим как технологию проектирования и кодирования алгоритмов программного обеспечения, базирующуюся на графическом способе представления программ, преследующую цель полной или частичной автоматизации процессов проектирования, кодирования и тестирования ПО.

Данная технология исповедует два основополагающих принципа:

- визуальную, графическую форму представления алгоритмов программ и других компонент их спецификации;
- принцип структурированного процедурного программирования.

Сразу условимся, что технология ГСП действует в рамках некоторой информационной среды, которую будем называть предметной областью программирования (ПрОП).

Под *предметной областью программирования* понимается среда программирования, состоящая из общего набора данных (словарь данных) и набора программных модулей (библиотека программных модулей).

Для представления алгоритмов в технологии ГСП используется модель объекта с дискретными состояниями. Основу такой модели составляет предположение о том, что для любого объекта программирования тем или иным способом можно выделить конечное число состояний, в которых он может пребывать в каждый момент времени. Тогда развитие вычислительного процесса можно ассоциировать с переходами объекта из одного состояния в другое. В математике такая концепция в качестве способа абстрагирования плодотворно используется достаточно давно: Марковские цепи,

теория массового обслуживания, теория формальных грамматик и автоматов, моделирование систем и т.д.

Для уточнения понятия *состояния*, определимся с используемой в технологии ГСП концепцией модели алгоритма. Существует следующие три основных типа универсальных алгоритмических моделей:

Первый тип модели связывает понятие алгоритма с наиболее традиционными понятиями математики - вычислениями и числовыми функциями. Наиболее известная и изученные модели такого типа - *рекурсивные функции*.

Второй - основан на представлении об алгоритме как о некотором детерминированном устройстве, способном выполнять на каждом шаге лишь примитивные операции. Одним из многочисленных представителей этого типа является *машина Тьюринга*.

Третий - это преобразование слов в произвольных алфавитах, в которых элементарными операциями являются подстановки. Среди моделей этого типа наиболее известны *канонические системы Поста, нормальные алгорифмы Маркова* и т.д.

В технологии графосимволического программирования используется первый тип формализации понятия алгоритма, когда программа A интерпретируется как некоторая вычисляемая функция:

$$f : in(D) \rightarrow out(D),$$

где $in(D)$ - множество входных данных функции f , $out(D)$ - множество выходных данных функции f . Далее для простоты мы не будем разделять множество программных модулей $\{A_i\}$ и приписанных им вычисляемых функций $\{f_j\}$. Предположим, что каждый модуль является вычисляемой функцией и $A_i \in \mathcal{F}$, где \mathcal{F} - множество вычисляемых функций.

Более строго эта концепция исследована для вычислительной модели, названной *машиной Колмогорова* [Успенский, 1987]. Там же можно найти едва ли не единственное формальное определение понятия *состояния* вычислительного процесса. По Колмогорову, *состояния* - суть конструктивные объекты, под которыми в случае технологии графосимволического программирования можно понимать ансамбли конкретизаций структур данных (входных или вычисляемых), используемых в алгоритме. На каждом шаге итерации реализуется переработка текущего состояния структур данных D в новое состояние D^* с помощью некоторой локальной функции $D^* = f_k(D)$. Процесс переработки $D^0 = D$ в $D^1 = f_1(D^0)$, D^1 в

$D^2 = f_2(D^1) = f_1(f_2(D))$ и т.д. продолжается до тех пор, пока не появится сигнал о получении решения.

Определим *граф состояний* G как ориентированный помеченный граф, вершины которого - суть состояния, а дугами отмечаются переходы системы из состояния в состояние.

Формально, *состояние* - это достижение объектом \mathcal{O} определенной конкретизации структур данных, требующее выполнения определенных действий над данными предметной области посредством вычислимой функции A_k . Поэтому каждая вершина графа помечается соответствующей локальной вычислимой функцией A_k . При этом следует помнить, что состояние графа S_j - это некое понятие («концепт»), связанное с описанием объекта \mathcal{O} , а вычислимая функция A_k , приписанная состоянию, - функциональные действия, которые необходимо выполнить при переходе объекта в это состояние.

Одна из вершин графа, соответствующая начальному состоянию, объявляется начальной вершиной и, таким образом, граф оказывается *инициальным*.

Дуги графа проще всего интерпретировать как *события*. С позиций предлагаемого подхода, *событие* - это изменение *состояния* объекта \mathcal{O} предметной области программирования, которое влияет на развитие вычислительного процесса.

На каждом конкретном шаге работы алгоритма в случае возникновения коллизии, когда из одной вершины исходят несколько дуг, соответствующее *событие* определяет дальнейший ход развития вычислительного процесса алгоритма. Активизация того или иного события так или иначе зависит от состояния объекта, которое в свою очередь определяется достигнутой конкретизацией структур данных D объекта \mathcal{O} .

Для реализации *событийного управления* на графе состояний G введем множество предикативных функций $P = \{P_1, P_2, \dots, P_l\}$. Под *предикатом* будем понимать логическую функцию $P_i(D)$, которая в зависимости от значений данных D принимает значение равно 0 или 1.

Дугам графа G поставим в соответствие предикативные функции. Событие, реализующее переход $S_i \rightarrow S_j$ на графе состояний G , инициируется, если модель объекта \mathcal{O} на текущем шаге работы алгоритма находится в состоянии S_i и соответствующий предикат $P_{ij}(D)$ (помечающий данный переход) истинен.

В общем случае предложенная концепция

допускает одновременное наступление нескольких событий, в том случае, когда несколько предикатов, помечающих дуги (исходящих из одной вершины), приняли значение истинности. В ГСП этот конфликт разрешается назначением приоритетов дугам графа.

Существенно, что изображение программ в виде ориентированного помеченного графа естественно для восприятия человеком. Направленная дуга служит очевидным изображением перехода из одного состояния вычислительного процесса в другое, вершина - выполняемой вычислительной функции, а в целом ориентированный граф наглядно представляет все пути, по которым может развиваться вычислительный процесс. В этом случае логические особенности разрабатываемого программного модуля проявляются в характерной для него топологии графа. Можно сказать, что графическое представление программ позволяет задействовать непосредственное образное восприятие, расширяя возможности человека при разработке и анализе сложных программ.

Определим универсальную алгоритмическую модель технологии ГСП четверкой

$$M = \langle D, \mathcal{F}, P, G \rangle, \quad (1)$$

где D - множество данных (ансамбль структур данных) некоторой предметной области программирования; \mathcal{F} - множество вычислимых функций; P - множество предикатов, действующих над структурами данных предметной области D ; G - граф состояний объекта \mathcal{O} .

Представленная алгоритмическая модель является универсальной, поскольку допускает описание любых алгоритмов.

2. Типы данных

В понятии *типа данных* T сигнатуры Σ обычно выделяют:

- *спецификацию типа* данных сигнатуры Σ ;
- и соответствующую ей *реализацию типа* данных [Замулин, 1990].

Здесь под сигнатурой понимается пара $\Sigma = \langle B, \Omega \rangle$, где B - множество имен-основ (имена базовых типов базового языка программирования или производных от этих типов данных), а $\Omega = (B^* \times B)$ - индексированное семейство множеств имен операций, B^* - множество всех цепочек элементов множества B .

$$\begin{aligned} \text{Например, } B &= \{\text{int, double, real, } \dots\}, \\ \Omega &= \{+(double, double \rightarrow double), \\ &+ (double, real \rightarrow double), \dots\}. \end{aligned}$$

В таком определении типа данных отражаются два аспекта: пользовательский, когда программист, составляя свою программу, видит тип как спецификацию; и машинный, связанный со

способом реализации в ЭВМ обозначенного типа данных.

Типизация данных полезна не только в гносеологическом смысле, как инструмент изучения теории программирования, но и в чисто прикладном аспекте, позволяя избежать большого количества ошибок, поскольку заставляет программиста точно определять все используемые им объекты и лучше контролировать свою программу. В этом смысле использование чисто синтаксического аспекта спецификации типа данных можно считать недостаточным.

В технологии ГСП понятие типа данных расширено понятием *интерпретации* данного. Действительно, во многих предметных областях чисто “языковое” представление типа данного является неполным. Например, в физике в формуле $F = am$ все три компонента формулы F , a , m имеют естественный языковой тип `double` с определенными на этом типе операциями (+, -, *, /). Однако каждое из перечисленных данных имеет самостоятельную смысловую интерпретацию (для физических параметров она определяется его размерностью): F - сила [н], a - ускорение [м/сек²], m - масса [кг].

Очевидно, что передача в подпрограмму вместо ускорения a (тип `double`) скорости v (тип `double`), не вызовет синтаксической ошибки, но приведет к неправильному исполнению программы. Следует отметить, что такого рода ошибки обычно очень сложно распознаются. В качестве типа *интерпретации* данного в ГСП предлагается использовать теорию размерности.

Определим множество основ типа интерпретации данных S_{int} как множество образующих типов интерпретации и их производных. Например, для физических задач это множество имеет вид:

$$S_{int} = \{[м], [кг], [сек], [м/сек], \dots\}.$$

Множество операций в простейшем случае может быть представлено естественными операциями над типами интерпретации $\Omega_{int} = \{+, -, *, /\}$. При описании типов интерпретации данных базовых модулей необходимо ввести множество аксиом A_{int} , действующих над типами интерпретаций. Множество A_{int} состоит из замкнутых σ -формул. Например, для формулы $F = am$ аксиомой типобразования служит формула: $T_1 = T_2 * T_3$.

Тип *интерпретации* данного определим как $T_{int} = T_{int} = \langle \Sigma_{int}, A_{int} \rangle$, где $\Sigma_{int} = \langle S_{int}, \Omega_{int} \rangle$. Таким образом, под **типом данного** в технологии ГСП понимается пара $T = \langle \Sigma, \Sigma_{int} \rangle$.

Если определение некоторого типа данного не расширено семантическим аспектом - описанием типа интерпретации, то считается, что данный тип

имеет “пустую” размерность «[]».

При верификации используемых типов данных в ГСП первоначально проверяется синтаксическая составляющая описания типа, а затем сравниваются типы интерпретации.

Спецификация данных в технологии ГСП реализована в виде совокупности таблиц информационного фонда системы. В отдельную таблицу, называемую *архивом типов данных*, сведены описания всех необходимых атрибутов типов данных, начиная с имени типа, описания родового языкового типа или его редукции, описания типа интерпретации и заканчивая описанием области возможных значений (доменов).

3. Базовые модули

В качестве исходного «строительного материала» в технологии ГСП выступают две категории: базовые модули (подпрограммы) и типы данных. Базовые модули представляют собой перечень локальных вычислимых функций, на основе которых в конечном итоге порождаются все объекты технологии ГСП. Типы данных описывают синтаксический и семантический аспекты строения данных, используемых в базовых функциях, а также и в объектах технологии ГСП.

Порождение первоначального множества вычислимых функций (базовых модулей) производится на любом из существующих языков программирования, например, на языке C++.

В технологии ГСП под **базовым модулем** будем понимать независимую программную единицу, реализующую определенную функцию в процессе преобразования некоторого агрегата данных.

4. Типы модулей

Введем понятие *типа модуля* (*вычислимой функции*) как обобщение понятия типа операции, которое определим как множество отображений из области определения функции в область результата.

Причем областью определения функции считается декартово произведение множеств значений нескольких типов данных (типов формальных параметров), а областью результатов - множество значений некоторого одного типа данных. При этом тип функции изображается $T_{i_1}, T_{i_2}, \dots, T_{i_n} \rightarrow T_{j_1}, T_{j_2}, \dots, T_{j_m}$, где $T_{i_1}, T_{i_2}, \dots, T_{i_n}, T_{j_1}, T_{j_2}, \dots, T_{j_m}$ - типы формальных параметров (исходных и результатов вычислений) рассматриваемой функции.

С каждым типом функции (подпрограммы) связывают две операции: создания функции и аппликации функций к своим аргументам. Первая операция в языках программирования связана с изображением функции в виде: заголовок функции плюс тело функции. Вторая операция - в виде обращения к функции.

В теории программирования подпрограммы это одно из фундаментальных средств абстрагирования и искусственного расширения возможностей языка, однако, правила реализации первой из перечисленных выше операций - составления тела подпрограммы, обычно синтаксически неотличимы от правил кодирования основного текста программы. Последнее достигается за счет введения понятия формальных параметров. В то время как между основной программой и подпрограммами имеет место существенная разница.

При разработке основной программы преследуется конкретная практическая цель, например, произвести расчеты некоторого технического устройства, физического явления, математических формул и т.д. При этом данным, используемым в программе “придается” вполне конкретный смысл (давление, скорость, пропускная способность и т.п.). В подпрограммах формальные параметры лишены конкретной смысловой нагрузки.

Действительно, в подпрограммах важен лишь тип параметра и порядок его использования, а “осмысление” назначения параметров возникает только после их аппликации к фактическим параметрам. Например, процедура, реализующая формулу $A = V * C$, в одной интерпретации типов данных вычисляет силу F по заданным ускорению a и массе m материальной точки ($F = a m$), в другой - путь S , по заданной скорости V и времени t ($S = V t$).

Таким образом, тип базового модуля определим как отображение типов данных из области определения на область их значений $B : T_1, T_2, \dots, T_n \rightarrow T_{j_1}, T_{j_2}, \dots, T_{j_m}$, которые реализуются в соответствии с принятыми в B операциями над типами данных.

5. Объекты технологии ГСП

В технологии ГСП в качестве программных единиц рассматриваются **объекты**. По способу порождения и функциональному назначению различают три типа объектов: **акторы**, **агрегаты** и **предикаты**. Все они имеют конкретный содержательный смысл и действуют в рамках предметной области программирования.

В предметной области, как правило, заранее уже определен терминологический словарь данных (параметров, переменных или констант). Поэтому программирование в рамках технологии ГСП начинается с формирования **словаря данных** ПрОП, представляющего собой таблицу, в которой каждому данному присвоено уникальное имя, задан тип, начальное значение данного и краткий комментарий его назначения в ПрОП.

Кроме словаря данных и каталога типов данных, информационную среду определяют объекты ГСП. Под объектом понимается специальным образом построенный в рамках технологии ГСП

программный модуль, выполняющий определенные действия над данными ПрОП.

5.1. Акторы

Одним из объектов технологии ГСП является **актор**. Актор формируется из базового модуля путем привязки абстрактных типов данных базового модуля к данным предметной области. По сути дела **актор** порождается в результате аппликации базового модуля к своим аргументам.

Актор производит те же действия, что и породивший его базовый модуль, но над конкретными данными ПрОП. В отличие от базовых модулей, каждый актер является содержательным программным модулем, который выполняет понятные функции в рамках заданной предметной области. Актеры в технологии ГСП реализуют отображение над множеством данных предметной области:

$$A_k : D_k^{in} \rightarrow D_k^{out}, \quad (2)$$

где $D_k^{in} = \{d_1^{in}, d_2^{in}, \dots, d_n^{in}\}$ - множество входных данных актора A_k , $D_k^{out} = \{d_1^{out}, d_2^{out}, \dots, d_n^{out}\}$ - множество выходных данных актора A_k .

Между базовым модулем и актором осуществляется односторонняя связь типа “один ко многим”. Каждый актер имеет свой прототип в виде базового модуля, а на основании каждого базового модуля можно построить один или несколько акторов. Это свойство полиморфизма объектов позволяет избежать избыточности при порождении новых акторов, которые различаются между собой только привязкой к данным. Другими словами, на основе одного отлаженного и оттестированного базового модуля за счет механизма автоматизированной привязки по данным можно построить несколько корректных акторов, что позволяет значительно повысить уровень надежности порождаемых программных кодов.

Порождение актора производится путем формирования, так называемого, **паспорта** объекта. Процедура паспортизации базового модуля заключается в установке соответствия между списком типов данных базового модуля и данными предметной области, таким образом, что каждому формальному параметру (типу данных) ставится в соответствие конкретное данное ПрОП.

Соответствие между базовым модулем B_i и актором A_j порождает соответствие между подмножеством типов T_i данных и подмножеством самих данных D_j предметной области:

$$\begin{cases} B_i(T_i^{in}, T_i^{out}) \rightarrow A_j(D_j^{in}, D_j^{out}) \\ T_i = (T_i^{in}, T_i^{out}) \rightarrow D_j = (D_j^{in}, D_j^{out}) \end{cases}$$

Для стандартизированной формы базового модуля операция конкретизации типов данных сводится к процедуре формирования списка фактических

параметров, который определяется паспортом порождаемого актора. В этом смысле паспорт актора и базовый модуль полностью определяют актор ПРОП.

5.2. Предикаты

В процессе реализации алгоритма в рамках технологии ГСП передача управления между объектами осуществляется с помощью управляющих объектов-предикатов. Формально предикат представляет собой отображение из множества данных предметной области на множество логических значений “истина” или “ложь”: $P_k : (d_1, d_2, \dots, d_m) \rightarrow \{0, 1\}$.

5.3. Агрегаты

Объекты (акторы или предикаты) являются исходным материалом для визуального программирования. Результатом визуального программирования являются агрегаты. Агрегат создается в форме графа, в котором объекты ПРОП играют роль вершин и дуг. Дуги - предикаты, а вершины - акторы или агрегаты. Дуги графа определяют передачу управления от одной вершины к другой.

Формально агрегат представляет собой помеченный ориентированный граф с одной входной и несколькими выходными (концевыми) вершинами:

$$G = \{ \mathcal{F}, P \},$$

где $\mathcal{F} = \{ A_1, A_2, \dots, A_n \}$ - множество акторов, которые являются вершинами графа, $P = \{ P_1, P_2, \dots, P_m \}$ - множество предикатов, которые представляют дуги графа.

Развитие вычислительного процесса в агрегате происходит путем передачи управления из одной вершины в другую, начиная с начальной. Этот процесс может быть завершен по двум причинам: либо достигнута концевая вершина графа, из которой нет исходящих дуг, либо из текущей вершины отсутствуют разрешенные другими предикатами переходы в другие вершины. Если в процессе передачи управления сложилась ситуация, когда истинными одновременно являются несколько предикатов, то управление будет передано по предикату, имеющему наибольший приоритет.

При таком подходе, между агрегатом в технологии ГСП и блок-схемой алгоритма существуют аналогии. Отличие заключается в том, что агрегат не имеет специальных управляющих блоков (условие и выбор), и передача управления всегда осуществляется посредством проверки предиката, который в частном случае может быть тождественно истинным. Это упрощает визуальный анализ алгоритма, за счет чего можно сократить число структурных ошибок. Например, ошибок, связанных с переусложненной структурой (неправильно вложенные циклы, неверная передача

управления и т.п.), либо ошибок вызванных противоречиями в самом графе (непредусмотренные циклы).

В отличие от акторов и предикатов, которые полностью определяются своими паспортами, при порождении агрегата с помощью специального компилятора формируется текст нового объекта ПРОП, который после трансляции заносится в библиотеку объектных модулей ПРОП.

6. Модель межмодульного интерфейса

Проблема передачи информации от одной программы к другой традиционно представляет собой одну из наименее популярных проблем в среде программистов и одну из проблем, которая служит источником наибольшего количества ошибок в разрабатываемом программном обеспечении.

В технологии ГСП введён стандарт на организацию межмодульного информационного интерфейса. Стандарт обеспечивается выполнением пяти основных правил:

1). Вводится единое для всей предметной области хранилище данных, актуальных для ПРОП (общая память). Полное описание данных размещено в словаре данных ПРОП. Любые переменные, не описанные в *словаре* данных, считаются локальными данными для тех объектов ГСП, где они используются.

2). В пределах ГСП описание типов данных размещается централизованно в *архиве типов* данных.

3). В базовых модулях в качестве механизма доступа к данным допускается только передача параметров *по адресам* данных.

4). Привязка данных объектов ПРОП реализована в паспортах объектов ПРОП.

5). В технологии ГСП не рекомендуется использовать иные способы организации межпрограммных связей по данным.

Предложенный стандарт позволяет полностью отделить задачу построения межмодульного информационного интерфейса от кодирования процедурной части программы, а также частично автоматизировать процессы построения информационного интерфейса.

С информационной точки зрения каждый объект ГСП f_i (2) представляет собой функциональное отображение области определения объекта D_i^{in} на область значений D_i^{out} .

В общем случае $D_i^{in} \cap D_i^{out} \neq \emptyset$ (в объекте могут быть модифицируемые данные) и $D_i^{in}, D_i^{out} \in D$, где D - полная область данных ПРОП.

Формально, сущность проблемы организации передачи данных между объектами в рамках некоторого модуля-агрегата f_Σ можно определить как задачу построения области данных агрегата f_Σ - $D_\Sigma = D_\Sigma^{in} \cup D_\Sigma^{out}$ и установления соответствий между данными $D_\Sigma = \{d_1, d_2, \dots, d_{n_\Sigma}\}$ и данными $D_i = \{d_1^i, d_2^i, \dots, d_{n_i}^i\}$ объектов f_1, f_2, \dots, f_m , из которых составлен агрегат f_Σ .

В традиционном программировании любое, даже незначительное изменение структур данных в модулях вызывает необходимость “ручной” переделки соответствующих информационных связей ρ_j . В технологии ГСП этот процесс автоматизирован за счет использования отношений ρ_j , описанных и хранящихся отдельно от программной реализации объекта в «паспортах» модулей.

Отношение ρ_j в ГСП формируется “паспортизацией” типов данных базовых модулей, т.е. за счет “опредмечивания” формальных параметров базовых модулей. В этом смысле отношение ρ_j является по сути “паспортом” модуля и вместе с базовым модулем определяют понятие актора или предиката.

7. Управление вычислительным процессом

В технологии ГСП для объектов - агрегатов в зависимости от стиля программирования (последовательного или параллельного) используются различные схемы управления вычислительным процессом. Каждому стилю программирования соответствует собственная схема управления в агрегате при незначительных изменениях в синтаксисе изображения агрегата.

Объекты, порожденные из базовых модулей (акторы), управляются в соответствии с правилами, принятыми в базовом языке программирования. Синтез агрегата происходит на основе его графического изображения, представленного соответствующими структурами данных, которые хранятся в информационном фонде технологии ГСП.

В ГСП агрегат фактически состоит из двух компонент:

- универсальной управляющей программы граф-машины (GM), которая в соответствии со структурой графа управления каждого из агрегатов, управляет развитием вычислительного процесса на агрегатах ПрОП;
- структур данных описания графа управления каждого из агрегатов.

Централизация функций управления в рамках одной программы (граф-машины) на самом деле очень удобное решение, поскольку позволяет:

1. контролировать вычислительный процесс в целом, и в случае нештатных ситуаций, принимать системные решения;
2. реализовать сбор статистической информации о характеристиках надежности каждого из модулей; вычислительной сложности модулей; маршрутах развития вычислительного процесса и т.д.

8. Онтологический аспект технологии ГСП

Основоположник использования онтологий в области информационных технологий Томас Груббер [Gruber, 1991] определяет онтологию как «спецификацию концептуализации». При этом под «концептуализацией» можно понимать спецификацию знаний об окружающем мире, т.е. описание структуры Бытия безотносительно к какой-либо инженерной задаче. Для программистов естественнее и ближе понимание концептуализации как построение модели решаемой задачи, т.е. её концептуальной схемы.

Под формальной моделью онтологии O часто понимают [Скобелев, 2011] упорядоченную тройку вида

$$O = \langle C, R, F \rangle, \quad (3)$$

где C – конечное множество понятий (концептов) предметной области, R – конечное множество отношений между понятиями предметной области, F – конечное множество функций интерпретации, заданных на понятиях и/или отношениях онтологии O .

Сравним между собой концептуальные схемы (1) и (3). Множество понятий (концептов) предметной области C онтологии O можно связать с множеством состояний S модели M объекта \mathcal{O} . Конечное множество отношений R онтологии O в технологии ГСП описываются графами состояний G_i . И, наконец, конечное множество функций интерпретации F – это множество вычислимых функций, \mathcal{F} и предикатов P . Фактически модель алгоритма (1) является *описанием онтологии* разрабатываемого объекта \mathcal{O} в данной предметной области. В совокупности, все модели объектов предметной области формируют описание её онтологии. Как правило, множество предикатов и, в меньшей степени, множество вычислимых функций общезначимы для всей предметной области и повторно используются в разных объектах ПрОП.

Данное обстоятельство наделяет технологию ГСП рядом полезных свойств. Удобной для программиста визуальной формы описания модели алгоритма. Сформированная средствами ГСП предметная область содержит информацию, необходимую для автоматической генерации

отчетов для разрабатываемых программных приложений. Но главной особенностью данной технологии является возможность построения модели алгоритма без использования заранее сформированного «плана» алгоритма. В процедурных языках программирования для написания программы решения некоторой практической задачи необходимо первоначально решить задачу, т.е. разработать алгоритм, а затем реализовать его в программных кодах. В технологии ГСП для разработки алгоритма, а, следовательно, и кодов программы, достаточно иметь лишь идею решения задачи.

Последнее связано, по-видимому, с присутствием онтологических аспектов в технологии ГСП. При разработке программных приложений в технологии ГСП пользователь описывает онтологию предметной области, вводя для неё новые данные, новые понятия и функциональные отношения по мере изучения объекта программирования, в итоге параллельно формируется и модель алгоритма программы, связывающая все эти понятия.

9. Пример. Задача «Ханойские башни»

Задача о «Ханойских башнях» - одна из наиболее известных головоломок, для которой разработано большое количество алгоритмов её решения [Легалов, 2002]. Суть задачи заключается в следующем.

Пусть имеются три стержня X, Y, и Z. На стержень X надето N дисков разного диаметра, упорядоченные в направлении убывания диаметров от основания стержня. Цель игры заключается в переносе всех дисков со стержня X на стержень Z (по одному диску за раз), используя при этом промежуточный стержень Y. Причем ни один диск большего диаметра нельзя ставить на диск меньшего диаметра.

Проиллюстрируем возможности технологии ГСП на примере решения задачи о ханойских башнях.

При разработке алгоритма решения будем руководствоваться следующими простыми принципами, входящими в условия задачи:

1. При перекладывании дисков всегда применяется правило о недопустимости установки диска большего диаметра на диск меньшего диаметра.
2. Соблюдать правило приоритетов для операций переноса дисков. Установить приоритеты операциям переноса дисков (в порядке убывания приоритетов) следующим образом:
 $(X \rightarrow Z), (X \rightarrow Y), (Y \rightarrow Z),$
 $(Y \rightarrow X), (Z \rightarrow Y), (Z \rightarrow X)$
3. По возможности желательно в первую очередь снимать диски со стержня X и переносить их на стержень Z.

Сформируем словарь данных предметной области. Данные, необходимые для решения задачи, приведены в таблице 1. Числом 777 обозначено основание стержней.

Таблица 1. Словарь данных

Имя данного	Тип	Нач. значение	Комментарий
Mx	STERJN	{1,2,3,5,6,777}	Массив дисков стержня X
My	STERJN	{777,0,0,0,0,0}	Массив дисков стержня Y
Mz	STERJN	{777,0,0,0,0,0}	Массив дисков стержня Z
X	int	777	Размер верхнего диска на стержне X
Y	int	777	Размер верхнего диска на стержне Y
Z	int	777	Размер верхнего диска на стержне Z
Nst	int	1	Текущий номер стержня

В дальнейшем нам потребуется программа выполняющая операцию переноса диска с одного стержня на другой.

Рассмотрим базовый модуль *perA_B(int A, int B, STERJN Ma, STERJN Mb)*, реализующий перенос верхнего диска с абстрактного стержня A на абстрактный стержень B. Сама по себе программа базового модуля достаточно проста: первый элемент массива Ma переносится на первое место массива Mb (при этом реализуются все необходимые перемещения остальных элементов этих массивов) и переменным A и B присваиваются размеры первых элементов массивов Ma и Mb.

На основе базового модуля *perA_B* путём привязки формальных параметров A, B, Ma, Mb к данным ПрОП X, Y, Mx, My порождаются акторы, связанные с понятиями «Перенос диска со стержня X на стержень Y» (это понятие условно обозначим «X->Y»). Соответствующую функциональность реализует актор: *perA_B(X, Y, Mx, My)*. Точно так же порождаются понятия «X->Z», «Y->Z», ..., «Z->X» с помощью акторов: *perA_B(X, Z, Mx, Mz), perA_B(Y, Z, My, Mz), ..., perA_B(Z, X, Mz, Mx)*.

Аналогичным образом вводятся понятие «Визуализация перемещений дисков» и актор, отображающий на экране дисплея положение дисков на стержнях: *move(Mx, My, Mz)*, а также понятия, связанные с управлением графическим режимом: «Инициализация графики», «Закрытие графического режима» и т.д.

Отметим, что многие понятия на схемах имеют свои графические образы, что упрощает восприятие граф-программ человеком.

Представим себе ситуацию, что уже выбран стержень, с которого будет сниматься диск. Необходимо построить алгоритм, определяющий на

какой стержень его необходимо переместить? Его несложно реализовать с помощью граф-программы «Перенос диска со стержня X», представленной на рисунке 1. В технологии ГСП такие объекты называются *агрегатами* и пополняют список функций интерпретации *R* онтологии предметной области.

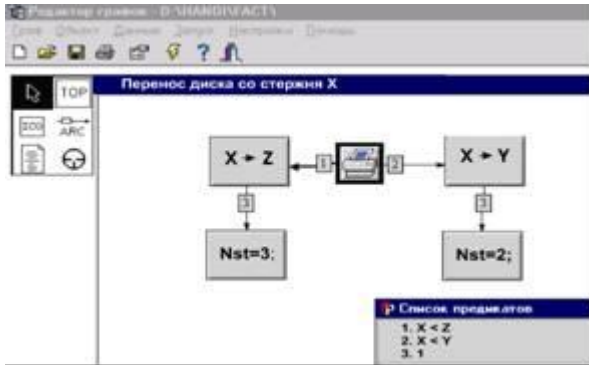


Рисунок 1 – Агрегат «Перенос диска со стержня X»

На рисунке корневая вершина (обведена жирной рамкой) выводит на экран дисплея текущее состояние стержней. Вершины «X->Z» и «X->Y» связаны с понятиями переноса диска со стержня X на стержни Z и Y. Причем переход в вершину «X->Z» происходит при выполнении условия $X < Z$, а в вершину «X->Y» - при истинности выражения $X < Y$. На графе более приоритетные дуги обозначены «жирными стрелками». Безусловный переход обозначен символом «1». Переменная *Nst* определяет номер стержня, на который реализован перенос диска. На графе этот факт отражен в вершинах $Nst=3;$ и $Nst=2;$. Стержням X, Y, Z присвоены номера 1, 2, 3.

Аналогично строятся агрегаты: «Перенос диска со стержня Y», «Перенос диска со стержня Z».

Первоначально все диски находятся на стержне X. С диска X, в общем случае, мы можем переместить верхний диск, либо на стержень Z, либо на стержень Y. Начальные действия построения агрегата «Ханойские башни» показаны на рисунке 2. Здесь вершина 1 «Инициализация графики» - устанавливает графический режим отображения информации

Вершина 2 привязана к агрегату «Перенос диска со стержня X», представленного на рисунке 1. Вершины 3 и 4 формально фиксируют факт переноса диска на стержни Y или Z (управляются предикатами $Nst==3$ и $Nst==2$). Эти вершины не имеют привязок к каким-либо акторам (т.е. являются «пустыми»), но необходимы для исключения циклических переносов дисков между двумя стержнями. Иными словами, если, например, диск был перенесен на стержень Y, то выбор стержня для выполнения следующей операции должен быть реализован между стержнями X и Y. Это обстоятельство приводит к развитию алгоритма решаемой задачи, представленному на рисунке 3.

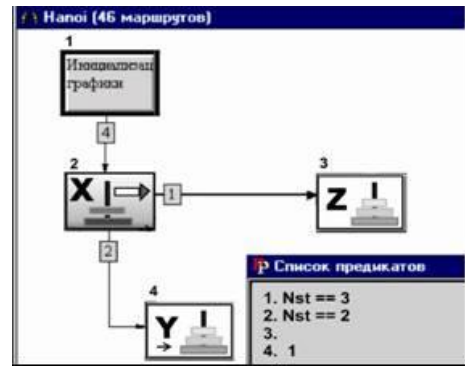


Рисунок 2 – Первый этап решения задачи

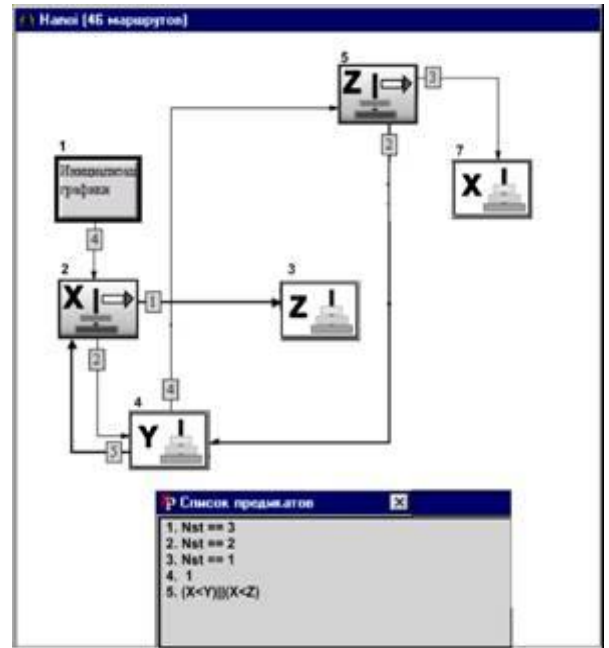


Рисунок 3 – Второй этап решения задачи

Как видно из рисунка, вершина 4 связана с вершинами 5 и 2, т.е. после переноса диска на стержень Y рассматриваются варианты возможности «снятия» дисков со стержней X или Z. В частности, переход в вершину 2 возможен в случае, если со стержня X имеется возможность переноса дисков на другие стержни, что управляется предикатом 5: $(X < Y) || (X < Z)$. Иначе, следующий диск «снимается» со стержня Z.

Очевидно, что, если мы будем снимать диск со стержня Z (вершина 5), то в качестве «операционных» стержней следует рассматривать стержни X и Y (вершины 4 и 7). Продолжая в том же духе, получим окончательный вариант агрегата «Ханойские башни», обеспечивающий решение поставленной задачи (см. рисунок 4).

Технологические акторы 9, 10 и 11 необходимы для отображения результата работы алгоритма и закрытия графического режима работы монитора. Предикат №7 обеспечивает остановку работы алгоритма.

Таким образом, формируя агрегат решения задачи о «Ханойских башнях», мы эволюционно разработали алгоритм её решения, опираясь

фактически только на правила, принятые в рассматриваемой игре, и применяя разумные эвристики.

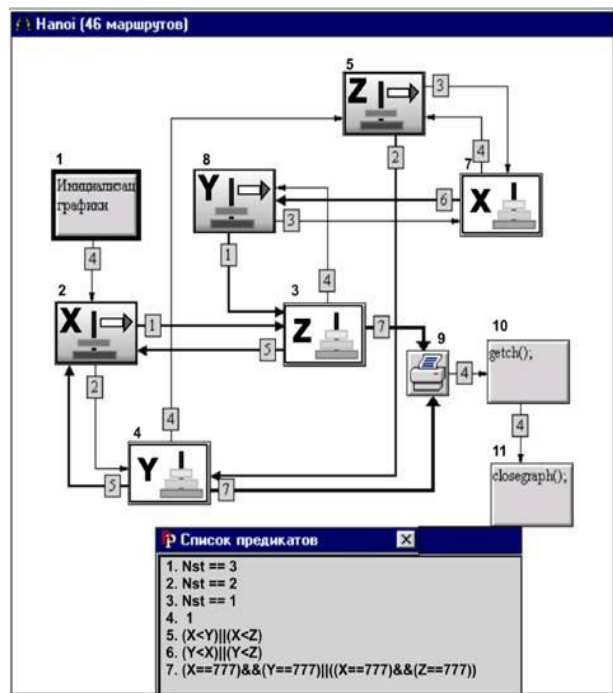


Рисунок 4 – Агрегат алгоритма решения задачи о ханойских башнях

ЗАКЛЮЧЕНИЕ

В работе были рассмотрены основные принципы построения технологии визуального программирования ГСП. Визуальное программирование повышает наглядность, представляемых кодов, существенно уменьшает число ошибок, допускаемых на этапе проектирования и кодирования программ, и тем самым повышает надежность кодов разрабатываемых программ.

На примере решения «творческой» задачи о ханойских башнях продемонстрированы возможности технологии ГСП с точки зрения применения визуальных форм представления модели алгоритма программы. Предлагаемый подход к разработке ПО, по существу, формирует онтологию в данной предметной области.

Рассмотрение технологии ГСП с этих позиций создает новые теоретические основы для дальнейшего развития технологии графосимволического программирования

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Замулин, 1990] Замулин А.В. Системы программирования баз данных и знаний. - Новосибирск: Наука, 1990
- [Коварцев, 1999] Коварцев А.Н. Автоматизация разра-ботки и тестирования программных средств. - Самарский государственный аэрокосмический университет, Самара, 1999. 148 с.
- [Легалов, 2002] Легалов А.И. В лабиринтах Ханойских башен. «Программист», №11, 2002
- [Мейер, 1982] Мейер Б., Бодуэн К. Методы программирования: В 2-х томах. Т.2.- М.: Мир, 1982

[Скобелев, 2011] Скобелев П.О. Онтологии деятельности для ситуационного управления предприятием в реальном времени. «Онтология проектирования», №1(3), 2011. С.6 – 39

[Успенский, 1987] Успенский В.А., Семенов А.Л. Теория алгоритмов: основные открытия и приложения. - М.: Наука, 1987. 288 с.

[Gruber, 1991] Gruber T.R. The role of common ontology in achieving sharable, reusable knowledge bases // Proc. of the 2 International Conf. – 1991. P 601-602

THE BASICS OF GRAPH-SYMBOLIC PROGRAMMING TECHNOLOGY

Kovartsev A.N., Zhidchenko V.V.,
Popova-Kovartseva D.A., Abolmasov P.V.

Samara State Aerospace University named after academician S.P. Korolyov (National Research University), Samara, Russian Federation

kovr_ssau@mail.ru

The article considers Graph-Symbolic Programming technology (GSP-technology), which implements visual programming style. Visual programming makes it easier to represent models of algorithms, significantly reduces the number of errors made during design and writing of programs, and thus improves the reliability of the developed programs.

INTRODUCTION

Modern software is complex. To reduce the complexity of software development and shorten the software development life-cycle the Graph-Symbolic Programming technology provides the means for visual description of algorithms and automatic synthesis of programs based on that description.

MAIN PART

The article introduces major components of GSP-technology:

- basic concepts;
- datatypes, modules and objects;
- interface between objects;
- calculation management;
- ontological aspects of GSP-technology
- example of visual programming in GSP-technology.

CONCLUSION

GSP-technology facilitates the development of numerical software. It provides visual environment not only for algorithm description but also for algorithm construction. In GSP it is possible to construct the algorithm “on the fly” using earlier created objects and creating new ones. The program is synthesized automatically on the basis of the algorithm. As a result development speed and program reliability improve.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИЧЕСКИЕ ЯЗЫКИ ДЛЯ ОПИСАНИЯ ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММ, ОРИЕНТИРОВАННЫХ НА ОБРАБОТКУ ЗНАНИЙ

Пивоварчик О.В.

** Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

pivovarchyk@tut.by

В работе описаны семантические языки представления знаний, используемые для создания баз знаний интеллектуальных help-систем для разработчиков программ, ориентированных на обработку знаний. Семантические языки являются sc-языками и определяются своим расширением множества ключевых узлов sc-кода.

Ключевые слова: интеллектуальная help-система, технология проектирования программ, язык программирования, язык представления знаний, база знаний.

ВВЕДЕНИЕ

Интеллектуальная help-система для разработчиков программ, ориентированных на обработку знаний, представляет собой документацию по соответствующей технологии проектирования программного обеспечения в виде интеллектуальной справочной и обучающей системы [Голенков и др., 2012]. Интеллектуальная help-система обеспечивает всестороннее информационное обслуживание пользователей, обучение пользователей, что приводит к увеличению количества разработчиков интеллектуальных систем и сокращению сроков проектирования интеллектуальных систем.

Для проектирования help-системы используется открытая семантическая технология компонентного проектирования интеллектуальных систем (Open Semantic Technologies for Intelligent System, OSTIS). В основе OSTIS лежит семантическая модель представления знаний, которая использует семантическую сеть с базовой теоретико-множественной интерпретацией и представление знаний в sc-коде. OSTIS предполагает этапы проектирования интеллектуальной системы: проектирование базы знаний, проектирование машины обработки знаний, проектирование пользовательского интерфейса.

Для представления информации в базе знаний help-системы используется универсальный открытый язык представления знаний sc-код. SC-код базируется на аппарате однородных семантических сетей и включает в качестве подязыков

специализированные семантические языки представления и обработки знаний различных предметных областей [Электронный ресурс], [Голенков и др., 2001a]. Подязыки sc-кода (далее – sc-языки) разрабатываются путем выделения необходимого набора ключевых узлов. В работе предлагаются sc-языки для представления знаний технологии проектирования программ, ориентированных на обработку знаний.

Технология разработки программ, ориентированных на обработку знаний, включает следующие компоненты:

- теория программ: язык программирования, язык представления знаний;
- библиотека типовых многократно используемых компонентов (ip-компонентов);
- инструментальное средство проектирования программ;
- методика проектирования программ;
- методика обучения проектированию программ.

Следовательно, основой построения базы знаний интеллектуальной help-системы являются формальные модели соответствующих компонентов: формальная модель языка программирования, формальная модель языка представления знаний, формальная модель инструментального средства, формальная модель библиотеки компонентов, формальная модель методики проектирования программ, формальная модель методики обучения программированию. А также, иерархическая структура разделов, каждый из которых описывает некоторый элемент технологии. Разделы базы

знаний именуется, нумеруются и являются оглавлением.

В соответствии с компонентами технологии выделим следующие sc-языки:

- семантический язык для описания языка программирования и языка представления знаний $L_{program}$;
- семантический язык для описания интегрированной среды разработки программ L_{ide} ;
- семантический язык для описания методики проектирования программ L_{method} ;
- семантический язык для описания методики обучения программированию $L_{training}$.

1. Семантический язык для описания языка программирования и языка представления знаний

Описание формальной модели языка программирования можно задать его спецификацией. Спецификация содержит описание синтаксиса и семантики языка.

Синтаксис определяет множество допустимых конструкций. Существует два способа описания синтаксиса: абстрактный синтаксис и конкретный синтаксис. Абстрактный синтаксис представляет собой абстрактную модель синтаксических конструкций языка, включающую правила построения корректных программных текстов, не зависящую от конкретного вычислителя. Конкретный синтаксис задает запись конструкций языка в конкретной нотации по заданной абстрактной модели. Основными формальными методами определения синтаксиса языка программирования являются форма Бэкуса-Наура (далее – БНФ) и контекстно-свободные грамматики, расширенная форма Бэкуса-Наура, синтаксические графы [Себеста, 2001], [Пратт и др., 2002]. В данной работе за основу взят метод БНФ. Предлагаемый в работе язык позволяет представить в виде семантической сети информацию о правилах БНФ.

Семантика разъясняет смысл синтаксических конструкций. Наиболее распространенными методами описания семантики языков программирования являются: денотационный, операционный, аксиоматический, алгебраический [Вольфенгаген, 2001], [Mitchell, 1996], [Shmidt, 1986]. В данной работе для описания языков программирования будут использоваться денотационный и операционный методы.

Семантический язык $L_{program}$, используемый для описания языков программирования, ориентированных на обработку знаний, задается объединением множеств

$$L_{program} = C_{syntax} \cup S_{syntax} \cup R_{syntax} \cup C_{semantic} \cup S_{semantic} \cup R_{semantic}$$

где C_{syntax} – множество ключевых узлов, предназначенных для описания синтаксиса языка

программирования; S_{syntax} – множество утверждений, предназначенных для описания синтаксиса языка программирования; R_{syntax} – множество отношений, определенных на множестве ключевых узлов C_{syntax} и S_{syntax} ; $C_{semantic}$ – множество ключевых узлов, предназначенных для описания денотационной и операционной семантик языка программирования; $S_{semantic}$ – множество утверждений, предназначенных для описания семантики языка программирования; $R_{semantic}$ – множество отношений, определенных на множестве ключевых узлов $C_{semantic}$ и $S_{semantic}$.

1.1. Описание синтаксиса

Для описания абстрактного синтаксиса C_{syntax} включает множество ключевых узлов, обозначающих типы синтаксических конструкций (синтаксических доменов), S_{syntax} – множество утверждений, описывающих правила построения синтаксических конструкций. Для определения конкретного синтаксиса множество C_{syntax} дополняется ключевыми узлами, необходимыми для задания определенной нотации языка.

Этапы спецификации синтаксиса языка программирования:

- Выделение множества типов синтаксических конструкций.
- Описание множества правил построения синтаксических конструкций, представляемых в виде утверждений следующей формы: *если <часть 1> то <часть 2>*, где *часть 1* содержит знаки типов синтаксических конструкций, *часть 2* содержит выражения, включающие знаки типов синтаксических конструкций, элементарных синтаксических конструкций и операторов конструирования составных синтаксических конструкций.
- Выделение множества типов отношений, заданных на множестве типов синтаксических конструкций.
- Выделение множества типов элементарных синтаксических конструкций, соответствующих задаваемой нотации языка.
- Выделение множества утверждений, необходимых для описания правил построения синтаксических конструкций в задаваемой нотации.

В соответствии со свойствами языков программирования, предназначенных для обработки знаний, были выделены следующие основные типы синтаксических конструкций: программа, оператор, переменная, константа. Операторы являются важным синтаксическим элементом языков программирования, ориентированных на обработку знаний. Операторы задают полное описание действий, которые необходимо выполнить в памяти. Кроме этого, были выделены вспомогательные понятия: тип компонента, тип обрабатываемых данных, тип оператора, атрибут. Введем множество C_{syntax} :

$$C_{syntax} = \{ \text{программа, оператор, переменная,} \}$$

константа, тип оператора, тип компонента, тип обрабатываемых данных, атрибут}.

Семантика ключевых узлов представлена в таблице 1.

Таблица 1 – Семантика ключевых узлов, используемых для описания типов синтаксических конструкций

Ключевой узел	Семантика ключевого узла
программа	знак множества всех синтаксически корректных текстов языка программирования
оператор	знак множества всех операторов, которые могут быть использованы в текстах языка программирования
переменная	знак множества переменных программ
константа	знак множества констант программ
тип оператора	знак множества, элементами которого являются все возможные типы операторов
тип компонента	знак множества, элементами которого являются все возможные типы компонентов, входящих в синтаксические конструкции
тип обрабатываемых данных	знак множества, элементами которого являются все возможные типы данных или структуры данных доступные для обработки синтаксическим элементом
атрибут	знак множества, элементами которого являются все возможные атрибуты синтаксических конструкций и их компонентов

S_{syntax} включает множество логических утверждений, заданных для всех типов синтаксических конструкций. Логические утверждения задают множество правил, которые определяют правильность построения синтаксических конструкций и их иерархию, завершающуюся конструкцией самого верхнего уровня – программой. Знак логического утверждения связывается со знаком типа синтаксической конструкции отношением *Утверждения синтаксиса**. Правильность конструкции проверяется путем подстановки в утверждение значений аргументов.

На множестве ключевых понятий задаются отношения, описывающие общие теоретико-множественные свойства ключевых понятий и связывающие описание синтаксиса языка в формальную модель. Введем множество R_{syntax} , включающее ключевые узлы отношений:

$R_{syntax} = \{\text{синоним}^*, \text{пояснение}^*, \text{разбиение}^*, \text{пример}^*, \text{область значений}^*, \text{обрабатываемые данные}^*, \text{компонент}^*, \text{утверждения синтаксиса}^*\}$.

Семантика ключевых узлов отношений представлена в таблице 2.

Таблица 2 – Семантика ключевых узлов отношений, используемых для описания синтаксиса

Отношение	Семантика отношения
синоним*	связывает ключевое понятие с множеством синонимов
пояснение*	связывает ключевое понятие с толкованием на других языках
разбиение*	связывает ключевое понятие с множеством его непересекающихся подмножеств
пример*	связывает ключевое понятие с множеством примеров
область значений*	связывает ключевое понятие с множеством возможных значений, в случае наличия ограничений
обрабатываемые данные*	связывает ключевое понятие с множеством обрабатываемых типов синтаксических конструкций
компонент*	связывает ключевое понятие с множеством компонентов, являющихся его структурными частями
утверждения синтаксиса*	связывает знак синтаксической конструкции с множеством правил ее построения и накладываемых ограничений

Ключевые узлы, используемые для описания нотации или множества нотаций языка программирования в базе знаний, зависят от конкретного языка и вводятся при реализации help-системы.

1.2. Описание денотационной семантики

Семантика позволяет описать значения всех синтаксически правильных конструкций языка программирования. Денотационный метод описания семантики основывается на выделении функциональных доменов языка, оперирующих состояниями программы, и придания им значений. Множество состояний определяется преобразованиями абстрактной памяти в процессе исполнения синтаксических конструкций. Таким

образом, денотационная семантика языка программирования рассматривается как множество функций определяющих состояние абстрактной памяти после исполнения синтаксических конструкций относительно исходного состояния абстрактной памяти. Денотационная семантика языка программирования задается функцией

$$F : P \rightarrow [S \rightarrow S],$$

где P – синтаксическая область программ, S – семантическая область состояний [Ильичева, 2003].

Языки для описания денотационной семантики тесно связаны с λ -исчислением и в качестве базисных используют операции λ -исчисления [Mitchell, 1996], [Shmidt, 1986], [Ильичева, 2003], [Филд и др., 1993], [Кораблин, 1992]. В данной работе методы денотационной семантики предлагается использовать для отображения синтаксиса языка программирования в теоретико-множественные объекты и объекты логики предикатов первого порядка. Описываемый в работе язык L_{program} использует теоретико-множественное описание данных и соответствующие логико-предикатные механизмы для их обработки. Порядок построения формальной модели денотационной семантики языка программирования заимствован из метода, представленного в работе [Ильичева, 2003]. Метод включает следующие шаги:

- выделение множества синтаксических доменов, характеризующих основные типы синтаксических конструкций;
- выделение множества семантических доменов, описывающих множество значений, которыми оперируют синтаксические конструкции;
- выделение множества семантических функций, которые отображают синтаксические конструкции языка программирования в соответствующие семантические домены;
- выделение множества семантических уравнений, которые отображают изменение состояний шаблонов синтаксических конструкций.

Синтаксическим доменам соответствует множество типов основных синтаксических конструкций языка программирования, выделенных в предыдущем разделе:

- константа – множество всех возможных констант;
- переменная – множество всех возможных переменных;
- оператор – множество всех возможных операторов;
- программа – множество всех возможных программ.

Множество семантических доменов включает множество состояний памяти, являющихся результатом выполнения синтаксических конструкций. Во многих языках программирования состояние памяти фиксируется значением переменных, и таким образом, множество

семантических доменов является множеством возможных значений переменных. Следовательно, простыми семантическими доменами являются:

- значения переменных (*value*);
- отсутствие значений (*undefined*).

На основании простых доменов строятся составные семантические домены:

- абстрактная память (*store*).

Кроме этого, в качестве семантического домена будем использовать ошибку выполнения (*error*), в случае неуспешного завершения выполняемого действия.

Введем множество $C_{\text{semantics}}$:

$C_{\text{semantics}} = \{\text{значение переменной, абстрактная память, отсутствие значения, ошибка выполнения}\}.$

Для каждого синтаксического домена выделяется одна семантическая функция, устанавливающая его смысл. Множество семантических функций описано в таблице 3.

Таблица 3 – Семантические функции

Синтаксический домен	Семантический домен	Функция
константа	значение переменной	<i>value</i>
переменная	значение переменной, отсутствие значения	<i>variable</i> \rightarrow [<i>value</i> \times <i>undefined</i>]
оператор	абстрактная память, значение переменной, отсутствие значения, ошибка	<i>operator</i> \rightarrow [[<i>store</i> \rightarrow <i>variable</i> , <i>store</i>] \times <i>error</i>]
программа	абстрактная память	<i>program</i> \rightarrow [<i>store</i> \rightarrow <i>store</i>]

Для каждого синтаксического шаблона задается семантическое уравнение, которое отражает действие функций. Уравнения задаются логическими средствами. Уравнение (1) определяет смысл константы программы. Смыслом константы является сама константа, изменение состояния памяти не происходит.

$$Eq(\text{константа}) \text{value} = \text{value} \quad (1)$$

Уравнение (2) показывает, что смыслом переменной является отношение значение*, связывающее знак переменной и элемент, являющийся ее значением. В случае отсутствия значения переменной отсутствует отношение значение*.

$$Eq(\text{переменная})value = \langle \text{переменная}, value \rangle \in \text{значение}^* \quad (2)$$

Уравнение (3) задает семантику операторов. Выполнение оператора приводит к изменению значений переменных и изменению состояния памяти, возможно возникновение ситуации, при которой вырабатывается ошибка. Следовательно, смыслом операторов являются: измененные значения переменных и состояние памяти или ошибка выполнения.

$$Eq(\text{оператор})store = \text{для} \forall \text{operator} \text{ если} \langle store \rangle \text{ то} \langle Eq(\text{переменная})value, store_{new} \rangle \quad (3)$$

Смыслом программы является состояние памяти. Программа изменяет состояние памяти, поэтому значением программы является функция от текущего состояния памяти до нового. Семантикой программы также можно назвать конечные значения переменных.

$R_{\text{semantics}}$ включает множество семантических уравнений. Знак семантического уравнения связывается со знаком типа синтаксической конструкции отношением *Утверждения денотационной семантики**. Данное отношение связывает множество синтаксических доменов с семантическими доменами посредством уравнения. Введем множество $R_{\text{semantics}}$:

$$R_{\text{semantics}} = \{ \text{утверждения денотационной семантики}^* \}$$

1.3. Описание операционной семантики

Описание операционной семантики для языков программирования, ориентированных на обработку знаний, представляет собой логическую процедуру построения трансляции программ и отдельных синтаксических конструкций. Трансляция строится на некотором метаязыке, содержащем набор инструкций, являющихся простейшими действиями абстрактной машины. В настоящей работе метаязыком является L_{program} .

Т.к. разрабатываемый язык L_{program} является подязыком sc-кода, то наиболее простым способом описания операционной семантики различных языков, ориентированных на обработку знаний, является переход от способов представления знаний и программ к их представлению в виде семантически эквивалентных sc-конструкций. Описываемый язык приводится к некоторому графовому sc-языку. Такому языку ставится в соответствие своя абстрактная sc-машина. Эквивалентность приведенного языка описана в [Голенков и др., 2001b]. В качестве базовой абстрактной sc-машины предлагается scp-машина. Переход от интерпретируемой абстрактной sc-машины произвольного вида к интерпретирующей scp-машине описан в [Голенков и др., 2001b]. Таким

образом, L_{program} содержит ключевые узлы для описания scp-машины. При реализации help-системы он дополняется необходимыми ключевыми узлами.

Рассмотрим абстрактную scp-машину. Абстрактная scp-машина задается запоминающей средой (sc-памятью), в которой хранятся перерабатываемые данные, и коллективом агентов (операций) над памятью, переводящих ее из состояния в состояние. Задачами памяти являются хранение программ, а также данных, обрабатываемых программами, и предоставление доступа к ним. Ключевыми понятиями для описания sc-памяти являются хранимые данные и операции, через которые описывается поведение scp-машины. Простейшими данными, хранящимися в sc-памяти, являются sc-элементы. Тип sc-элемента задается структурным типом, константностью и позитивностью. Структура данных для хранения sc-элемента зависит от реализации sc-памяти. В данной работе реализация не описывается, поэтому для описания данных в L_{program} включаем только ключевой узел *sc-элемент*.

Агенты содержат описание методов трансляции правильных синтаксических конструкций описываемого языка в последовательность действий абстрактной scp-машины. Процесс выполнения операции описывается понятием scp-процесс. Для описания текущего состояния scp-процесса в L_{program} включаем ключевые узлы: scp-процесс, тип состояния scp-процесса, ошибка. Простейшими операциями scp-машины, в которые отображаются операционная семантика транслируемых синтаксических конструкций, являются:

- Генерация sc-элемента (genEl) – операция генерирует sc-элемент в sc-памяти, тип задается атрибутами. В случае генерации sc-дуги указываются инцидентные элементы. Операции передаются атрибуты, инцидентные элементы (в случае необходимости).
- Удаление sc-элементов (eraseEl) – операция удаления sc-элемента из sc-памяти. Для выполнения операции sc-элемент должен быть определен. Операции передается sc-элемент.
- Поиск sc-элементов или конструкций состоящих из sc-элементов по заданному шаблону (searchEl, searchEls) – операция осуществляет поиск sc-элемента или sc-элементов в sc-памяти по указанным атрибутам. Операции передаются атрибуты.
- Поиск по содержимому sc-элементов (searchContentEl) – операция осуществляет поиск sc-элемента в sc-памяти по указанному содержимому. Операции передаются атрибуты, указывающие тип sc-элемента, и содержимое.
- Преобразование содержимого sc-элементов (convContentEl) – операция осуществляет преобразование содержимого к заданному значению. Для выполнения операции должен быть определен sc-элемент с установленным

содержимым. Операции передается sc-элемент и содержимое.

- Проверка типа sc-элементов (ifTypeEl) – операция проверяет соответствие типа sc-элемента переданным атрибутам. Операции передается sc-элемент и атрибуты.

- Преобразование типа sc-элементов (convTypeEl) – операция преобразовывает sc-элемент к указанному типу. Операции передается sc-элемент и атрибуты.

Дополним множество $C_{\text{semantics}}$:

$$C_{\text{semantics}} = C_{\text{semantics}} \cup \{\text{sc-элемент, scr-процесс, тип состояния scr-процесса, ошибка, генерация sc-элемента, удаление sc-элемента, поиск sc-элемента, поиск sc-элементов по шаблону, поиск по содержимому sc-элемента, преобразование содержимого sc-элемента, проверка типа sc-элемента, преобразование типа sc-элемента}\}.$$

Семантика подмножества ключевых узлов $C_{\text{semantics}}$ представлена в таблице 4.

Таблица 4 – Семантика ключевых узлов, используемых для описания операционной семантики

Ключевой узел	Семантика ключевого узла
sc-элемент	знак множества всех элементов, хранящихся в sc-памяти
scr-процесс	знак множества всех scr-процессов, каждый из которых выполняется одной из операций scr-машины
тип состояния scr-процесса	знак множества типов состояний scr-процесса
ошибка	знак множества ошибок, возникающих при выполнении sc-процесса
операция	знак множества простейших операций, вызываемых при выполнении синтаксических конструкций
генерация sc-элемента	знак множества операций генерации sc-элемента
удаление sc-элемента	знак множества операций удаления sc-элемента
поиск sc-элемента	знак множества операций поиска sc-элемента
поиск sc-элементов по шаблону	знак множества операций поиска sc-элементов по шаблону
поиск по содержимому sc-элемента	знак множества операций поиска по содержимому sc-элемента
преобразование	знак множества операций преобразования содержимого sc-

содержимого sc-элемента	элемента
проверка типа sc-элемента	знак множества операций проверки типа sc-элемента
преобразование типа sc-элемента	знак множества операций преобразования типа sc-элемента

Каждая операция характеризуется именем, условием инициирования, входными/выходными данными, трансляцией. Дополним множество $R_{\text{semantics}}$:

$$R_{\text{semantics}} = R_{\text{semantics}} \cup \{\text{условие инициирования}^*, \text{входные данные}^*, \text{выходные данные}^*, \text{трансляция}^*\}.$$

Семантика подмножества ключевых узлов отношений $R_{\text{semantics}}$ представлена в таблице 5.

Таблица 5 – Семантика ключевых узлов отношений, используемых для описания операционной семантики

Отношение	Семантика отношения
условия инициирования*	связывает операцию с шаблоном ее инициирования
входные данные*	связывает операцию с множеством sc-элементов, являющимися входными данными операции
выходные данные*	связывает операцию с множеством sc-элементов, являющимися выходными данными операции
трансляция*	связывает операцию с последовательностью простейших операций, отображающих алгоритм ее выполнения

Таким образом, для описания операционной семантики требуется выделение ключевых узлов, обозначающих типы синтаксических конструкций. Необходимые ключевые узлы были включены ранее. Требуются ключевые узлы, описывающие переход от используемых способов представления знаний и программ к их представлению в виде семантически эквивалентных sc-конструкций. Требуемые ключевые узлы определяются при реализации help-системы и зависят от описываемого языка программирования. Требуются ключевые узлы для описания scr-машины.

2. Семантический язык для описания интегрированной среды разработки программ

Интегрированная среда разработки программного обеспечения состоит из

инструментов, обеспечивающих разработку программ и среду существования программ, а также вспомогательных средств, используемых для управления разработкой программ. Обычно она ориентирована на определенный язык программирования или языки программирования одной методологии. Базовыми функциями сред являются: управление кодом программ или проектов, хранение версий проектов и его различных компонентов, сборка проектов, подключение и настройка транслятора, отладка кода, получение справочной информации. Обзор сред, используемых для проектирования программ, ориентированных на обработку знаний, показал, что они обладают свойствами и функциональностью сред для проектирования программ на традиционных языках.

Для формального описания интегрированной среды предлагается построить ее компонентную модель. Каждый компонент определяется своей функциональностью и может включаться в состав среды или отсутствовать. Компонент представляется своей формальной моделью. Таким образом, формальной моделью интегрированной среды разработки программного обеспечения является ее компонентная модель и совокупность формальных моделей каждого компонента. Формальная модель среды описывается на семантическом языке L_{ide} . L_{ide} задается объединением множеств

$$L_{ide} = C_{ide} \cup S_{ide} \cup R_{ide},$$

где C_{ide} – множество ключевых узлов, предназначенных для описания формальных моделей компонентов интегрированных сред; S_{ide} – множество утверждений, предназначенных для описания компонентов интегрированных сред; R_{ide} – множество отношений, определенных на множествах C_{ide} и R_{ide} .

На основании базовых функций выделим основные компоненты, включаемые в интегрированные среды, и введем множество C_{ide} :

$C_{ide} = \{\text{редактор исходного кода, транслятор, компоновщик, отладчик, интерфейс, справочная система}\}$.

Рассмотрим формальные модели каждого компонента и дополним язык C_{ide} .

Редактор предназначен для управления текстами программ или проектов. Редактор может поддерживать один или несколько языков программирования, обнаруживать синтаксические ошибки пользователя по мере ввода текста, показывать всплывающие подсказки, осуществлять дополнение кода. Среда программирования включает один или несколько редакторов для объектов различного вида (например, различные нотации языка могут редактироваться в различных редакторах). При использовании линейной записи текстов языка редактор обеспечивает следующие

функции: подсветка синтаксиса, автоматическое указание ошибок, форматирование текста, загрузка шаблонов синтаксических конструкций языка, автоматическое дополнение кода, всплывающие подсказки при наведении на синтаксический элемент (помощник при работе с содержимым исходного файла). При использовании графической записи текстов языка редактор обеспечивает следующие функции: подсветка синтаксиса, загрузка шаблонов синтаксических конструкций языка, всплывающие подсказки при наведении на синтаксический элемент. Дополним язык C_{ide} :

$C_{ide} = C_{ide} \cup \{\text{язык программирования, нотация языка программирования, подсветка синтаксиса, автоматическое указание ошибок, форматирование текста, загрузка шаблона, дополнение кода, всплывающие подсказки}\}$.

Транслятор выполняет преобразование программ с одного языка в эквивалентные им программы на другом языке. В случае языка программирования – на язык понятный компьютеру. Реализация транслятора не является задачей данной работы, поэтому ограничимся описанием операционной семантики языка программирования, представленной ранее. Среда включает только функции, используемые для запуска или остановки работы транслятора. Дополним язык C_{ide} :

$C_{ide} = C_{ide} \cup \{\text{запуск процесса, завершение процесса}\}$.

Компоновщик формирует проект, собирая необходимые файлы-компоненты (программы, библиотеки, файлы данных и др.) и редактируя перекрестные ссылки. Среда вызывает компоновщик при сборке проекта, при запуске программ на выполнение, при обновлении ресурсов и сохранении измененных ресурсов на диск. Дополним язык C_{ide} :

$C_{ide} = C_{ide} \cup \{\text{библиотека, файл данных, ресурс}\}$.

Отладчик используется для обнаружения ошибок в программе или причин их появления при пошаговом просмотре результатов ее выполнения. Для описания отладчика необходимо представить модель процесса трансляции программы и дополнить ее параметрами отладки. Основными функциями отладчика являются:

- управление выполняемым процессом: запуск, завершение, выполнение в пошаговом режиме, приостановка, возобновление, создание контрольных точек останова;
- модификация памяти: изменение значений переменных, системных структур данных;
- представление пользователю информации о состоянии памяти и сведений о ходе выполнения процесса.

Реализация отладчика зависит от каждого конкретного языка и от средств, используемых для его реализации. Традиционно выполняемому процессу ставится в соответствие микропрограмма,

которая получает сведения о его выполнении и позволяет его выполнять в пошаговом режиме. Далее микропрограмму в работе будем называть трассировщиком. Трассировщик запускает процесс или присоединяется к существующему процессу и имеет доступ ко всем данным процесса и параметрам отладки, что позволяет ему выполнять все вышеперечисленные функции. Пошаговый режим отладки выполняется в соответствии с управляемой пользователем спецификацией шагов: выполнить одну инструкцию с заходом в функцию, выполнить одну инструкцию без захода в функцию, выполнить выход из функции, продолжить выполнение программы до следующей точки останова, продолжить выполнение программы до курсора, продолжить выполнение программы до выполнения заданного условия. Контрольные точки останова загружаются в память вместе с текстом программы и связываются с синтаксической конструкцией. Контрольные точки представляются в виде структуры данных, содержащей всю необходимую информацию. Например, структура данных может содержать: тип точки останова, связь со строкой (в случае линейной записи программ) или связь с ключевым узлом (в случае графического представления программ), маркер ресурсов для сохранения информации о контрольных точках между сессиями и др.

Отображение текущего состояния памяти обеспечивается компонентами пользовательского интерфейса, настраиваемыми пользователем. Возможно отображение содержимого областей памяти, значений переменных, стека вызова процедур. Дополним язык C_{ide} :

$C_{ide} = C_{ide} \cup \{\text{процесс, проект, программа, память, трассировщик, режим отладки, точка останова, тип точки останова}\}.$

Интерфейс включает настройки для различных режимов работы. Например, наиболее часто встречаются следующие режимы работы: режим разработки программ, режим отладки, режим выполнения. Настройки включают конфигурирование вида и расположения используемых инструментов и пунктов меню, настройку внешнего вида иконок и количества иконок при различных режимах работы, настройки используемого транслятора и настройки редактора кода. Интерфейс обеспечивает работу мастеров при создании проектов или отдельных файлов. Дополним язык C_{ide} ключевыми узлами, обеспечивающими описание интерфейса:

$C_{ide} = C_{ide} \cup \{\text{режим разработки, режим отладки, режим выполнения, вид инструментального средства, проект, файл, мастер создания проекта}\}.$

Справочная система представляет собой консультационную программную систему по технологии разработки программного обеспечения. Существующие инструментальные средства включают различные виды справочных систем. Поэтому в данной работе не предлагается единая

модель для описания используемых справочных систем, а предлагается модель их построения. В качестве справочной системы может использоваться предлагаемая в работе help-система, либо ее справочная подсистема.

S_{ide} включает множество логических утверждений, описывающих особенности работы инструментального средства.

R_{ide} включает множество отношений, описывающих общие теоретико-множественные свойства ключевых узлов и связывающих ключевые понятия множеств C_{ide} и S_{ide} в единую формальную систему. Введем множество R_{ide} :

$R_{ide} = \{\text{синоним*}, \text{пояснение*}, \text{разбиение*}, \text{компонент*}, \text{утверждения*}, \text{поддерживаемые нотации языка*}, \text{шаблон*}, \text{используемый транслятор*}, \text{выполняемые действия*}\}.$

Семантика подмножества ключевых узлов отношений R_{ide} представлена в таблице 6.

Таблица 6 – Семантика ключевых узлов отношений, используемых для описания интегрированной среды разработки программ

Отношение	Семантика отношения
синоним*	связывает ключевое понятие с множеством синонимов
пояснение*	связывает ключевое понятие с толкованием на других языках
разбиение*	связывает ключевое понятие с множеством его подмножеств
компонент*	связывает ключевое понятие с множеством его компонентов, файлов-компонентов
утверждения*	связывает ключевое понятие с множеством утверждений
поддерживаемые нотации языка*	связывает ключевое понятие редактора исходного кода с множеством поддерживаемых нотаций языка (языков) программирования
шаблон*	связывает ключевое понятие с множеством шаблонов, доступных для дополнения
используемый транслятор*	связывает ключевое понятие (знак нотации языка программирования) со знаком используемого транслятора или множеством знаков используемых трансляторов
выполняемые действия*	связывает ключевое понятие с действием, которое выполняется при его активации в инструментальной среде разработки программ

3. Семантический язык для описания методики проектирования программ, ориентированных на обработку знаний

Проектирование программ требует специальных методов. Существует два основных класса методов, используемых как для разработки программ на традиционных языках, так и для разработки на языках, ориентированных на обработку знаний. Классическим методом разработки программ является метод водопада, который включает определение требований, проектирование, кодирование, интеграцию, тестирование, внедрение и поддержку. Наиболее популярным является итеративный метод. Итеративный метод представляет собой многократный проход этапов разработки с запланированным улучшением результата. Каждый этап, называемый итерацией, является мини-проектом фиксированной длительности разрабатываемой системы. Существует множество примеров использования данного метода: быстрая разработка приложений (Rapid application development, RAD), экстремальное программирование (Extreme programming, XP), Microsoft Solution Framework (MSF), Rational Unified Process (RUP), Dynamic Systems Development Method (DSDM), SADT (IDEF x), Scrum и др.

На основании анализа методов выделены их общие свойства. В работе представлена обобщенная формальная модель некоторых методов проектирования программ и формальная модель эволюционной методики проектирования интеллектуальных систем, которая описана в [Электронный ресурс], [Гулякина и др., 2011], [Голенков и др., 2012]. Методы тесно связаны со средствами проектирования программ, поэтому при их описании следует учитывать перспективы развития средств и их влияние на развитие методов. Для описания методики проектирования программ в работе предлагается семантический язык

$$L_{method} = C_{method} \cup R_{method},$$

где C_{method} – множество ключевых узлов, предназначенных для описания методики проектирования программ; R_{method} – множество отношений, определенных на множестве ключевых узлов.

На основании анализа используемых методов выделим общие ключевые понятия для их описания и введем множество C_{method} :

$C_{method} = \{\text{методология, метод, этапы проектирования, требования, проектирование, кодирование, интеграция, тестирование, внедрение, поддержка, прототип, инструментальное средство, язык программирования, язык представления знаний}\}.$

Формальная модель эволюционной методики проектирования интеллектуальных систем включает методику проектирования баз знаний, методику проектирования машины обработки знаний,

методику проектирования пользовательских интерфейсов, методику проектирования интеллектуальных решателей задач. Дополним множество C_{method} :

$C_{method} = C_{method} \cup \{\text{методика проектирования базы знаний, методика проектирования операций, методика проектирования пользовательского интерфейса, методика проектирования интеллектуального решателя задач}\}.$

Каждая методика применяется на одном из этапов построения интеллектуальной системы. На первом этапе проектируется база знаний интеллектуальной системы. Методика построения базы знаний описана в [Электронный ресурс]. Она включает следующие шаги:

1. Разработка тестового сборника вопросов (спецификации вопросов).
2. Запись ответов на вопросы в текстовом и формальном виде.
3. Тестирование стартовой версии базы знаний.
4. Выделение набора объектов, входящих в состав базы знаний: классов объектов, отношений, связей, набора основных логических высказываний (утверждений) для описания свойств объектов.
5. Представление выделенных понятий на формальном языке.
6. Верификация базы знаний.
7. Анализ и уточнение распределения понятий по логическим уровням.
8. Логико-дидактическая структуризация базы знаний: декомпозиция и упорядочение выделенных разделов.
9. Анализ полноты базы знаний.
10. Тестирование полученной версии базы знаний.

В результате получаем первый прототип интеллектуальной системы, который включает разработанную базу знаний, базовую информационно-поисковую машину и базовый пользовательский интерфейс. Для описания формальной модели методики проектирования баз знаний дополним множество C_{method} множеством шагов построения базы знаний:

$C_{method} = C_{method} \cup \{\text{спецификация вопросов, ответы на вопросы, тестирование стартовой версии, выделение классов объектов, выделение отношений, выделение связей отношений, выделение утверждений, представление понятий на формальном языке, верификация базы знаний, распределение понятий по логическим уровням, структуризация базы знаний, анализ полноты базы знаний, тестирование базы знаний, базовая информационно-поисковая машина, базовый пользовательский интерфейс}\}.$

На втором этапе интеллектуальная система дополняется множеством специализированных операций машины обработки знаний. Методика проектирования операций:

1. Выделение множества операций (на основании анализа спецификации вопросов).
2. Спецификация каждой операции на формальном языке.
3. Реализация операций на языке программирования.
4. Отладка операций.
5. Тестирование полученной версии машины обработки знаний.

В результате получаем второй прототип интеллектуальной системы, который включает разработанную базу знаний, специализированную машину обработки знаний и базовый пользовательский интерфейс. Для описания формальной модели машины обработки знаний дополним множество C_{method} :

$$C_{method} = C_{method} \cup \{\text{операция, спецификация операции, отладка операции, тестирование машины обработки знаний}\}.$$

На третьем этапе проектируется предметно-ориентированный пользовательский интерфейс. Методика проектирования пользовательского интерфейса описана в работе [Корончик, 2012]. Процесс проектирования включает:

1. Спецификацию интерфейса: список решаемых задач, описание внешних языков представления знаний.
2. Создание задачно-ориентированной декомпозиции пользовательского интерфейса.
3. Разработка компонентов, каждый из которых является подсистемой.
4. Тестирование пользовательского интерфейса.

Третий прототип интеллектуальной системы является реализованной системой, удовлетворяющей требованиям разработки. Дополним множество C_{method} :

$$C_{method} = C_{method} \cup \{\text{интерфейс, спецификация интерфейса, декомпозиция интерфейса, тестирование интерфейса}\}.$$

На четвертом этапе интеллектуальная система дополняется решателем задач, который осуществляет генерацию ответов на заданные пользователем вопросы в случае, если ответы отсутствуют в текущем состоянии базы знаний. Методика проектирования описана в [Заливако и др., 2012]. Компонентами решателя задач являются операции, поэтому методика проектирования сводится к методике проектирования операций.

4. Семантический язык для описания методики обучения проектированию программ

Интеллектуальная help-система может использоваться в качестве компьютерного средства обучения (далее – КСО). В [Башмаков и др., 2003] выделены основные педагогические задачи, которые должны решать КСО:

- начальное ознакомление с предметной областью, освоение базовых понятий и концепций;
- базовая подготовка на разных уровнях глубины и детальности;
- выработка умений и навыков решения типовых практических задач;
- выработка умений анализа и принятия решений в нестандартных проблемных ситуациях;
- развитие способностей к определенным видам деятельности;
- восстановление знаний, умений и навыков (для редко встречающихся ситуаций, задач и технологических операций);
- контроль и оценивание уровней знаний и умений.

Для решения вышеперечисленных задач база знаний интеллектуальной help-системы должна содержать формальные модели методики обучения проектированию программ и стратегии обучения программированию. Следовательно, семантический язык описания методики обучения программированию $L_{training}$ является объединением множеств

$$L_{training} = C_{training} \cup R_{training},$$

где $C_{training}$ – множество ключевых узлов, предназначенных для описания формальных моделей методики обучения проектированию программ и стратегии обучения; $R_{training}$ – множество отношений, определенных на множестве ключевых узлов.

Рассмотрим методику обучения как совокупность следующих взаимосвязанных иерархических компонентов: целевой, содержательный, деятельностный, результативный, управленческий. Введем множество $C_{training}$:

$$C_{training} = \{\text{целевой компонент, содержательный компонент, деятельностный компонент, результативный компонент, управленческий компонент}\}.$$

Целевой компонент содержит формулировку целей и задач обучения. Дополним множество $C_{training}$:

$$C_{training} = C_{training} \cup \{\text{цель обучения, задача обучения}\}.$$

Содержательный компонент включает содержание технологии и теоретическую часть, позволяющую предоставить пользователю целостную картину процесса проектирования программ. Теоретическая часть представляется в виде онтологии технологии. Так как в данной работе рассматривается технология проектирования интеллектуальных систем, то содержательный компонент дополняется теорией для осуществления предварительного этапа обучения. Предварительный этап формирует у разработчиков понимание отличия традиционного программирования от программирования

интеллектуальных систем. Приведем описание на примере разработки теоретико-графовых задач. Для этого в базу знаний включим формализацию теоретико-графовых алгоритмов и запись их в виде программ на различных языках. Например:

- на процедурном языке программирования высокого уровня, ориентированном на обработку семантических сетей;
- на базовом языке программирования, ориентированном на обработку семантических сетей;
- на традиционном языке программирования с использованием специализированной библиотеки моделирования графодинамической памяти.

Далее теоретическая часть содержит описание эволюционных этапов проектирования интеллектуальных систем. Дополним множество C_{training} :

$C_{\text{training}} = C_{\text{training}} \cup \{\text{технология программирования, язык программирования, алгоритм, этап проектирования интеллектуальной системы}\}.$

Деятельностный компонент направлен на формирование у пользователя практических навыков в области разработки интеллектуальных систем и приобретения опыта применения теоретических знаний при проектировании программ. Компонент может осуществлять решение задач и объяснять способы их решения. Дополним множество C_{training} :

$C_{\text{training}} = C_{\text{training}} \cup \{\text{задача, способы решения задачи}\}.$

Результативный компонент обеспечивает генерацию тестовых и практических заданий, проведение тестирования и интерпретацию результатов. Компонент может проводить оценку знаний и умений пользователя. Дополним множество C_{training} :

$C_{\text{training}} = C_{\text{training}} \cup \{\text{тестовое задание, тестирование, формы проведения тестирования}\}.$

Управленческий компонент несет организационную, обучающую и контролирующую функции, организует адаптивный диалог с пользователем. Основной задачей компонента является обеспечение пользователю помощи, соответствующей ситуации, и выдача рекомендаций с учетом истории его взаимодействия с системой, т.е. осуществление поддержки принятия решений при разработке программного обеспечения для конкретного разработчика программ. Управленческий компонент определяет стратегию обучения для минимизации времени обучения. Дополним множество C_{training} :

$C_{\text{training}} = C_{\text{training}} \cup \{\text{стратегия обучения}\}.$

Наряду с иерархией понятий технологии программирования используются онтологии знаний, умений, навыков пользователей, которые

представлены в формальной модели разработчика программ. Формальная модель задается в виде шаблона, который наполняется в процессе работы пользователя с системой и фиксируется для каждого конкретного пользователя. Дополним множество C_{training} :

$C_{\text{training}} = C_{\text{training}} \cup \{\text{модель пользователя}\}.$

Кроме этого, в связи с высокими темпами обновления технологий разработки программ, появления новых языков программирования, развития новых инструментальных средств разработки программ меняются и подходы к методике обучения проектированию программ. Это необходимо учитывать при проектировании help-системы.

ЗАКЛЮЧЕНИЕ

В работе представлены семантические языки представления знаний, являющиеся подязыками sc-кода. Семантические языки предоставляют средства для описания технологии разработки программ, ориентированных на обработку знаний, в базе знаний интеллектуальной help-системы. Предлагаемые языки позволяют явно представить семантическую структуру предметной области, что облегчает реализацию семантической навигации по всей структуре информации, а также ассоциативного поиска информации. Также язык позволяет строить спецификацию технологии, используемую для формирования ответов на вопросы пользователя, для обучения пользователя, контроля знаний и умений пользователя. Построенная спецификация представляет собой модель транслятора языка программирования, инструментального средства, и может использоваться для проектирования и реализации данных компонентов технологии.

Преимуществом семантических языков является предоставление возможности интеграции различных компонентов, построенных с их помощью и с помощью других sc-языков. Благодаря чему можно объединять компоненты, содержащие знания по различным технологиям проектирования программного обеспечения, в одну help-систему.

Предлагаемые языки являются легко расширяемыми, поскольку их всегда можно пополнить множеством требуемых ключевых узлов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Голенков и др., 2012] Голенков, В.В. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования / В.В.Голенков, Н.А.Гулякина // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск: БГУИР, 2012. – с. 23-52

[Электронный ресурс] Открытая семантическая технология компонентного проектирования интеллектуальных систем [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://ostis.net/mediawiki/index.php/>.

[Голенков и др., 2001a] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных

машинах. Монография / В.В. Голенков, О.Е. Елисеева, В.П. Ивашенко и др. Под ред. В.В. Голенкова. – Мн.: БГУИР, 2001. – 412 с.

[Себеста, 2001] Себеста, Роберт У. Основные концепции языков программирования, 5-е изд.: Пер. с англ. / Роберт У. Себеста. – М.: Издательский дом «Вильямс», 2001. – 672 с.

[Пратт и др., 2002] Пратт, Т. Языки программирования: разработка и реализация. 4-е изд.: Пер. с англ. / Т.Пратт., М.Зелковиц. Под общей ред. А.Матросова. – СПб.: Питер, 2002. – 688 с.

[Вольфенгаген, 2001] Вольфенгаген, В.Э. Конструкции языков программирования. Приемы описания / В.Э. Вольфенгаген. – М.: АО «Центр ЮрИнфоР», 2001. – 276 с.

[Mitchell, 1996] Mitchell, John C. Foundations for programming languages / John C. Mitchell. – The MIT Press, 1996. – 845 с.

[Shmidt, 1986] Shmidt, David A. Denotational semantics: a methodology for language development / David A. Shmidt. – Allyn and Bacon, 1986. – 331 с.

[Ильичева, 2003] Ильичева, О.А. Формальное описание семантики языков программирования. Электронный учебник / О.А. Ильичева. – Ростов на Дону: ЮФУ, 2007. – 223 с.

[Филд и др., 1993] Филд, А. Функциональное программирование: Пер. с англ. / А.Филд, П. Харрисон. – М.: Мир, 1993. – 637 с.

[Кораблин, 1992] Кораблин, Ю.П. Семантика языков программирования: учебное пособие / Ю.П. Кораблин. Под ред. – М.: МЭИ, 1992. – 102 с.

[Голенков и др., 2001b] Голенков, В.В. Программирование в ассоциативных машинах / В.В.Голенков, Г.С.Осипов, Н.А.Гулякина и др. – Мн.: БГУИР, 2001. – 276 с.

[Гулякина и др., 2011] Гулякина, Н.А. Комплексная методика проектирования и обучения проектированию интеллектуальных справочных систем / Н.А.Гулякина, О.В.Пивоварчик // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». – Минск: БГУИР, 2011. – с. 519 – 522

[Голенков и др., 2012] Голенков, В.В. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования / В.В.Голенков, Н.А.Гулякина // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». – Минск: БГУИР, 2012. – с. 23 – 52

[Корончик, 2012] Корончик, Д. Н. Семантические модели мультимодальных пользовательских интерфейсов и семантическая технология их проектирования / Д.Н.Корончик // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». – Минск: БГУИР, 2012. – с. 339 – 346

[Заливако и др., 2012] Заливако, С. С. Семантическая технология компонентного проектирования интеллектуальных решателей задач / С.С.Заливако, Д.В.Шункевич // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». – Минск: БГУИР, 2012. – с. 297 – 314

[Башмаков и др., 2003] Башмаков, А. И. Разработка компьютерных учебников и обучающих систем / А. И. Башмаков, И. А. Башмаков — М.: Информационно-издательский дом «Филинь», 2003.

SEMANTIC LANGUAGES USED FOR DESCRIPTION OF LANGUAGES PROGRAMMING, ORIENTED TO KNOWLEDGE TREATMENT

Pivovarchyk O.V.

**Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

pivovarchyk@tut.by

The knowledge representation semantic languages is described in the article. The semantic languages are used to develop knowledge base of help-system for software developers. The semantic languages are sc-

languages and have their extension in the set of key nodes.

INTRODUCTION

It is necessary to create semantic languages for describing a technology software engineering to build the intellectual help-system knowledge base. The semantic language for description of programming language (L_{program}), the semantic language for description of integrated development environment (L_{ide}), the semantic language for description of methodology software engineering (L_{method}), the semantic language for description of methodology of training of software engineering (L_{training}) are proposed in this article.

MAIN PART

The programming language formal model are presented its specification, which contains language syntax and semantics. L_{program} includes following sets: the set of key nodes and the set of statement for description of programming language syntax, denotational semantics and operational semantics; the set of relation for union the sets of key nodes in the integrated formal system.

In the knowledge base of the integrated development environment componet models are presented. L_{ide} includes following sets: the set of key nodes and the set of statement for description of formal models of integrated development environment components; the set of relation for union the sets of key nodes in the integrated formal system.

L_{method} allows to describe the generalize model of software engineering methods and the model of intellectual systems engineering evolutionary methodology. L_{method} includes following sets: the set of key nodes for description of software engineering methodology; the set of relation for union the set of key nodes in the integrated formal system.

In the knowledge base the methodology of training of software engineering are presented formal models of training methods and formal models of training strategies. Consequently, L_{training} includes following sets: the set of key nodes for description the formal models of methods of software engineering training and the training strategies; the set of relation for union the set of key nodes in the integrated formal system.

CONCLUSION

The knowledge representation semantic languages allow to design the technology software engineering semantic model in the knowledge base. The semantic model is based on meaning links. This facilitates implementation of semantic navigation, of associative search, of the formation answer to the user query.

The knowledge representation semantic languages are scalable.

СЕКЦИЯ 4.

**СЕМАНТИЧЕСКИЕ МОДЕЛИ, СРЕДСТВА И МЕТОДЫ
КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ РЕШАТЕЛЕЙ ЗАДАЧ**

SECTION 4.

**SEMANTIC MODELS, TOOLS AND METHODS OF COMPONENT
DESIGN PROBLEMS SOLVERS**



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОНТОЛОГИЧЕСКИ-ОРИЕНТИРОВАННАЯ СИСТЕМА КЛАСТЕРИЗАЦИИ И ПОЛНОТЕКСТОВОГО ПОИСКА ПРОЕКТНЫХ ДОКУМЕНТОВ

Наместников А.М. *, Субхангулов Р.А. *, Филиппов А.А. *

* *Ульяновский государственный технический университет,
г. Ульяновск, Россия*

nam@ulstu.ru

subkhangulov-ruslan@yandex.ru

al.filippov@ulstu.ru

В статье представлено описание программной системы, которая позволяет выполнять кластеризацию проектных текстовых документов, основываясь на текущем состоянии предметной области в виде прикладной онтологии. Показана архитектура системы, структура онтологии и ее фрагменты в формате RDF, перечислены основные функции системы. Приведены результаты экспериментальных исследований, доказывающие эффективность разработанной программной системы.

Ключевые слова: интеллектуальная система; онтология; индексирование; кластеризация; полнотекстовый поиск.

ВВЕДЕНИЕ

В настоящее время во многих проектных организациях фактически завершен перевод архива проектной документации (ПД) в электронный формат. В связи с этим возникла необходимость в систематизации и автоматизации работы с электронным архивом ПД. Формирование к электронному архиву запросов допускается с использованием формализованного языка (такого, например, как SQL) при заранее известных атрибутах: децимальный номер, дата создания документа, автор и т.п. При таком подходе к построению архива ПД у проектировщика отсутствует возможность решать слабоформализованные задачи поиска. Такими задачами могут быть полнотекстовый поиск документа, нахождение близкого по содержанию документа, кластеризация всего множества документов и другие. Для решения подобного рода задач применяются интеллектуальные системы, функционирование которых основано на предметно-ориентированных знаниях. Эти знания могут быть представлены в виде онтологии предметной области [Добров и др., 2006].

В данной статье представлено описание программной системы, основными функциями которой является построение с использованием модели разметки Resource Description Framework (RDF) предметно-ориентированной онтологии,

индексирование и кластеризация ПД на основе созданного онтологического описания предметной области и полнотекстовый информационный поиск.

1. Архитектура системы и модель онтологии

Система кластеризации и поиска ПД обладает следующими функциями:

- хранение и обработка документов в формате XML;
- создание и редактирование онтологии на основе моделей RDF и RDFS;
- онтологически-ориентированное индексирование ПД;
- онтологическая кластеризация документов;
- онтологический полнотекстовый поиск документов.

На рисунке 1 представлена архитектура разработанной системы. В качестве хранилища XML-документов используется XML-ориентированная СУБД Tamino, а в качестве хранилища онтологий Web-фреймворк Sesame.

Формально онтологию представим следующим образом:

$$O = \langle r, T, S, C, W, R \rangle, \quad (1)$$

где r – корневая вершина онтологии, соответствующая классу проектных документов;

$T = \{t_1, t_2, \dots, t_n\}$ – множество типов проектных документов ИПР, t_i – i -й тип проектного документа;

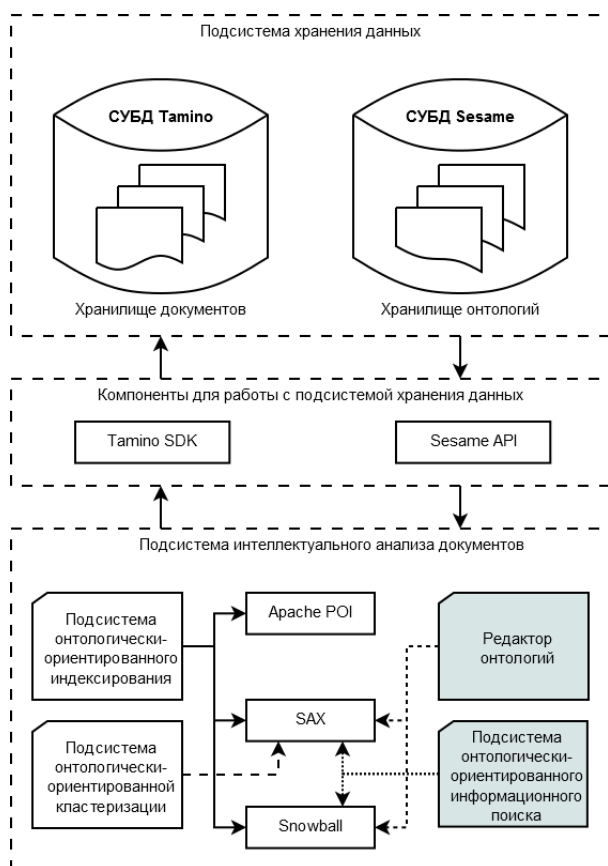


Рисунок 1 – Архитектура системы кластеризации и поиска ПД

$S = S^1 \cup S^2 \cup \dots \cup S^n$ – множество структур документов;

$C = \{c_1, c_2, \dots, c_k\}$ – множество понятий предметной области ИПР;

$W = \{w_1, w_2, \dots, w_l\}$ – множество терминов предметной области ИПР.

R – множество отношений, определяемое следующим образом:

$$R = R_G \cup R_C \cup R_A,$$

где R_G – антисимметричное, транзитивное, нереплексивное бинарное отношение обобщения;

R_C – бинарное транзитивное отношение композиции («часть–целое»);

R_A – конечное множество ассоциативных отношений.

Для структур документов справедливо соотношение:

$$\forall t_i \exists S^j : i = j.$$

Другими словами, для каждого типа документа в онтологии определена его структура.

Множество S^j содержит разделы и подразделы

проектного документа типа t_j . В общем случае имеет место следующее неравенство:

$$S^i \cap S^j \neq \emptyset,$$

что означает допустимость пересечения структурных элементов между различными типами документов.

Онтология предметной области включает в себя два уровня: концептуальный и терминологический [Добров и др., 2006]. Концептуальный уровень представляется в виде дерева

$$(C, E),$$

где $C = \{c\}$ – множество концептов (понятий) предметной области, зафиксированных в онтологии;

$E = \{<c_i, c_k>, <c_i, c_k> \in C^2\}$ – множество дуг, соединяющих понятия.

Терминологический уровень для k -го понятия записывается в виде множества

$$\{(w_1^k, f_1^k), (w_2^k, f_2^k), \dots, (w_i^k, f_i^k), \dots, (w_l^k, f_l^k)\},$$

где w_i^k – i -й терм k -го понятия онтологии;

l_k – общее количество термов, ассоциированных с k -м понятием;

f_i^k – частота встречаемости i -го термина в описании k -го понятия.

2. Функциональные возможности системы

В редакторе онтологии реализованы следующие модули системы:

- Модуль формирования схемы онтологии и модуль формирования содержания онтологии предоставляют пользователю интерфейс для создания схемы онтологии и набора экземпляров классов онтологии.
- Модуль формирования терминологического окружения понятий онтологии реализована в модуле онтологической индексации.
- Модуль взаимодействия с RDF-хранилищем Sesame, где реализованы функции, позволяющие сохранять онтологию в хранилище Sesame и загружать онтологию из данного хранилища.
- Модуль визуализации онтологии позволяет производить редактирование создаваемой онтологии, представляя ее в виде графа, внешний вид которого можно настраивать (форму узлов, дуг и цветовую гамму).

Фрагмент созданной онтологии предметной области «Проектирование информационных систем» для интеллектуального анализа ПД имеет следующий вид:

<Conceptrdf:ID="Субъект"/>

```

<Conceptrdf:ID="Эксперт">
<Generalization rdf:resource="#Субъект"/>
</Concept>
<Concept rdf:ID="Проектировщик">
<Generalization rdf:resource="#Субъект"/>
</Concept>
<Concept rdf:ID="Тестировщик">
<Generalization rdf:resource="#Субъект"/>
</Concept>
<Concept rdf:ID="Программист">
<Generalization rdf:resource="#Субъект"/>
</Concept>
<Conceptrdf:ID="Объект"/>.

```

Модуль визуализации онтологии предназначен для получения изображения онтологии в виде ориентированного графа [Берштейн и др., 2005]. В качестве вершин графа изображаются понятия онтологии, а в качестве ребер – отношения между понятиями. Для любого понятия онтологии можно получить список всех ассоциированных с ним термов. Данная информация может быть представлена в виде таблицы, либо в виде дополнительного графа. Из полученной таблицы имеется возможность выбрать любой терм и система выделит на графе все понятия, с которыми он связан.

Индексация проектных документов состоит из следующих этапов:

- загрузка документов;
- анализ структуры документов;
- удаление стоп-слов;
- стемминг (выделение основы слова, получение термов);
- подсчет относительной частоты встречаемости термов;
- расчет степени выраженности понятий онтологии, построение онтологического представления для разделов и документов;
- генетическая оптимизация онтологических представлений.

В качестве входных данных подсистемы онтологически-ориентированной индексации выступают проектные документы в формате XML. Для данной подсистемы объектом обработки служит не целый ПД, а каждый его раздел в отдельности [Наместников 2009, Наместников и др., 2010]. Онтологическим представлением ПД считается такое описание ПД, которое состоит из множества понятий онтологии с соответствующими степенями выраженности данных понятий в документе.

В процессе онтологически-ориентированного индексирования ПД необходимо определить набор понятий предметной области, который содержится в тексте анализируемого документа.

Степень выраженности понятия c_k в j -м разделе ПД d будем вычислять по следующей формуле [Наместников и др., 2010]:

$$\mu_{S_j^d}(c_k) = 1 - \frac{1}{\sum_{s=1}^{l_k} \max(f_s^k, f_s^j)} \sum_{s=1}^{l_k} |f_s^k - f_s^j|,$$

где S_j^d – j -й раздел ПД d ;

f_s^j, f_s^k – частоты встречаемости термина s в j -м разделе документа и в описании k -го понятия онтологии соответственно;

l_k – мощность текстового входа понятия c_k . В том случае, если термин s отсутствует в j -м разделе документа d , f_s^j принимается равным нулю.

В основе идеи оптимизации онтологического представления ПД лежит гипотеза – *любой текстовый документ можно разделить на множество непересекающихся фрагментов, в каждом из которых будет доминировать то или иное понятие предметной области.*

Пусть имеется предобработанный текстовый документ d , состоящий из последовательности термов:

$$S^d = w_{11}^d, w_{21}^d, \dots, w_{i_1}^d, \dots, w_{n_1}^d, w_{12}^d, \dots, w_{i_2}^d, \dots, w_{n_2}^d, \dots, w_{i_j}^d, \dots, w_{n_m}^d, \quad (2)$$

где i_j – номер термина в k -м предложении, $j = \overline{1, m}$;

$i_j = \overline{1, n_j}$, где n_j – количество термов в j -м предложении.

Обозначим через S_p^d часть последовательности S^d , которая определяется выражением (2), и запишем ее следующим образом:

$$S_p^d = w_{1p}^d, w_{2p}^d, \dots, w_{j,p}^d, \dots, w_{n_p}^d, p = \overline{1, s},$$

при этом выполняется равенство:

$$S_1^d, S_2^d, \dots, S_s^d = S^d, \quad (3)$$

Для нахождения значения доминирования концептов будем применять метод сравнения текстового входа каждого понятия в онтологии предметной области с анализируемым текстом.

Алгоритм вычисления степени доминирования понятия в текстовом фрагменте состоит из следующих шагов:

Шаг 1. Определение максимальной степени выраженности концептов в текстовом фрагменте:

$$\hat{\mu}_{S_p^d}(c) = \max_c(\mu_{S_p^d}(c)).$$

Шаг 2. Определение среднего значения степени выраженности концептов онтологии, исключая концепт с максимальной степенью выраженности (определенный на предыдущем шаге):

$$\tilde{\mu}_{S_p^d}(c) = \frac{1}{n-1} \sum_{i=1}^{n-1} \mu_{S_p^d}(c_i),$$

где $c_i \in c - c_k$, $c_k = \arg \max_c (\mu_{S_p^d}(c))$, n – количество концептов с ненулевой степенью выраженности для текстового фрагмента S_p^d .

Шаг 3. Определение степени детерминированности понятия в текстовом фрагменте S_p^d :

$$\Delta_{S_p^d}(c) = \hat{\mu}_{S_p^d}(c) - \tilde{\mu}_{S_p^d}(c), \quad (4)$$

Выражение (4) фактически определяет качество выделения текстового фрагмента в ПД с целью ограничения в тексте определенного понятия предметной области, которое зафиксировано в онтологии интеллектуального проектного репозитория.

Канонический генетический алгоритм характеризуется следующими особенностями [Скурихин и др., 1995]:

1. Задается целевая функция, определяющая эффективность найденного решения.
2. В соответствии с определенными ограничениями инициализируется исходная популяция потенциальных решений.
3. Каждая хромосома в популяции декодируется в необходимую форму для последующей оценки и затем ей присваивается значение эффективности в соответствии с целевой функцией.
4. Каждой хромосоме присваивается вероятность воспроизведения, которая зависит от эффективности данной хромосомы.
5. В соответствии с вероятностями воспроизведения создается новая популяция хромосом, причем с большей вероятностью воспроизводятся наиболее эффективные элементы.

Формально генетический алгоритм можно описать следующим образом [Скурихин и др., 1995]:

$$GA = (P^0, \lambda, l, \nu, \rho, F, \tau),$$

где $P^0 = (a_1^0, \dots, a_\lambda^0)$ – исходная популяция, где

a_i^0 – решение задачи, представленное в виде хромосомы;

λ – целое число (размер популяции);

l – целое число (длина каждой хромосомы популяции);

ν – оператор отбора;

ρ – отображение, определяющее рекомбинацию

(кроссинговер, мутация);

F – целевая функция;

τ – критерий остановки.

Для решения конкретной задачи оптимизации текстовых фрагментов ПД генетический алгоритм требует следующих уточнений:

- способа кодирования хромосом (потенциальных решений);
- вида целевой функции;
- реализации операций кроссинговера и мутации.

Целью генетической оптимизации в процессе концептуального индексирования ПД является нахождение такой последовательности (3), которая соответствует минимальному значению целевой функции

$$F(S^d) = \frac{1}{s} \sum_p (1 - \Delta_{S_p^d}(c)) \rightarrow \min, \quad (5)$$

$p = \overline{1, s}$, где s – количество текстовых фрагментов;

$s = \overline{1, m}$, где m – количество предложений в индексированном документе.

Таким образом, минимальный текстовый фрагмент соответствует одному отдельно взятому предложению ПД, а максимальный – целому ПД.

Потенциальное решение (хромосома) генетического алгоритма концептуального индекса имеет следующий вид:

$$a_i^t = (< p, j >), p = \overline{1, s}, j = \overline{1, m}, 1 \leq s \leq m, \quad (6)$$

где p – номер текстового фрагмента;

j – номер предложения;

s – количество текстовых фрагментов;

m – количество предложений;

i – номер хромосомы;

t – номер поколения.

Таким образом, хромосома, определяемая выражением (6), представляет собой, в действительности, последовательность текстовых фрагментов (3).

Целевая функция определяет способ отображения хромосомы на единичный отрезок:

$$F : a_i^t \rightarrow [0,1].$$

В качестве целевой функции F будем использовать выражение (5).

На первом шаге работы генетического алгоритма формируется начальная популяция хромосом $P^0 = (a_1^0, \dots, a_\lambda^0)$. Для каждой хромосомы a_i^0 определяется значение целевой функции $F(a_i^0)$. Затем производится ранжирование хромосом. Ранг элементов популяции $rank$ задается следующим образом:

$$\forall i \in \{1, \dots, \lambda\} : rank(a_i^t) = i,$$

если для $\forall j \in \{1, \dots, \lambda - 1\} : F(a_j^t) < F(a_{j+1}^t)$.

Первые g хромосом без изменения переходят в следующий пул (поколение), а остальное количество $(\lambda - g)$ формируется посредством операции кроссинговера. При определении оператора кроссинговера будем учитывать то, что последовательность предложений в тексте и их количество должны оставаться неизменными в процессе трансформации хромосом. Точка кроссинговера определяется случайным образом на границе двух текстовых фрагментов:

$$a_i^0 = (\dots, \langle p, j \rangle, \langle p+1, j+1 \rangle, \dots)$$

для первой из двух хромосом, участвующих в кроссинговере. Так как в процессе рассматриваемой операции происходит взаимообмен частями хромосом и, принимая во внимание вышеприведенные ограничения, точку кроссинговера для второй хромосомы выбираем так, чтобы в левой части остались j первых предложений, как и у первой хромосомы.

Заключительным этапом формирования новой популяции является применение оператора мутации. В задаче концептуального индексирования предлагается применять два варианта мутации хромосом: 1) сдвиг границы текстового фрагмента и 2) объединение текстовых фрагментов.

Первый вариант мутации со сдвигом границы текстового фрагмента предполагает вероятностный выбор границы между двумя текстовыми фрагментами ПД. Далее принимается решение о направлении сдвига границы в правую или в левую сторону, учитывая количество предложений в соседних текстовых фрагментах. Граница перемещается на одно предложение в сторону с большим количеством предложений. При равенстве предложений направление выбирается случайным образом. Сдвиг границы не происходит в случае, если текстовый фрагмент содержит одно предложение.

Вариант мутации, объединяющий два соседних фрагмента, фактически уменьшает количество текстовых фрагментов в ПД за счет их укрупнения.

В основе подсистемы онтологически-ориентированной кластеризации ПД лежит модифицированный алгоритм нечеткой кластеризации Fuzzy C-Means, с учетом рассмотрения онтологического представления ПД как иерархии. Тем самым, мера сходства между ПД находится через сложность превращения одной иерархии в другую [Загоруйко, 1999].

В системе есть возможность учитывать две меры сходства ПД:

- мера сходства содержимого ПД;
- мера сходства структур ПД.

Подсистема онтологически-ориентированной кластеризации работает со следующими представлениями ПД: онтологическое представление документа как нечеткий вершинный

подграф онтологии, оптимизированное онтологическое представление и классическое представление ПД в виде множества пар «термин–частота».

Подсистема онтологически-ориентированного информационного поиска состоит из следующих модулей:

- Модуль взаимодействия с онтологией – осуществляет процедуру подключения к онтологии, выполнение запросов к ней и обработку полученных результатов.
- Модуль поиска наиболее выраженного понятия – выполняет поиск в прикладной онтологии наиболее значимого понятия для набора ключевых слов (запроса пользователя).
- Модуль поиска терминов – для заданного понятия выполняет поиск дополнительных терминов в онтологии, которые в большей степени соответствуют найденному понятию.

Модуль поиска документов – осуществляет поиск среди множества документов по расширенному набору терминов.

3. Результаты экспериментов

Тестовое множество состоит из 262 проектных документов. Эксперт разбил данную выборку по четырем признакам:

- по классу документации (2 группы);
- по виду документации (14 групп);
- по разделу документации (14 групп);
- по тематике работ (38 групп).

Для оценки качества онтологически-ориентированной кластеризации ПД использовалась целевая функция следующего вида:

$$\hat{f}_i = 1 - \frac{\max\left(\sum_{i=1}^M \bar{K}_r^i, \sum_{i=1}^M \hat{K}_r^i\right)}{N},$$

где \bar{K}^i – множество отсутствующих документов, входящих в i -й кластер согласно сопоставлению результатов экспертного и автоматического разбиений;

\hat{K}^i – множество «лишних» документов, входящих в i -й кластер согласно сопоставлению результатов экспертного и автоматического разбиений;

$i = \overline{1, M}$ – номер кластера;

M – количество кластеров;

N – количество документов.

На основе тестового множества ПД, с помощью подсистемы онтологически-ориентированного индексирования, был построен набор индексов, включающих в себя онтологические представления (ОП), оптимизированные онтологические представления (ООП) и классические индексы (КИ), содержащие пары «термин, частота». Оценка

качества кластеризации представлена в таблице 1.

Таблица 1 – Оценка качества кластеризации

Вид экспертного разбиения	Тип индекса	Время кластеризации	Значение оценочной функции
Класс документации	ОП	14,181	0,61
	ООП	27,410	0,82
	КИ	197,815	0,51
Вид документации	ОП	465,137	0,29
	ООП	1077,836	0,24
	КИ	48488,873	0,21
Раздел документации	ОП	465,137	0,23
	ООП	1077,836	0,25
	КИ	48488,873	0,17
Тематика работ	ОП	2273,952	0,27
	ООП	5315,631	0,24
	КИ	133262,323	0,14

Как видно из таблицы 1, процесс кластеризации ОП и ООП проходит быстрее (до 104 раз) относительно времени кластеризации КИ, а качество кластеризации данных типов индексов выше по сравнению с результатами кластеризации классических индексов. Временные затраты на кластеризацию ООП примерно в 2 раза больше, чем на кластеризацию ОП, и при этом ООП показывают лучшие результаты при кластеризации по классу и разделу документации.

ЗАКЛЮЧЕНИЕ

Результаты исследований показали, что онтологически-ориентированная кластеризация ПД является эффективной по сравнению с методом кластеризации, где документы представляются в виде набора «термин – частота». Значительно (приблизительно до 104 раз) сокращается время кластеризации онтологических представлений документов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Берштейн и др., 2005] Берштейн А.С., Боженюк А.В. Нечеткие графы и гиперграфы. – М.: Научный мир, 2005.
- [Добров и др., 2006] Добров Б.В., Лукашевич Н.В., Лингвистическая онтология по естественным наукам и технологиям: основные принципы разработки и текущее состояние // Десятая национальная конференция по искусственному интеллекту с международным участием (Обнинск, 25-28 сентября 2006 г.) – М.: Физматлит, 2006.
- [Загоруйко 1999] Загоруйко Н.Г. Прикладные методы анализа данных и знаний - Новосибирск: ИМ СО РАН, 1999. - 270 с.
- [Наместников 2009] Интеллектуальные проектные репозитории. – Ульяновск: УлГТУ, 2009. Наместников А.М. Интеллектуальные проектные репозитории. – Ульяновск: УлГТУ, 2009.
- [Наместников и др., 2010] Наместников А.М., Филиппов А.А. Концептуальная индексация проектных документов //

Автоматизация процессов управления. – 2010. – №2(20). – С. 34-39.

[Скурихин и др., 1995] Скурихин А.Н. Генетические алгоритмы // Новости искусственного интеллекта; 1995. – № 4. – С. 6–17.

ONTOLOGICAL SYSTEM FOR CLUSTERING AND FULL-TEXT SEARCHING OF THE CAD DOCUMENTS

Namestnikov A.M. *, Subkhangulov R.A. *,
Filippov A.A. *

* Ulyanovsk State Technical University,
Ulyanovsk, Russia

nam@ulstu.ru

subkhangulov-ruslan@yandex.ru

al.filippov@ulstu.ru

In article the description of program system for clustering of CAD text documents is provided. The method based on a current status of domain ontology. The system architecture, structure of ontology and its fragments in the RDF format is shown, basic functions of system are listed. The results of the experiments proving efficiency of developed program system are given.

INTRODUCTION

In CAD archive the designer has no possibility to solve semi-structured problems of search. Full-text query search for the document, finding of the close document according to the contents, a clustering of documents can be such tasks.

MAIN PART

The architecture of the developed system is provided. The domain ontology includes two levels: conceptual and terminological. In the editor of ontology the following modules are implemented: module of formation for the diagram of ontology and module of formation for the domain maintenance; module of formation for a terminological surround of ontology concepts; the interaction module with Sesame RDF storage; the module for visualization of domain ontology.

CONCLUSION

Results of researches showed that method of ontological clustering is more effective than a method where documents are represented in the form of a set «term – frequency». Considerably time of a ontological clustering is reduced.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

НЕЧЁТКИЕ СЕМАНТИЧЕСКИЕ МОДЕЛИ ТЕКСТОВОГО ПОИСКА

Панкова Л.А. *, Пронина В.А. **

*Федеральное государственное бюджетное Учреждение науки Институт проблем управления им.
В.А. Трапезникова Российской академии наук, г. Москва, Россия*

*pankova@ipu.ru

**pron@ipu.ru

Понятия текстового поиска интерпретированы в терминах теории нечётких множеств. Предложены модели текстового поиска на основе теории нечётких множеств.

Ключевые слова: текстовый поиск, связанность, нечёткое отношение, релевантность, принцип обобщения

ВВЕДЕНИЕ

В данной работе рассматривается семантический поиск по запросу в коллекции научных документов на основании содержимого этих документов. Модель текстового поиска включает модель поискового запроса, модель документа и модель релевантности (соответствия) документа запросу. В работе рассматриваются модели запроса и документа как наборы понятий (терминов) онтологии предметной области коллекции с коэффициентами (веса) от 0 до 1, отражающими важность понятий для описания содержания. В запросе назначенные пользователем веса определяют его информационную потребность. В тексте документа веса определяются в автоматическом процессе концептуального индексирования [Соловьев и др., 2006]. Релевантность (семантическое соответствие) документа запросу в рассматриваемых моделях формально определяется с использованием отношения семантической связанности понятий. Семантическая связанность понятий может вычисляться формальным образом по онтологии предметной области данной коллекции или с использованием статистических методов, а может задаваться экспертом.

Текстовый поиск имеет дело с нечёткой априорной информацией, что не принимается в расчет в большинстве существующих чётких моделей (см., например, обзор из [Панкова и др., 2011]). Теория нечётких множеств даёт средства обращения с нечёткой информацией. В существующих работах по информационному поиску теория нечётких множеств применяется в

основном для представления онтологии и реализации более гибких способов формулирования запросов. На практике применяемые модели информационного поиска по-прежнему основаны на других подходах. Отчасти это можно объяснить тем, что необходимы экспериментальные доказательства, чтобы продемонстрировать, действительно ли нечеткие модели в состоянии превзойти современные подходы. Кроме того, необходимы дополнительные усилия, чтобы обнажить то, что теория нечетких множеств может предложить в этой области элегантные и интуитивно привлекательные методы.

В данной работе рассматриваются модели текстового поиска, использующие теорию нечётких множеств [Заде, 2000]. Предлагаемые модели используют принцип обобщения – универсальный принцип теории нечётких множеств [Орловский, 1981] – для перехода от отношений между понятиями к отношениям между документами и запросами, от связанности понятий к релевантности документов и запросов.

Структура работы. В первом разделе основные понятия текстового поиска интерпретируются в терминах теории нечётких множеств. Во втором разделе предложены модели текстового поиска в рамках теории нечётких множеств. В третьем разделе дан пример ранжирования коллекции документов по релевантности запросу, вычисленной предложенными методами.

1. ПОНЯТИЯ ТЕКСТОВОГО ПОИСКА В ТЕРМИНАХ ТЕОРИИ НЕЧЁТКИХ МНОЖЕСТВ

Интерпретируем понятия текстового поиска в терминах теории нечётких множеств.

Пусть D – конечное множество документов коллекции, C – конечное множество понятий предметной области коллекции, Q – конечное множество запросов.

1.1. Множество концептуальных индексов документов можно представить как нечёткое бинарное индексирующее отношение I :

$$I = \{\mu_I(d, c)/(d, c) \mid d \in D; c \in C\},$$

где $\mu_I: D \times C \rightarrow [0, 1]$ – функция принадлежности, обозначающая для каждой пары (d, c) степень принадлежности понятия c документу d (вес понятия в концептуальном индексе). Индексирующее отношение I индуцирует множества I_d (концептуальные индексы) как нечеткие множества на множестве понятий:

$$I_d = \{\mu_{I_d}(c)/c \mid c \in C, \mu_{I_d}(c) = \mu_I(d, c)\},$$

где $\mu_{I_d}(c)$ – вес понятия в концептуальном индексе документа.

1.2. Множество концептуальных индексов запросов можно представить как нечёткое бинарное индексирующее отношение:

$$U = \{\mu_U(q, c)/(q, c) \mid q \in Q; c \in C\},$$

где $\mu_U(q, c)$ – функция принадлежности, обозначающая для каждой пары (q, c) степень информационной потребности понятия c в запросе q (вес понятия в концептуальном индексе запроса). Запрос q представляется как нечеткое множество понятий:

$$I_q = \{\mu_{I_q}(c)/c \mid c \in C, \mu_{I_q}(c) = \mu_U(q, c)\}$$

1.3. Отношение семантической связанности понятий S можно представить как нечёткое рефлексивное отношение на $C \times C$ с функцией принадлежности $\mu_S(c_i, c_j) = s(c_i, c_j)$, где $s(c_i, c_j) \in [0, 1]$ – семантическая связанность понятий c_i и c_j :

$$S = \{\mu_S(c_i, c_j)/(c_i, c_j) \mid c_i, c_j \in C\}.$$

2. НЕЧЁТКИЕ МОДЕЛИ ТЕКСТОВОГО ПОИСКА, ОСНОВАННЫЕ НА ПРИНЦИПЕ ОБОБЩЕНИЯ

Предлагаемые модели используют принцип обобщения и интуитивно просты. Принцип обобщения – это универсальный принцип теории нечётких множеств. В предлагаемых моделях принцип обобщения используется для перехода от отношения на понятиях к отношению на документах и запросах, а именно, от связанности понятий к релевантности документов и запросов.

На первом этапе отношение связанности на понятиях обобщается, чтобы получить нечёткое отношение связанности S' нечетких запросов с

одним понятием:

$$\mu_{S'}(I_q, c_j) = \max\{\min_{c_i \in C} \{\mu_{I_q}(c_i), \mu_S(c_i, c_j)\}\}.$$

Затем принцип обобщения используется еще раз. При этом нечёткое отношение связанности S' нечетких запросов с понятием обобщается, чтобы получить обобщённое нечёткое отношение связанности S'' на $\{I_q\} \times \{I_d\}$ с функцией принадлежности $\mu_{S''}(I_q, I_d)$, которую будем называть **обобщённой связанностью документов и запросов**:

$$\begin{aligned} \mu_{S''}(I_q, I_d) &= \max\{\min_{c_j \in C} \{\mu_{I_d}(c_j), \mu_{S'}(I_q, c_j)\}\} = \\ &= \max\{\min_{c_j \in C} \{\mu_{I_d}(c_j), \max_{c_i \in C} \{\mu_{I_q}(c_i), \mu_S(c_i, c_j)\}\}\}. \end{aligned}$$

Эта формула преобразуется к виду:

$$\mu_{S''}(I_q, I_d) = \max\{\min_{c_i, c_j \in C} \{\mu_{I_q}(c_i), \mu_{I_d}(c_j), \mu_S(c_i, c_j)\}\}.$$

Таким образом, обобщённое нечёткое отношение связанности документов с запросами S'' имеет вид:

$$S'' = \{\mu_{S''}(I_q, I_d)/(I_q, I_d) \mid I_q \in \{I_q\}, I_d \in \{I_d\}\}.$$

Рисунки 1-3 иллюстрируют применение принципа обобщения.

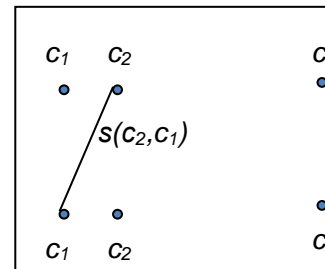


Рисунок 1 – Семантическая связанность понятий

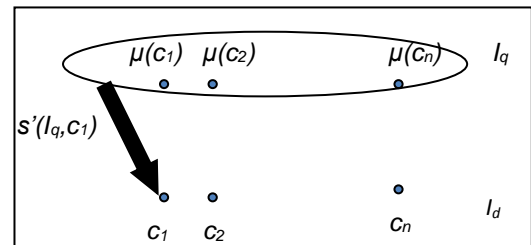


Рисунок 2 – Семантическая связанность запроса и понятия

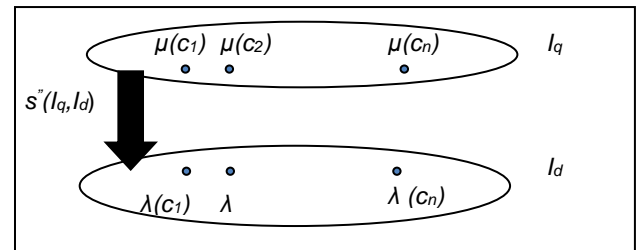


Рисунок 3 – Семантическая связанность запроса и документа

Если I_q и I_d – одноэлементные нечеткие множества, то S'' интерпретируется как обобщённое нечёткое отношение связанности двух нечётких понятий с функцией принадлежности (максимум исчезает, т.к. рассматривается одна пара понятий), которое будем называть **обобщённой связанностью**

двух понятий:

$$\mu_S(\tilde{c}_i, \tilde{c}_j) = \min\{\mu_{I_q}(c_i), \mu_{I_d}(c_j), \mu_S(c_i, c_j)\}.$$

Заметим, что обобщённая связанность I_q и I_d можно представить как максимальное значение обобщённых связанностей всех пар нечётких понятий запроса и документа:

$$\mu_S(I_q, I_d) = \max_{\tilde{c}_i \in I_q, \tilde{c}_j \in I_d} \{\mu_S(\tilde{c}_i, \tilde{c}_j)\}.$$

Вводится параметр **ширины связанности запроса и документа** $N(q, d)$, определяющий число пар понятий запроса и документа, для которых обобщённая связанность принадлежит заданному полуинтервалу:

$$N(q, d) = |\{(c_i, c_j) \mid \mu_S(\tilde{c}_i, \tilde{c}_j) \in (\delta_1, \delta_2]\}|,$$

$$\tilde{c}_i \in I_q, \tilde{c}_j \in I_d, \delta_1 < \delta_2,$$

$$\delta_1, \delta_2 \in (0, \max_{\tilde{c}_i \in I_q, \tilde{c}_j \in I_d} \{\mu_S(\tilde{c}_i, \tilde{c}_j)\}].$$

Введённые понятия можно использовать для определения релевантности (семантического соответствия) документа запросу и ранжирования документов.

1. Релевантность документа запросу $R(q, d)$ вычисляется как обобщённая связанность документа и запроса, и документы ранжируются по значению релевантности:

$$\begin{aligned} R(q, d) &= \mu_S(I_q, I_d) = \\ &= \max_{c_i, c_j \in C} \{\min\{\mu_{I_q}(c_i), \mu_{I_d}(c_j), \mu_S(c_i, c_j)\}\}. \end{aligned}$$

2. Релевантность документа запросу вычисляется как ширина связанности $N(q, d)$ при заданном δ_1 , и документы ранжируются по значению релевантности.

3. Для ранжирования документов используются $R(q, d)$ и $N(q, d)$ по алгоритму, описанному ниже.

- Для каждого документа вычисляется $R(q, d)$.
- Определяется α -срез – множество документов D_α , релевантных запросу с релевантностью большей α .
- На полуинтервале $(\alpha, \max_{d \in D} \{R(q, d)\})$ вводится лингвистическая переменная «релевантность». Множество документов D_α разбивается на классы эквивалентности D_α^i в соответствии со значениями «релевантности»:

$$D_\alpha^i = \{d \mid \delta_{i1} < R(q, d) \leq \delta_{i2}, i = 1, \dots, k\},$$

где k – число значений лингвистической переменной «релевантность».

- Для каждого документа $d \in D_\alpha^i$ вычисляется $N^i(q, d)$:

$$N^i(q, d) = |\{(c_i, c_j) \mid \mu_S(\tilde{c}_i, \tilde{c}_j) \in (\delta_1, \delta_2], \tilde{c}_i \in I_q, \tilde{c}_j \in I_d\}|.$$

- Внутри каждого класса эквивалентности документы ранжируются по $N^i(q, d)$.

3. ПРИМЕР

Пусть множество C состоит из следующих понятий:

$c_1 = Fuzzy\ logic$

$c_2 = Fuzzy\ relation\ equations$

$c_3 = Fuzzy\ modus\ ponens$

$c_4 = Approximate\ reasoning$

$c_5 = Max\text{-}min\ composition$

$c_6 = Fuzzy\ implication$

Запрос включает понятия c_1, c_2, c_3 и представлен вектором:

$$I_q = \begin{bmatrix} c_1 & c_2 & c_3 \\ 1 & .4 & .1 \end{bmatrix}.$$

Отношение семантической связанности понятий S (необходимый для вычислений фрагмент) задаётся матрицей:

$$S = \begin{bmatrix} & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \\ c_1 & 1 & .2 & 1 & 1 & .5 & 1 \\ c_2 & .2 & 1 & .1 & .7 & .9 & 0 \\ c_3 & 1 & .4 & 1 & .9 & .3 & 1 \end{bmatrix},$$

индексирующее отношение I – матрицей:

$$I = \begin{bmatrix} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 & d_9 & d_{10} \\ c_1 & .2 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ c_2 & 1 & 0 & 0 & .3 & 0 & .4 & 0 & 0 & 1 & 0 \\ c_3 & 0 & 0 & .8 & 0 & .4 & 0 & 1 & 0 & 0 & 0 \\ c_4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & .9 & .7 & .5 \\ c_5 & 1 & 0 & .5 & 0 & 0 & .6 & 0 & 0 & 0 & 0 \\ c_6 & 0 & 1 & 0 & 0 & .2 & 0 & 1 & 0 & 0 & .5 \end{bmatrix}.$$

3.1. Ранжирование по обобщённой связанности документа и запроса

Релевантности документов запросу вычисляются как обобщённые связанности документов и запроса:

$$R(q, d) = \begin{bmatrix} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 & d_9 & d_{10} \\ .5 & 1 & 1 & .3 & .4 & .5 & 1 & .9 & .7 & .5 \end{bmatrix}.$$

Определяется α -срез при $\alpha = .5$:

$$R_{q(.5)} = \begin{bmatrix} d_2 & d_3 & d_7 & d_8 & d_9 \\ 1 & 1 & 1 & .9 & .7 \end{bmatrix}.$$

Документы упорядочиваются по значению $R_{q(.5)}$:

$$\begin{matrix} 1 \\ .9 \\ .7 \end{matrix} \begin{bmatrix} d_2 & d_3 & d_7 \\ & d_8 & \\ & & d_9 \end{bmatrix}.$$

3.2. Ранжирование по ширине связанности запроса и документа

Релевантность документа запросу вычисляется как значение $N(q,d)$ при $\delta_l=.5$:

$$N(q,d) = \begin{bmatrix} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 & d_9 & d_{10} \\ 0 & 2 & 2 & 0 & 0 & 0 & 3 & 1 & 1 & 0 \end{bmatrix}.$$

Документы упорядочиваются по значению $N(q,d)$:

$$\begin{matrix} 3 \\ 2 \\ 1 \end{matrix} \begin{bmatrix} d_7 \\ d_2 & d_3 \\ d_8 & d_9 \end{bmatrix}.$$

3.3. Ранжирование по обобщённой связанности и по ширине связанности

Вычисляется α -срез при $\alpha = .5$ (по 3.1.).

$$R_{q(.5)} = \begin{bmatrix} d_2 & d_3 & d_7 & d_8 & d_9 \\ 1 & 1 & 1 & .9 & .7 \end{bmatrix}.$$

Полуинтервал $(.5, 1]$ делится на два полуинтервала $(.5, .9]$ и $(.9, 1]$, соответствующих значениям лингвистической переменной: «умеренная» и «сильная» релевантность. Множество документов $D_{0.5}$ разбивается на два класса эквивалентности: $\{d_8, d_9\}$ и $\{d_2, d_3, d_7\}$.

Внутри каждого класса документы ранжируются по $N^i(q,d)$. Итоговая ранжировка имеет вид:

$$\begin{matrix} (.9,1] & 3 \\ (.9,1] & 2 \\ (.9,1] & 1 \\ (.5,.9] & 1 \end{matrix} \begin{bmatrix} d_7 \\ d_2 & d_3 \\ d_8 & d_9 \end{bmatrix}.$$

ЗАКЛЮЧЕНИЕ

Понятия текстового поиска интерпретируются в терминах теории нечётких множеств. Предлагаются модели текстового поиска в рамках теории нечётких множеств. Дан пример вычисления релевантности коллекции документов запросу по предложенным моделям.

Для проверки адекватности предложенных моделей текстового поиска планируется экспериментальное исследование на коллекции научно-технических текстов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Соловьев и др., 2006] Соловьев В.Д., Добров Б.В., Иванов В.В., Лукашевич Н.В. Онтологии и тезаурусы: Учебное пособие. Казань, Москва: Казанский государственный университет, МГУ им. М.В. Ломоносова, 2006.

[Панкова и др., 2011] Панкова Л.А., Пронина В.А., Крюков К.В. Онтологические модели поиска экспертов в системах управления знаниями научных организаций // Проблемы управления. – 2011. – № 6. – С. 52–60.

[Заде, 2000] Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. СПб.: Питер, 2000.

[Орловский, 1981] Орловский С.А. Проблемы принятия решений при нечеткой исходной информации. М: Наука. Главная редакция физико-математической литературы, 1981.

SEMANTIC TEXT RETRIEVAL BASED ON FUZZY SET THEORY

Pankova L.A. *, Pronina V.A. **

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Profsoyuznaya ul., 65, Moscow, 117997 Russia

*pankova@ipu.ru

**pron@ipu.ru

Text retrieval concepts are interpreted in terms of the fuzzy set theory. The text retrieval models based on the fuzzy set theory are proposed.

INTRODUCTION

The crisp models of text retrieval does not count fuzziness of information. The fuzzy set theory provides means of handling fuzzy information. In the existing works the fuzzy set theory is mainly used to represent the ontology and the implementation of more flexible ways for formulating queries. In practice, text retrieval models are still based on other approaches. The fuzzy set theory can offer in this area are elegant and intuitively attractive methods.

MAIN PART

The proposed models are based on the generalization principle and intuitively simple. The generalization principle is universal principle of the fuzzy set theory.

In proposed models the generalization principle is used for moving from the relation on the concepts to the relation on documents and queries, namely, from relatedness of concepts to the relevance of documents and queries.

The article presents the example that shows results of modeling.

CONCLUSION

There is planned experimental verification on a collection of scientific-technical texts.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.912

ПОИСК И РЕФЕРИРОВАНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ В МНОГОЯЗЫЧНОЙ СРЕДЕ

Липницкий С.Ф., Мамчич А.А., Степура Л.В.

*Объединенный институт проблем информатики НАН Беларуси,
г. Минск, Республика Беларусь*

lipn@newman.bas-net.by

lexamam@newman.bas-net.by

stepura@newman.bas-net.by

Рассмотрена технология поиска и обработки научно-технической информации из различных источников (Интернет, локальная сеть, жесткий диск компьютера пользователя) в многоязычной среде. Предложена архитектура информационной системы. Представлены ее основные функции: индексирование текстовых документов, их поиск и автоматическое реферирование.

Ключевые слова: информационный поиск; индексирование; автоматическое реферирование; текст.

ВВЕДЕНИЕ

В докладе рассматривается автоматизированная система поиска и реферирования текстовой информации в многоязычной среде, разработанная в ОИПИ НАН Беларуси. Реализованные в системе модели и алгоритмы позволяют активизировать и эффективно использовать информационные ресурсы из различных источников (Интернет, локальная сеть, жесткие диски отдельных компьютеров пользователей).

Предложенный в данной работе подход к поиску и реферированию текстовой информации, в отличие от существующих, основан на использовании тематических и динамических корпусов текстов (совокупностей текстов по конкретной тематике), что обеспечивает адаптацию системы к решаемой задаче и информационным потребностям пользователей, а также независимость программного обеспечения от входных языков. Такие корпусы текстов могут создаваться предварительно под прогнозируемые задачи, а также формироваться оперативно после поступления запроса (динамические корпусы текстов).

1. Архитектура системы

Функциональными компонентами системы поиска и аналитической обработки научно-технической информации являются три подсистемы:

– подсистема индексирования текстовых

документов в Интернете, локальной сети и на жестком диске;

– подсистема информационного поиска;

– подсистема реферирования текстовых документов.

В состав подсистемы индексирования входят следующие информационно-программные средства:

– программы выявления информативных слов и предложений. Используются при индексировании текстовых документов и веб-страниц;

– программы индексирования текстовых документов и веб-страниц. Каждому документу (странице) приписывается совокупность поисковых признаков с их весами (числовыми значениями информативности).

Подсистема информационного поиска включает:

– программы поиска текстовых документов и веб-страниц. Результатом поиска являются документы и адреса страниц, упорядоченные по убыванию их информативности;

– программы поиска по содержанию. Реализуют фактографический информационный поиск в полнотекстовых документах;

Подсистема реферирования текстовых документов состоит из программ выявления информативных предложений и синтеза связных рефератов.

Предложенный разработчиками метод поиска и реферирования текстовой информации обеспечивает функционирование системы в многоязычной среде. Адаптация программного

комплекса к новому входному языку не требует доработки и корректировки программ. Необходимо лишь сформировать в базе данных корпус текстов на этом языке. Процедуры создания словарей базы знаний реализуются в автоматизированном режиме. В настоящее время пользователи системы могут работать на русском, белорусском, английском и немецком языках.

Все подсистемы реализованы в виде единого программного приложения, что позволяет наиболее эффективно организовать процесс работы пользователей и администратора информационной системы.

1.1. Корпусы текстов

Задачи создания и использования корпусов текстов решаются в рамках специального раздела языкознания – корпусной лингвистики. Под корпусом текстов понимают совокупность документов, накопленных и размеченных по определенным правилам в зависимости от назначения. В случае отсутствия разметки эти совокупности называют корпусами текстов первого порядка. Различают тематические корпусы текстов (наборы текстов по предметным областям) и полные корпусы текстов, каждый из которых объединяет все тематические корпусы на данном входном языке. Для каждого языка (например, русского, белорусского, английского) создается свой полный корпус текстов.

1.2. Словари базы знаний

1.2.1. Словари словоформ и парадигм

В словаре словоформ каждой словоформе поставлены в соответствие:

- частота в полном корпусе текстов;
- частоты во всех тематических корпусах текстов;
- номер (код) парадигмы.

Словоформы и их статистические характеристики хранятся в словаре словоформ. В первоначальном состоянии каждая словоформа словаря образует отдельную парадигму. После объединения некоторых (или всех) словоформ в словоизменительные парадигмы словоформам присваивается номер парадигмы, элементом которой эта словоформа является.

Словарь парадигм служит для поиска всех словоформ парадигмы после нахождения словоформы и ее кода в словаре словоформ. Процедура поиска используется при вычислении информативности слов. Создается и актуализируется словарь парадигм в человеко-машинном режиме с использованием соответствующего инструментария. В первоначальном варианте каждая парадигма словаря парадигм содержит одну-единственную словоформу для каждого кода словоформы. После формирования парадигм коды меняются.

1.2.2. Словарь синонимичных словоформ

Словарь состоит из групп синонимичных словоформ, которые могут быть использованы при определении их информативности (две синонимичные словоформы считаются двумя вхождениями лексемы в текст документа).

На первоначальном этапе информационная система может работать без сформированных словарей парадигм и синонимичных словоформ (т. е. с «пустыми» словарями).

2. Основные задачи системы

Конкретные задачи в системе решаются на основе реализации двух видов информационного поиска – поиска веб-страниц с выдачей их адресов, упорядоченных по убыванию информативности этих страниц, и фактографического поиска в полнотекстовых документах.

2.1. Индексирование документов и тематических корпусов текстов

Целью индексирования текста является приписывание ему совокупности ключевых слов с их весами (вес – это информативность слова). При индексировании используются абсолютные частоты слов в документе (если его объем достаточно большой) или в релевантном тексте тематическом корпусе текстов (если это краткое сообщение, т. е. объем текста небольшой), а также абсолютные частоты слов в полном корпусе текстов. Информативность слова вычисляется как отношение этих частот. При этом:

- частота слова в документе – это сумма частот всех словоформ, встречающихся в документе и являющихся словоизменениями исходной словоформы или ее синонимами, зафиксированными в словаре словоизменительных парадигм и в словаре синонимичных словоформ;
- частота слова в полном корпусе текстов – это сумма частот всех словоформ в полном корпусе текстов, в которой учтены словоизменения и синонимы.

Поисковый образ тематического корпуса текстов создается в виде набора ключевых слов с весами. Индексирование реализуется по аналогии с индексированием текстовых документов большого объема. В данном случае все тексты тематического корпуса объединяются и индексируются как единый текстовый документ.

2.2. Индексирование кратких сообщений и запросов пользователей

Краткое сообщение – это текстовый документ, объем которого не позволяет выявить статистические характеристики его словоформ. Для индексирования краткого сообщения используется релевантный ему тематический или динамический корпус текстов. Поисковым образом краткого сообщения считается поисковый образ найденного релевантного тематического корпуса текстов, из

которого исключены все словоформы, не содержащиеся в кратком сообщении (с учетом словоизменения и синонимии).

При информационном поиске в системе используются два основных типа запросов:

– свободно формулируемые запросы на естественном языке. Это традиционный тип запроса. При индексировании всем его словам приписывается информативность, равная 100%. Возможно также индексирование запроса с предварительным поиском наиболее релевантного ему тематического корпуса текстов. В этом случае ключевым словам запроса ставятся в соответствие весовые коэффициенты из словаря словоформ;

– запросы, формулируемые пользователем с применением специального графического интерфейса, где каждое слово запроса располагается на вертикальной шкале информативности. В зависимости от местоположения слова ему присваивается соответствующий весовой коэффициент.

Запрос индексируется аналогично индексированию краткого сообщения. Возможности подсистемы индексирования проиллюстрированы на рисунке 1.

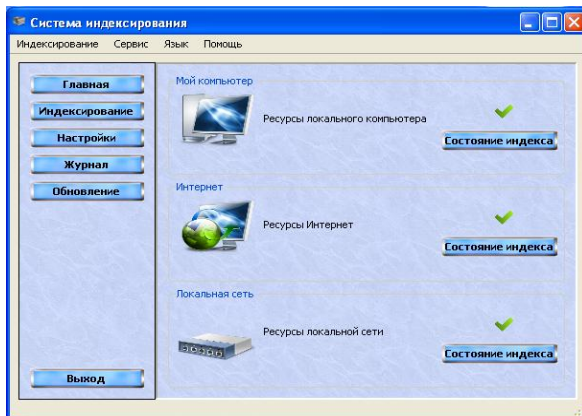


Рисунок 1 – Главное окно подсистемы индексирования

2.3. Поиск текстовых документов

Процесс поиска информации заключается в сравнении запросов пользователей с поисковыми образами проиндексированных документов. Поиску предшествует автоматическая коррекция запроса с целью адаптации системы к информационным потребностям пользователя. Коррекция реализуется следующим образом: на основе первоначального запроса создается динамический корпус текстов как подмножество полного корпуса; документы из динамического корпуса предъявляются пользователю, который исключает из него все непертинентные тексты; полученное в результате множество считается уточненным динамическим корпусом, на основе которого путем его индексирования формируется уточненное поисковое предписание. Процедура оценки пользователем пертинентности текстов может не проводиться. В этом случае для создания уточненного запроса

используется исходный динамический корпус текстов. Главное окно подсистемы поиска представлено на рисунке 2.

2.4. Реферирование текстовых документов

Процесс реферирования включает следующие основные этапы: вычисление информативности слов и предложений реферлируемого документа; разбиение текста на монотематические фрагменты и установление ситуативных связей между ними; вычисление информативности монотематических фрагментов; синтез реферата. Реферат строится из информативных предложений путем поиска релевантной информации в специальной системе словарей и последующего синтеза выходного текста. Алгоритм реферирования функционирует следующим образом.

Лингвистический процессор проводит синтаксический анализ реферлируемого текста. В результате получаем упорядоченную совокупность синтаксических деревьев всех его предложений.

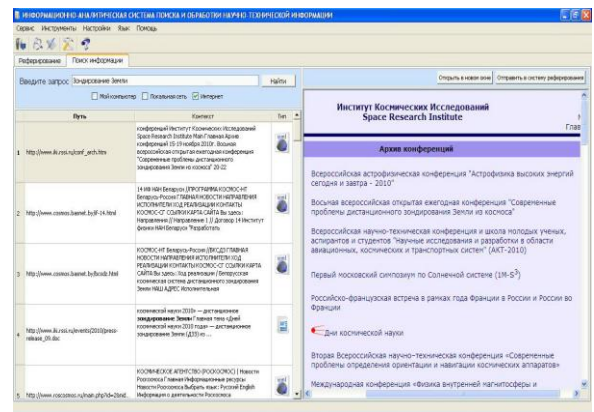


Рисунок 2 – Главное окно подсистемы поиска

Далее статистический анализатор определяет информативность каждого слова, т. е. эмпирическую вероятность того, что это слово извлечено из тематического корпуса текстов при условии, что оно уже извлечено из полного. Из полученной на этом шаге алгоритма совокупности синтаксических деревьев последовательно исключаются деревья (в порядке возрастания информативности слов) до получения требуемого объема будущего реферата. За информативность синтаксического дерева принимается максимальный из показателей информативности его слов. Далее из каждого оставшегося синтаксического дерева удаляются их неинформативные висячие поддерева. Заключительными шагами алгоритма реферирования являются поиск в базе знаний адекватного синтаксического шаблона реферата, заполнение его слотов полученными на предыдущем шаге синтаксическими деревьями и синтез реферата на выходном языке. Главное окно подсистемы реферирования научно-технической информации изображено на рисунке 3.

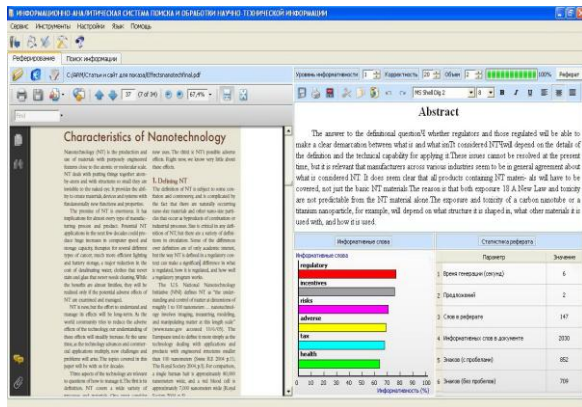


Рисунок 3 – Главное окно подсистемы реферирования

ЗАКЛЮЧЕНИЕ

Разработанная технология поиска и реферирования текстовой информации может быть использована в библиотеках, в информационно-аналитических отделах различных служб и организаций, которые осуществляют оперативный сбор и аналитическую обработку текстовых документов по различным предметным областям в Интернете, локальных сетях и на жестких или съемных дисках отдельных компьютеров. Созданный программный комплекс обеспечивает:

- индексирование, поиск и реферирование текстовых документов из различных информационных источников в многоязычной среде;
- многопоточность процессов индексирования, поиска и реферирования;
- поддержку наиболее распространенных форматов представления текстовых документов (html, shtml, doc, rtf, docx, pdf, txt) с возможностью подключения дополнительных форматов, таких как ppt, xls, wpd, hlp, odt и xml.

RETRIEVAL AND SUMMARIZATION OF TEXT INFORMATION IN A MULTILINGUAL ENVIRONMENT

Lipnitsky S.F., Mamchich A.A., Stepura L.V.

The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Republic of Belarus

lipn@newman.bas-net.by

lexamam@newman.bas-net.by

stepura@newman.bas-net.by

The technology of search and processing of scientific and technical information from different sources (Internet, LAN, user's hard disk) in a multilingual environment is considered. Information system architecture is proposed. Its main functions are presents: indexing, retrieval and automatic summarization of text documents.

INTRODUCTION

Automated system of retrieval and summarization of textual information in a multilingual environment was developed in UIIP NASB. Implemented in the system models and algorithms are used to enhance and effectively use information resources from various sources (Internet, LAN, hard drives of individual computer users).

Proposed in this paper approach to retrieving and summarization the text information, in contrast to existing ones, based on thematic and dynamic text corpora (set of texts on specific topics), which adapts the system to the task and the information needs of users, and independent software of the input language. Such a text corpora can be created under the projected pre-task and quickly formed after the request (dynamic text corpora).

MAIN PART

In order to improve the efficiency of indexing, retrieving and summarization documents from various information sources, in this paper we proposed an approach which, unlike existing methods, is based on using thematic corpora (collections of texts on specific topics) as a knowledge domain and specialized knowledge base dictionaries formed on their basis.

This technique provides the system adaptation to the task and the independence of a software complex from the input language. Text corpora can be created under the projected pre-task or formed directly on-line by combining sets of documents that are relevant to each particular text or a user query (we called them – dynamic corpora). This provides the adaptation to user information needs and gives a possibility to index and search not only full-text documents but short messages too, volumes of which are small and don't allow identifying their statistical characteristics.

The proposed algorithms are notable for universality, i. e. for the independence from topics. Information system adjustment to a particular data domain can be fully automated, it adds up to the creation of a respective thematic text corpus and an actualization of dictionaries of the knowledge database. At the same time, formation of the given dictionaries can completely be carried out in the hands off.

CONCLUSION

The algorithms presented in this paper can be used in various systems designed for processing and analyzing texts. The proposed technique of calculating the informativity of index terms can be used in automatic summarization systems for detecting informative wordforms in documents and for synthesizing connected summaries. With an appropriate selection of subjects and the hierarchical structure of a text corpus, it is possible to search regarding documents stylistic color (e.g. journalism, popular or scientific literature).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

АВТОМАТИЗАЦИЯ ПРОЦЕССОВ ВЫЯВЛЕНИЯ ТЕХНОЛОГИЧЕСКИХ ТРЕНДОВ В СИСТЕМЕ АРМ «ТРЕНД»

Хорошевский В.Ф.

Федеральное государственное бюджетное учреждение науки, Вычислительный центр им.

А.А. Дородницына РАН,

Центр информационно-аналитических систем ИСИЭЗ НИУ ВШЭ

г. Москва, Россия

khor@ccas.ru

vkhoroshevsky@hse.ru

В работе обсуждается АРМ «Тренд» - автоматизированное рабочее место поддержки процессов выявления новых технологических трендов на основе гибридного подхода к извлечению информации из текстов публикаций. Приводятся результаты обработки представительной коллекции документов, в которых представлены аннотации научно-технических статей.

Ключевые слова: технологический тренд; гибридный подход; лингво-статистический метод извлечения информации из текстов; автоматизированное рабочее место.

ВВЕДЕНИЕ

Общепризнанной «горячей» точкой в анализе тенденций научно-технического прогресса в настоящее время является выявление новых технологических трендов. На современном этапе работы в данной области концентрируются, в основном, на построении дорожных технологических карт с использованием методологий Форсайт-прогнозирования, а также построению паттернов данных и анализу временных рядов, специфицирующих существующие и прогнозируемые тенденции.

Настоящая работа концентрируется на обсуждении вопросов автоматизации процессов выявления новых технологических трендов на основе гибридного подхода к обработке документов, ориентированного на интеграцию классических методов прогнозирования и гибридных методов автоматической обработки корпусов текстов с использованием статистических методов и методов извлечения информации из текстов [Glance, et al., 2004; Daim et al., 2006; Shibata et al., 2008; Bagheri et al., 2009; Kim, et al., 2009; Wang et al., 2010].

При этом основные проблемы автоматизации выявления новых технологических трендов связаны с тем, что в рамках обработки соответствующих информационных ресурсов необходимо

- для каждого из жанров использовать собственные модели извлечения информации;
- интегрировать полученные частные результаты в рамках единой модели представления знаний и соответствующей системы алгоритмов постобработки.

С учетом вышесказанного в настоящем исследовании для выявления новых технологических трендов предлагается использовать гибридный подход, который, с одной стороны, развивает методы, предложенные в работах [Kim, et al., 2009; Wang, et al., 2010], а с другой – обобщает их на случай мультязычных коллекций документов различных жанров с активным использованием онтологических моделей, под управлением которых осуществляется автоматическое извлечение информации из тестов и формирование не просто “bag of words” для каждого текста, но обогащение характеристических векторов важными для предметной области ключевыми выражениями, которые формируются за счет использования специальных паттернов. Спецификой предлагаемого подхода является и то, что после предварительной статистической обработки текстов происходит автоматическое объединение полученных представлений для коллекций документов одного жанра, а также генерация OWL-представления экземплярной части онтологической модели тренда.

- различать тексты разных жанров;

1. Реализация гибридного подхода к выявлению трендов

1.1. Базовые гипотезы

В настоящем исследовании в процессе выявления новых технологических трендов используются следующие базовые гипотезы:

- Использование кривых Гартнера [GARTNER, 2012], где явно выделяются области «Технологический триггер», «Пик завышенных ожиданий», «Ущелье утраты иллюзий», «Склон осознания» и «Плато продуктивности», в качестве модели прогнозирования.
- Использование коллекций научно-технических публикаций в области охвата прогнозируемого тренда для анализа информации на уровне «Технологического триггера».
- Использование новостных сайтов по тематике исследуемого тренда, обработка которых, как правило, фиксирует всплеск интереса к новым технологическим трендам, для уровней «Пика завышенных ожиданий» и «Ущелья утраты иллюзий».
- Патентный анализ в области охвата прогнозируемого тренда для обработки информации на уровнях «Склона осознания» и «Плато продуктивности».
- Интеграция результатов обработки коллекций отдельных жанров на основе пересечения и/или объединения результатов статистической обработки отдельных коллекций.

1.2. Онтологические модели технологических трендов

Как известно, **тренд** (от англ. Trend) это долговременная общая тенденция изменения исследуемого временного ряда [Wiki, 2012]. Таким образом, базовым понятием тренда является понятие **тенденции** (от ср.-век. лат. tendentia - направленность), как направления развития какого-либо явления, мысли, идеи... При этом в модель тренда естественным образом вовлекается понятие **процесса** (действия по знач. глагола «направлять»), что, в свою очередь, предполагает наличие совокупности последовательных действий, направленных на достижение определенного результата. При этом, с учетом целей и задач настоящей работы, понятие тренда целесообразно конкретизировать следующим образом: **технологический тренд** – активно развивающееся в последние 5 лет технологическое направление, которое, как ожидается, продолжит свое активное развитие в ближайшие 10 лет.

С учетом этого в модель технологического тренда вводятся следующие дополнительные понятия: **технология** (совокупность методов и средств, направленных на достижение определенных целей); **инновационная деятельность** (процесс трансформации фундаментальных знаний в новые практические

приложения [Рудь и др., 2011]), **технологическая инновация** (конечный результат инновационной деятельности, получивший воплощение в виде нового или усовершенствованного продукта или услуги, внедренных на рынке, нового или усовершенствованного технологического процесса или способа производства (передачи) услуг, используемых в практической деятельности) и некоторые другие понятия.

В качестве модели технологических трендов в данной работе используются результаты ОКР, выполняемой в НИУ ВШЭ по Госконтракту № 07.524.12.4018, где на основании выявленной системы базовых понятий была построена и реализована в системе Protégé [Protégé, 2012] онтологическая модель технологического тренда.

Вместе с тем, следует отметить, что в настоящее время модели технологических трендов, адекватные целям настоящей работы, которые бы акцептовались специалистами в данной области, отсутствуют. И более того, среди специалистов нет единого мнения о том, какие индикаторы определяют наличие и/или отсутствие описания тренда в коллекциях документов. Учитывая это, в настоящей работе предлагается к понятию технологического тренда «идти» через онтологическую модель выделения в документах системы терминов, которые потенциально могут представлять тренд на основе гибридного подхода.

1.3. Система автоматизации процессов выявления технологических трендов

1.3.1. Общая архитектура системы АРМ «Тренд»

Как показывает анализ литературы по методам и средствам автоматизированного выявления технологических трендов [Nallpati, 2003; Gance, et al., 2004; Yoon, et al., 2004; Shibata, et al., 2008; Kim, et al., 2009], в общей схеме обсуждаемой в настоящей работе АРМ «Тренд» целесообразно выделить следующие этапы:

- Выявление центров компетенции (включая организации и авторские коллективы) в области охвата исследуемого тренда.
- Формирование коллекций документов, соответствующих исследуемому тренду с учетом выявленных на предыдущем этапе центров компетенции.
- Собственно обработка сформированных коллекций документов.

Методы и средства автоматизированного формирования экспертных групп и выявления центров экспертной компетенции в определенных предметных областях представлены в работе [Хорошевский, 2010], формирование коллекций документов, соответствующих исследуемому тренду, является темой отдельной работы. Поэтому ниже обсуждаются вопросы собственно обработки сформированных коллекций документов.

При этом основными функциональными подсистемами в АРМ «Тренд» являются:

- Подсистема предварительной обработки сформированной коллекции документов.
- Подсистема гибридной обработки отдельных документов коллекции.
- Подсистема интеграции результатов обработки всей коллекции документов.
- Подсистема генерации онтологического представления результатов обработки всех коллекций документов.
- Подсистема визуализации и анализа результатов.

С учетом вышесказанного, общая архитектура АРМ «Тренд» может быть представлена схемой, показанной на Рисунок 1, а ее основные подсистемы обсуждается в следующем подразделе настоящей работы.

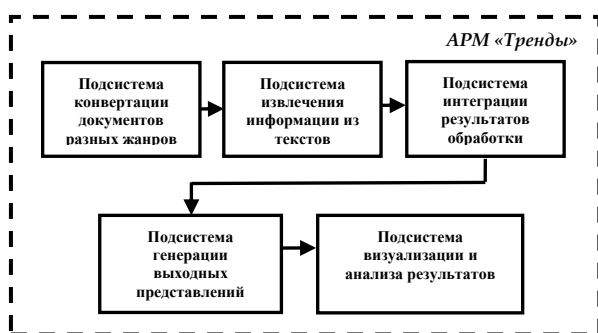


Рисунок 1 – Общая архитектура АРМ «Тренд»

1.3.2. Функциональные подсистемы АРМ «Тренд»

Подсистема предварительной обработки документов обеспечивает анализ форматов представления информации в сформированных коллекциях документов разных жанров и их планаризацию с помощью соответствующей системы конверторов.

Подсистема обработки отдельных документов коллекций на основе гибридного подхода предполагает использование лингвистических моделей извлечения информации из документов разных жанров и методов статистической обработки результатов извлечения информации из документов с помощью лингво-статистических процессоров, результатом работы которых являются характеристические вектора документов, соответствующие модифицированной модели «мешка слов» (Bag of Words).

В подсистеме интеграции результатов обработки всей коллекции документов осуществляется параметрическое слияние результатов обработки отдельных документов одной коллекции и результатов обработки разных коллекций.

Подсистема генерации выходных представлений результатов обработки всех коллекций документов поддерживает выбор выходного представления и собственно генерацию таких представлений в формате OWL и/или Tag Cloud.

Работа пользователя АРМ «Тренд» завершается в подсистеме визуализации и анализа результатов обработки коллекций документов.

Процессоры лингво-статистической обработки документов реализованы в АРМ «Тренд» с использованием инструментальной среды GATE [GATE, 2012], расширенной соответствующими модулями статистической обработки результатов извлечения терминов из текстов, а остальные подсистемы – в языке Java.

2. Выявление технологических трендов с использованием АРМ «Тренд»

2.1. Постановка задачи

Тестирование представленной выше АРМ «Тренд» осуществлялось в процессе обработки системы коллекций англоязычных документов, предоставленных ИСИЭЗ НИУ ВШЭ. Результат тестирования – информация о возможных технологических трендах, которые выявлены в процессе обработки исходных коллекций. Цель исследования – анализ полезности функционалов поддержки принятия решений экспертами-пользователями АРМ «Тренд» о присутствии в информационных материалах описаний технологических трендов.

2.2. Коллекции документов

Предоставленные для экспериментов исходные коллекции документов включали:

- аннотации научных статей (130867 док.),
- информацию из блогов (21 док.),
- информацию с новостных сайтов (560 док.),
- диссертации (29 док.) и некоторые другие коллекции.

Анализ предоставленных коллекций показал, что отдельные их документы, по сути дела, являются выгрузками из различных баз данных с различной структурой, не вполне соответствующих целям и задачам их последующей обработки. Поэтому для дальнейшего использования в АРМ «Тренд» было решено сконцентрироваться на коллекции аннотаций научных статей, предварительная планаризация которых была выполнена с помощью специально разработанного конвертора. В процессе предварительной обработки конвертор выделял в документе поля года публикации и аннотации публикации или, если таковое присутствует в документе, поле текста документа. Результатом работы конвертора было сохранение поля аннотации или текста публикации в отдельном планарном файле в коллекции, соответствующей году публикации. В случае отсутствия у документа поля года публикации, формировалась отдельная коллекция документов (Unknown) с неизвестным годом публикации.

2.3. Модуль гибридной обработки документов

Как показывает опыт создания систем извлечения информации из многоязычных коллекций документов [Efimenko, et al. 2009; Хорошевский, 2011], а также указанные выше цели разработки гибридного модуля формирования характеристических векторов текстов различных жанров, в данном случае целесообразно использовать следующую совокупность программных ресурсов обработки текстов:

- Лексическое форматирование.
- Морфологизация.
- Словарное означивание.
- Извлечение простых именных групп.
- Формирование системы терминов (однословных – Word и многословных – Expr).
- Предварительная статистическая обработка сформированной системы терминов.
- Формирование характеристических векторов документа (Bag of Words).
- Сохранение сформированных характеристических векторов в отдельных файлах.

В качестве инструментария для извлечения информации из текстов в настоящем исследовании использована платформа GATE, расширенная плагинами NP Chunker и Russian Morph Tagger [GATE, 2012], в котором используется открытая версия модуля русской морфологии компании Яндекс [Yandex, 2012].

Дополнительно к указанным выше компонентам извлечения информации из текстов для АРМ «Тренд» были разработаны специализированные ресурсы выделения простых именных групп в русскоязычных текстах, а также модуль генерации характеристических векторов документа (Bag of Words) и модуль сохранения сформированных характеристических векторов в отдельных файлах. Общие схемы цепочек ресурсов для обработки англоязычных и русскоязычных текстов представлены на рисунке 2.

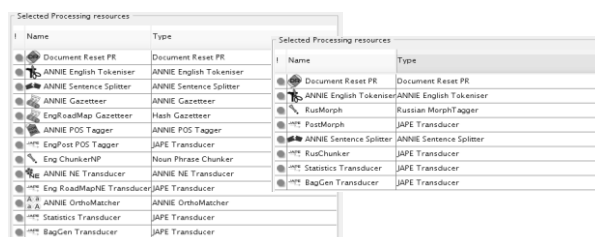


Рисунок 2 – Схемы цепочек ресурсов для обработки англоязычных и русскоязычных текстов

Для примера ниже показан фрагмент результатов работы модуля гибридной обработки англоязычного документа:

```
wos132472.txt
mesoporous_materials type=Expr;freq=1;TF=0.030303031
metal_oxides type=Expr;freq=1;TF=0.030303031
individual_polymer_chains type=Expr;freq=1;TF=0.030303031
fluorescent_dye type=Expr;freq=1;TF=0.030303031
```

```
oxide_nanostructures type=Expr;freq=1;TF=0.030303031
novel_nanocomposites type=Expr;freq=1;TF=0.030303031
molecular_imprinting type=Expr;freq=1;TF=0.030303031
shape_of_organic_molecules type=Expr;freq=1;TF=0.030303031
polyoxometalates type=Word;freq=1;TF=0.041666668
nanotechnology type=Word;freq=1;TF=0.041666668
```

Таким образом, на выходе модуля гибридной обработки документов формируются планарные файлы со следующей структурой:

- первая строка: имя файла (без расширения),
- каждая оставшаяся строка:
 - ключевое-слово | ключевое-выражение
 - type=Word | Expr,
 - freq=<целое> (количество повторений термина в документе),
 - TF=<десятичная-дробь> (частота встречаемости термина в документе).

2.4. Интеграция результатов обработки

Характеристические вектора, полученные в результате работы модуля гибридной обработки документов, обрабатываются специально разработанными и реализованными в рамках настоящего исследования модулями.

Первый модуль постобработки (ontoMerger) предназначен для интеграции характеристических векторов отдельных документов коллекции в единый характеристический вектор коллекции.

На вход данного модуля поступает коллекция характеристических векторов обработанных документов, а на выходе формируется единый файл характеристического вектора коллекции со следующей структурой:

- первая строка: N=<целое> (количество документов в коллекции),
- каждая оставшаяся строка:
 - ключевое-слово | ключевое-выражение
 - type=Word | Expr
 - DF=<целое> (количество документов в коллекции, в которых встречался термин);
 - TF=<десятичная-дробь> (средняя частота встречаемости термина в коллекции)

Второй модуль постобработки (ontoJoiner) предназначен для объединения характеристических векторов отдельных коллекций в единый характеристический вектор системы коллекций. На вход данного модуля поступают характеристические вектора обработанных коллекций, а на выходе формируется единый файл характеристического вектора системы коллекций. При этом в процессе объединения характеристических векторов отдельных коллекций происходит фильтрация терминов по параметру DF, что позволяет установить порог отсекаемых терминов, входящих в общий характеристический вектор системы коллекций.

Последний модуль постобработки предназначен для расчета индексов терминов, входящих в систему коллекций. На вход этого модуля поступает файл

характеристического вектора системы коллекций, а на выходе формируется индексный файл терминов коллекции со следующей структурой:

- первая строка: $N = \langle \text{целое} \rangle$ (количество документов во всех коллекциях),
- каждая оставшаяся строка:
 - ключевое-слово | ключевое-выражение
 - $\text{type} = \text{Word} | \text{Expr}$
 - $\text{DF} = \langle \text{целое} \rangle$ (количество документов во всех коллекциях, где встречался термин);
 - $\text{TF} = \langle \text{десятичная-дробь} \rangle$ (средняя частота встречаемости термина в системе коллекций);
 - $\text{IDF} = \langle \text{десятичная-дробь} \rangle$ (обратная частота термина на системе коллекций)
 - $\text{TF} * \text{IDF} = \langle \text{десятичная-дробь} \rangle$ (индекс термина в системе коллекций).

Таким образом, в процессе постобработки документов отдельных коллекций и систем коллекций формируется информация, достаточная для последующего анализа экспертами на предмет наличия в системе коллекций статистически значимой информации о новых технологических трендах.

2.5. Генерация онтологического представления результатов

Для удобства анализа результатов обработки коллекций документов в рамках настоящего исследования был разработан и реализован специальный модуль (OWL Generator), который базируется на идеях, представленных в работе [Witte, et al., 2010]. На вход этого модуля поступает характеристический вектор коллекции (системы коллекций), а на выходе формируется ее OWL-представление, соответствующее онтологической модели тренда.

Полученные OWL-представления загружаются в систему онтологического инжиниринга Protégé и используются экспертами для дальнейшего анализа и принятия решения о наличии/отсутствии в обработанной системе коллекций документов информации о новых технологических трендах.

3. Предварительные результаты и направления дальнейших исследований и разработок

3.1. Коллекции документов

Для проверки базовых гипотез, сформулированных выше, были сформированы 11 коллекций научных публикаций за 2002-2012 гг. общим объемом 130370 документов.

Каждая из коллекций была предварительно планаризована для дальнейшей обработки, каждый документ каждой коллекции был обработан с помощью реализованного гибридного модуля формирования характеристических векторов, а

результаты были обработаны реализованными модулями интеграции результатов.

Полученные характеристические вектора поступали на вход реализованного модуля генерации OWL-представлений, на выходе которого формировалась экземплярная часть OWL-представления коллекции. Валидация полученных результатов осуществлялась в системе Protégé, которая в данном случае использовалась в качестве инструментария онтологического инжиниринга.

3.2. Результаты обработки коллекций документов

После обработки всех коллекций были получены результаты, представленные в Таблица 1.

Таблица 1 – Статистика коллекций научных публикаций

Год	К-во док.	Терминов типа Word		Терминов типа Expr	
		Уник.	Всего	Уник.	Всего
2002	5762	4541	10729	13576	110785
2003	5916	4685	11065	13996	114314
2004	5885	4625	10898	13888	111987
2005	8976	6874	16018	28872	197315
2006	12136	8322	19058	38024	261994
2007	13038	8608	20243	39122	283060
2008	14031	8972	21047	40490	301267
2009	15181	9832	21290	59882	307716
2010	15748	9346	22682	40834	339256
2011	19053	11241	24700	71941	383765
2012	16145	12865	17926	140489	244760
Всего:	131871		184927		2545434

Для случая стандартных алгоритмов обработки документов всех коллекций распределение терминов представлено на рисунке 3, а для случая обработки с учетом только тех предложений, которые содержат индикаторы присутствия трендов в документах, - на Рисунке 4.

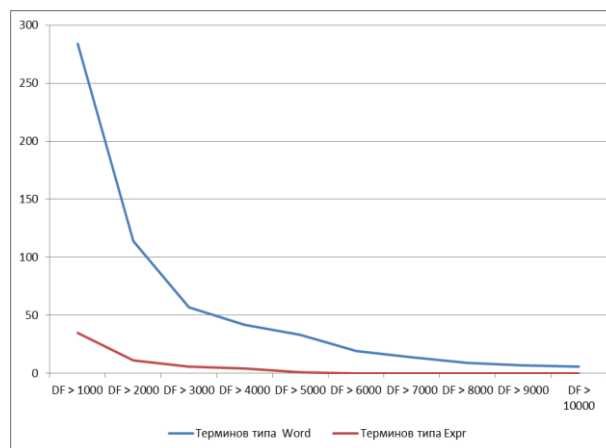


Рисунок 3 - Распределение терминов для случая стандартной обработки документов

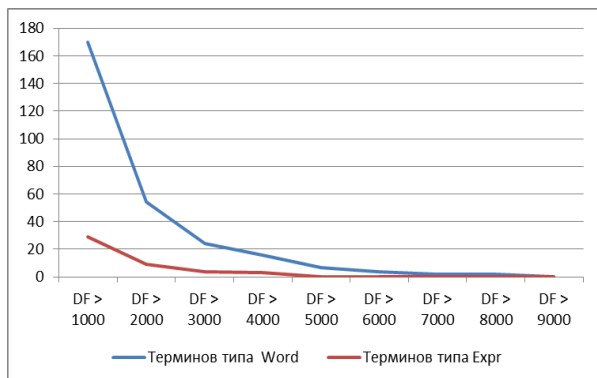


Рисунок 4 - Распределение терминов для случая гибридной обработки документов с учетом индикативных предложений

Для примера, на Рисунок 5 приведено OWL-представление результатов обработки коллекций научных публикаций за 2002-2012 г.г. (DF>1500), а на Рисунок 6 - OWL-представление результатов обработки тех же коллекций для случая DF>2500.

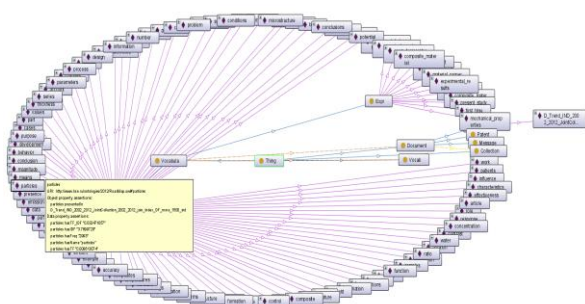


Рисунок 5 - OWL-представление коллекций научных публикаций за 2002-2012 гг. (DF>1500) в системе Protege

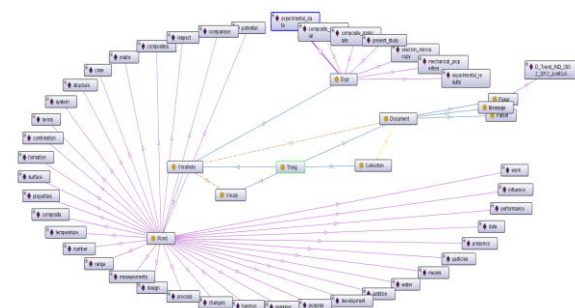


Рисунок 6 - OWL-представление коллекций научных публикаций за 2002-2012 гг. (DF>2500) в системе Protege

Как показывает анализ представленных выше результатов, в случае использования алгоритмов гибридной обработки документов с учетом только индикативных предложений количество выделенных терминов существенно меньше, чем в случае стандартных алгоритмов. Таким образом, использование алгоритмов, в которых обрабатывались только предложения, содержащие индикаторы возможного присутствия терминов трендов, позволяет существенно сократить объемы характеристических векторов коллекций, а динамика уменьшения числа выделенных терминов показывает более быстрое их сокращение. Вместе с тем, как показывает более детальный анализ результатов обработки одних и тех же коллекций научных публикаций, уменьшение числа обрабатываемых предложений в обеих схемах

примерно одинаково. На наш взгляд, такая ситуация определяется тем, что коллекция научных публикаций в данном случае была представлена их аннотациями, в которых практически все предложения должны быть значимыми. В связи с этим представляет интерес дальнейший анализ коллекций полных текстов научных публикаций и сравнение результатов с результатами анализа аннотаций.

Особый интерес представляет анализ состава выделенных терминов в коллекциях научных публикаций. Для проведения такого анализа ниже представлены однословные (Таблица 2) и многословные (Таблица 3) термины для случая DF > 1000.

Таблица 2 – Термины типа Word (коллекции 2002-2012 г.г., DF > 1000, Количество документов в коллекции – 129833)

№/№	Key	DF	TF	IDF	TF*IDF
1.	ability	2253	0.005	4.053	0.023
2.	advantages	1048	0.003	4.819	0.016
3.	agreement	1034	0.003	4.832	0.016
4.	aircraft	1288	0.003	4.613	0.015
5.	carbon	1135	0.002	4.739	0.013
6.	composite	5169	0.016	3.223	0.052
.....					
	microstructure	2091	0.006	4.128	0.025
	nanocomposites	1316	0.003	4.591	0.016
	nanoparticles	1203	0.002	4.681	0.013
.....					
178.	water	3006	0.007	3.765	0.027

Таблица 3 – Термины типа Expr (коллекции 2002-2012 г.г., DF > 1000, Количество документов в коллекции – 129833)

№/№	Key	DF	TF	IDF	TF*IDF
1.	carbon_nanotubes	1100	0.001	4.770	0.007
2.	composite_material	2728	0.005	3.862	0.020
3.	composite_materials	4842	0.008	3.288	0.029
4.	electron_microscopy	2557	0.003	3.927	0.014
5.	tensile_strength	1091	0.001	4.779	0.008
6.	thermal_conductivity	1151	0.001	4.725	0.008
7.	thermal_stability	1102	0.001	4.769	0.008
8.	thermogravimetric_analysis	1056	0.001	4.811	0.007
9.	transmission_electron_microscopy	1143	0.001	4.732	0.007
10.	x_ray_diffraction	2177	0.002	4.088	0.012

Как показывает анализ приведенных данных, среди терминов типа Word встречаются как интересные с точки зрения выявления новых

технологических трендов понятия, так и термины общей лексики.

Иная ситуация наблюдается в случае терминов типа Exrg. Здесь практически все многословные термины представляют для экспертов интерес в плане выявления технологических трендов.

Дальнейший анализ полученных результатов осуществлялся путем построения временных рядов частоты появления многословных терминов в коллекциях документов разных лет по параметрам DF (Рисунок 7) и TF*IDF (Рисунок 8).

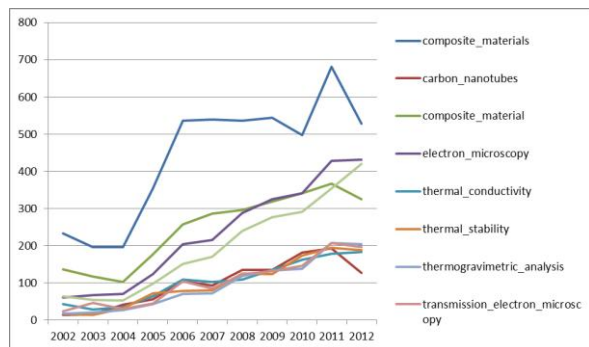


Рисунок 7 – Временной ряд частоты появления многословных терминов в коллекциях научных публикаций за 2002-2012 г.г. по параметрам DF

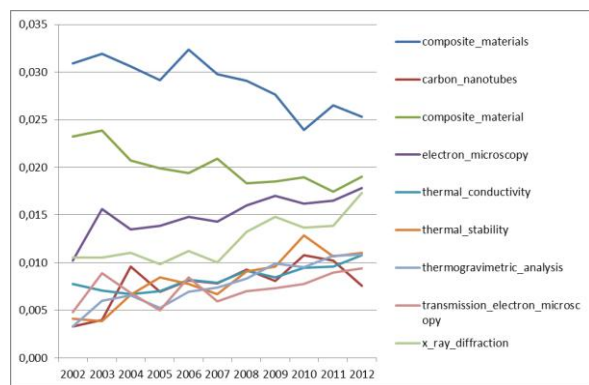


Рисунок 8 – Временной ряд частоты появления многословных терминов в коллекциях научных публикаций за 2002-2012 г.г. по параметрам TF*IDF

Как показывает анализ приведенных выше данных, по частоте встречаемости практически все многословные термины демонстрируют «бычий» тренд. В то же время, по параметру TF*IDF некоторые из многословных терминов демонстрируют «бычий» тренд, а другие – «медвежий» тренд.

Обсуждение полученных результатов с экспертами показало, что термины «бычьего» тренда могут идентифицировать ситуацию появления нового технологического тренда в области технологического триггера, а «медвежьего» – выход соответствующих результатов на плато продуктивности.

Таким образом, можно констатировать, что АРМ «Тренд» является полезным инструментом помощи экспертам в выявлении новых технологических трендов за счет автоматизации процессов обработки

и анализа результатов на больших коллекциях документов.

Заключение

В работе рассмотрены вопросы разработки и реализации автоматизированного рабочего места поддержки процессов выявления новых технологических трендов АРМ «Тренд». Приведены результаты обработки представительной коллекции документов, в которых представлены аннотации научно-технических статей.

Предполагается, что направления дальнейших работ по автоматизации процессов выявления новых технологических трендов будут связаны с анализом групп индикаторов с точки зрения их эффективности и значимости для различных жанров документов и частотным анализом словосочетаний и шаблонов индикаторов; доработкой онтологических моделей трендов, задающих семантические поля индикаторов и стоп-слов; автоматизацией процессов анализа динамики изменения терминологического поля, описывающего технологический тренд; интеграцией программного обеспечения АРМ «Тренд» с инновационными средствами визуализации результатов анализа.

Благодарности. Работа выполнена при финансовой поддержке Минобрнауки России по государственному контракту от 16.05.2012 г. № 07.524.12.4018 в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы».

Библиографический список

- [Рудь и др., 2011] Рудь В.А., Фурсов К.С. Роль статистики в дискуссии о научно-технологическом и инновационном развитии. // Вопросы экономики, 2011. № 1. С. 120—133.
- [Хорошевский, 2010] Хорошевский В.Ф., Извлечение информации из текстов на конференциях серии ДИАЛОГ: взгляд соседа по лестничной клетке. // Труды международной конференции "Диалог 2010" М. Наука – 2010.
- [Хорошевский, 2011] Хорошевский В.Ф., Пространства знаний в сети Интернет и Semantic Web (Часть 3), Искусственный Интеллект и Принятие решений, № 2 (2011).
- [Bagheri et al., 2009] Bagheri S. K., Nilforoushan H., Rezapour M., Rashtchi M., A new approach to Technology Roadmapping in the Open Innovation context: The Case of Membrane Technology for RPI, Journal of Science & Technology Policy, Vol. 2, N 1, Spring 2009.
- [Daim et al., 2006] T.U. Daim, G. Rueda, H. Martin, Forecasting emerging technologies: use of bibliometrics and patent analysis, Technol. Forecast. Soc. Change, vol. 73, N 8, 2006.
- [Efimenko, et al. 2009] Efimenko I., Minor S., Starostin A., Drobnyazko G., Khoroshevsky V., Generating Semantic Content for the Next Generation Web, Chapter in Monograph "Semantic Web", Publisher IN-TECH, 2009, ISBN 978-953-7619-33-6
- [GARTNER, 2012] Gartner home page, URL: <http://www.gartner.com/technology/research.jsp>
- [GATE, 2012] Developing Language Processing Components with GATE Version 7 (a User Guide). <http://gate.ac.uk/sale/tao>
- [Glance, et al., 2004] Natalie S. Glance, Matthew Hurst, Takashi Tomokiyo. BlogPulse: Automated trend discovery for weblogs // WWW 2004, Workshop on the weblogging ecosystem: aggregation, analysis and dynamics, ACM, 2004.

[**Kim, et al., 2009**] Youngho Kim, Yingshi Tian, Yoonjae Jeong, Ryu Jihee, Sung-Hyon Myaeng. Automatic Discovery of Technology Trends from Patent Text. In: Proc, SAC'09, March 8-12, 2009, Honolulu, Hawaii, U.S.A., 2009.

[**Nallpati, 2003**] R. Nallpati. Semantic language models for topic detection and tracking. In Proceedings of the conference of the North American chapter of the Association for Computational Linguistics on Human Language Technology (HLTNAACL' 03), 2003.

[**Protégé, 2012**] Protege Homepage, <http://protege.stanford.edu/>

[**Shibata et al., 2008**] Shibata et al. Detecting emerging research fronts based on topological measures in citation networks of scientific publications, Technovation, N 28 (2008).

[**Wang et al., 2010**] Wang et al. Identifying technology trends for RD planning using TRIZ and text mining, RD Management, vol. 40, N 5, 2010.

[**Wiki, 2012**] Понятие тренда. <http://ru.wikipedia.org/wiki/Тренд>

[**Witte, et al. 2010**] Witte R., Khamis N., Rilling J., Flexible ontology population from text: The owl exporter. In International Conference on Language Resources and Evaluation (LREC), Valletta, Malta, 05/2010 2010.

[**Yandex, 2012**] Морфология Яндекса. <http://company.yandex.ru/technology/mystem/>

[**Yoon, et al., 2004**] B. Yoon, and Y. Park. A text mining-based patent network: analytical tool for high-technology trend. Journal of High Technology Management Research, Vol. 15 (1), 2004.

TECHNOLOGY TRENDS WATCHING IN AUTOMATED WORKSTATION "TREND"

Khoroshevsky V. F.

*Institution of Russian Academy of Sciences
Dorodnicyn Computing Centre of RAS,
Center for Information Intelligence Applications of
Institute for Statistical Studies and Economics of
Knowledge, NU HSE
Moscow, Russia
khor@ccas.ru
vkhoroshevsky@hse.ru*

Workstation ARM "Trend" oriented to the support of new technological trends watching based on a hybrid approach to information extracting is presented in the paper. The results of the processing of a representative collection of scientific and technical papers annotations are discussed.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ГИБРИДНЫЙ ПОДХОД К ВЫЯВЛЕНИЮ КОМПЛЕКСНЫХ ОБЪЕКТОВ В ОБЛАСТИ НАУЧНО-ТЕХНИЧЕСКОГО ПРОГНОЗИРОВАНИЯ: ПРИНЦИП «ЧЕРНОГО ЯЩИКА»

Ефименко И.В.

Центр информационно-аналитических систем ИСИЭЗ НИУ ВШЭ

г. Москва, Россия

iefimenko@hse.ru

В работе обсуждаются методы и алгоритмы идентификации в текстах объектов высокого уровня концептуальной сложности, значимых для области научно-технического прогнозирования. Приводятся результаты применения представленных методов, которые используются, в частности, при создании программного комплекса «Интерактивная дорожная карта с обратной связью».

Ключевые слова: дорожная карта; научно-техническое прогнозирование; технологический тренд; гибридный подход; лингво-статистический метод обработки текстов; идентификация объектов в текстах.

ВВЕДЕНИЕ

Настоящая статья посвящена методам и алгоритмам идентификации в текстах объектов, значимых для области научно-технического прогнозирования. Представленные методы и алгоритмы применяются, в частности, при создании программного комплекса «Интерактивная дорожная карта с обратной связью».

Дорожная карта (ДК) — форма наглядного визуального представления многоуровневой системы стратегического развития предметной области в рамках единой временной шкалы, которая содержит показатели экономической эффективности перспективных технологий и продуктов, обладающих высоким потенциалом спроса и привлекательными потребительскими свойствами. ДК иллюстрирует взаимосвязи между основными технологиями, определяющими развитие предметной области, существующими и перспективными продуктами и их характеристиками, динамикой российского и мирового рынков в контексте научно-технологического развития (далее – НТР).

ДК формируется в виде набора слоев. Форма представления, состав элементов ДК отличаются в зависимости от требований заказчика и предметной области картирования. Информационное наполнение элементов и взаимосвязи между ними определяются (рассчитываются) базовыми алгоритмами расчета элементов или алгоритмами

консолидации экспертных сведений. Для разработки каждой ДК привлекается от десятков до нескольких сотен экспертов предметной области.

В качестве слоев дорожной карты в общем случае выступают: 1) научно-технологическое развитие; 2) технологии; 3) продукты; 4) рынки. В некоторых случаях при разработке дорожной карты формируется дополнительный слой: 5) глобальный контекст развития.

Для всех слоев используется один и тот же горизонт планирования (от 20 лет). Как правило, на одном слое ДК размещается 5-7 элементов.

1. Принцип «черного ящика»

В целях реализации описываемых методов и алгоритмов был предложен подход, в основе которого лежит принцип «черного ящика». Суть подхода состоит в следующем.

Предметная область научно-технического прогнозирования (далее – НТП) характеризуется высокой степенью сложности – и для случаев, когда речь идет об исследовании той или иной предметной области с применением экспертных процедур, т.е. вручную и с использованием специальных знаний, и, тем более, для сценариев, основанных на применении средств автоматизации. Информационные объекты, относящиеся к сфере НТП (элементы дорожных карт, технологические тренды и др.), плохо поддаются автоматической идентификации в документах в силу:

- Высокого уровня концептуальной сложности, комплексной природы;
- Низкого уровня формализации (так, например, фактически отсутствуют однозначные и общепринятые определения ряда явлений; для значительного числа объектов в области НТП представляется затруднительным дать определения, которые можно считать формальными);
- Высокой степени зависимости от специфики конкретной предметной области (нанотехнологии, медицина, транспорт, энергетика, информационные технологии и т.п.) и, как следствие, необходимости применения специальных (профессиональных) знаний, обеспечивающих возможность корректной интерпретации информации.

При этом однотипные объекты даже в далеких друг от друга предметных областях обладают родственными чертами, позволяющими отнести соответствующие явления к одному и тому же концепту сфера НТП. Таким образом, внешнее поведение однотипных объектов для различных предметных областей существенно более универсально, чем сами объекты, и можно говорить о «признаках присутствия» объекта НТП, по которым его следует идентифицировать. Сами же объекты на этапе анализа могут оставаться для наблюдателя «черным ящиком».

Аналогичный принцип проявляется также в следующем: потребностей и проблем, на решение которых нацелен процесс НТР, в каждой предметной области существенно меньше, чем способов их решения. На глобальном уровне можно говорить о незначительном по объему перечне ценностей, на обеспечение которых направлено развитие во всех предметных областях (стабильность общества, здоровье человека и др.). Также универсальными являются параметры, изменение значений которых (в сторону повышения или понижения, в зависимости от специфики параметра) является целью разработки и внедрения новых технологий: стоимость (положительным эффектом считается снижение стоимости производства и использования), эффективность (новые технологии призваны, среди прочего, повышать эффективность чего-либо), размер (в зависимости от ситуации желательно уменьшение или увеличение размеров тех или иных объектов), безопасность (может восприниматься и как глобальная ценность, и как параметр, на увеличение значения которого должно быть направлено НТР в любой области) и т.п.

Методы выявления объектов НТП, разработанные с учетом отмеченных выше особенностей, позволяют избежать таких проблем, как существенная зависимость от предметной области и др., и сформировать универсальный подход к обработке текстов в целях информационно-аналитической поддержки процессов научно-технического прогнозирования.

В результате выполненных работ была сформирована система онтологических моделей для предметной области НТП, специфицирующая как ключевые типы объектов для предметной области дорожного картирования и НТП в целом, отношения между такими объектами и их атрибуты, так и сопутствующие явления и процессы, которые задают систему индикаторов для объектов НТП в концепции «черного ящика».

2. Общий алгоритм. Модификация статистических методов с целью разработки гибридного подхода

С учетом постоянного развития науки и техники, появления новых направлений исследований, технологий, продуктов, рынков и т.п., для обработки документов с целью выявления объектов НТП представляется целесообразным использовать статистические методы. В качестве основы могут быть использованы классические методы анализа n-грамм с применением метрик частоты (term frequency, TF) и обратной частоты (inverse document frequency, IDF). Однако опыт показывает, что для сложных задач и комплексных, плохо формализуемых предметных областей применение классических статистических методов часто приводит к означиванию ложных групп слов. Такого рода эффект не удается преодолеть ни с использованием стоп-слов (списки стоп-слов должны быть настолько большими и постоянно пополняться и модифицироваться вручную с учетом специфики коллекции – жанровой, по предметной области и т.п., – что теряется смысл автоматизации процессов анализа), ни с помощью введения метрики IDF. Цель IDF состоит в уменьшении веса широкоупотребительных слов, однако в ряде случаев речь идет о словах и словосочетаниях, не являющихся в общем смысле широкоупотребительными. Их частота в коллекции такова, что использование меры TF-IDF приводит к попаданию их в число значимых словосочетаний, однако такого рода словосочетания не являются желательными результатами. В качестве примера можно привести термины, обладающие специфичной для предметной области, но слишком широкой семантикой (например, термины типа architecture, warehouse и т.п. для предметной области «Информационные технологии»).

С целью уменьшения уровня шума целесообразно ввести в общий алгоритм анализа документов следующие этапы:

- Морфологический анализ документа. При этом обеспечивается означивание слов по частям речи и морфологическим формам;
- Синтаксический анализ. Выполняется на уровне отдельных элементов, без полномасштабного построения лингвистических деревьев. В частности, обозначаются однословные и многословные именные группы;

- Выбор из документов наиболее значимых фрагментов. Настоящая процедура реализуется следующим образом. На основании индикаторов объектов НТП, заданных в онтологиях, обеспечивается выбор предложений и абзацев для последующего анализа, при этом наличие в выбранном фрагменте отдельного индикатора или их совокупности является гарантом релевантности данного фрагмента для задачи идентификации объекта НТП.

Предусматривается обработка документов на английском и на русском языках.

В результате обеспечивается:

- Анализ отдельных слов и многословных словосочетаний, гарантированно являющихся именными группами, т.е. потенциальными терминами, и отсеивание до проведения анализа биграмм и триграмм, не являющихся таковыми. Кроме того, сборка словосочетаний на основе лингвистических признаков позволяет не задавать ограничений на их количество. Таким образом, обеспечивается обработка терминов любой длины;

- Отсеивание менее значимых с точки зрения задачи выявления элементов дорожных карт и других объектов НТП частей текста и выполнение анализа на наиболее релевантных фрагментах, что позволяет включать в выдачу только словосочетания, потенциально относящиеся к наименованию объекта НТП или его атрибутам.

Для полученных на выходе словосочетаний выполняется статистический анализ. В ряде случаев, например, для идентификации технологических трендов, необходим учет временного аспекта, динамики появления терминов. В этом случае формируются отдельные коллекции, структурированные по периодам времени, например, по годам. Для каждой коллекции формируются результаты, которые затем подвергаются обработке с целью выявления динамики употребления терминов.

Таким образом, предлагаемый алгоритм сочетает в себе черты лингвистического и статистического подходов, его общая схема включает следующие шаги:

- Морфологический анализ текста. При этом для английского языка используется модуль морфологического анализа платформы с открытым кодом (open source) GATE – General Architecture for Text Engineering, – разработанной в Шеффилдском университете, Великобритания. Для русского языка используется модуль морфологического анализа, разработанный компанией Яндекс;

- Синтаксический анализ, основной целью которого является сборка именных групп. При этом используются модули GATE, доработанные в Центре информационно-аналитических систем Института статистических исследований и экономики знаний (ЦИАС ИСИЭЗ) НИУ ВШЭ;

- Выбор релевантных фрагментов документа с использованием системы индикаторов, т.е. фрагментов, с наибольшей вероятностью содержащие информацию об объектах НТП (разработка ЦИАС ИСИЭЗ с использованием редактора онтологий Protégé – для спецификации моделей, платформы GATE – для генерации специальных аннотаций, среды IntelliJIDEA – для создания системы индикаторов и правил их идентификации);

- Выбор именных групп на заданных фрагментах документа;

- Статистический анализ именных групп (разработка ЦИАС ИСИЭЗ с использованием среды IntelliJIDEA);

- Выдача результатов, обеспечивающая возможность анализа динамики (разработка ЦИАС ИСИЭЗ);

- Представление результатов для интерпретации экспертам в предметной области.

В последующих пунктах более детально описаны некоторые из представленных выше шагов алгоритма.

Сведения о реализации алгоритма представлены в статье [Хорошевский, 2013].

3. Индикаторы объектов НТП

3.1. Система эвристик

Как уже указывалось выше, цель поиска в тексте индикаторов состоит в выявлении в документах объектов НТП по «внешним проявлениям». Индикаторы позволяют также выбрать наиболее релевантные фрагменты документов коллекции, на которых затем может выполняться анализ терминов, в т.ч., статистический. Целесообразно ввести параметр веса для отдельных индикаторов и их сочетаний. Таким образом, становится возможным применение подхода на предобработанных, искусственно сформированных коллекциях документов – в зависимости от веса самого индикатора или задающего его лингвистического шаблона идентифицируются:

- Непосредственно информационные объекты из контекста индикатора;

- Наиболее релевантные фрагменты текста (предложения, абзацы).

При этом применяется система эвристик о наличии в анализируемом контексте объекта НТП или, в случае комплексного объекта, его отдельных составляющих.

Все эвристики могут быть сгруппированы по следующим уровням (рисунок 1):

- Внешний контекст (экстралингвистические признаки);

- Метаинформация на уровне коллекций документов;

- Данные о структуре документа;

- Лингвистические маркеры в теле документа.

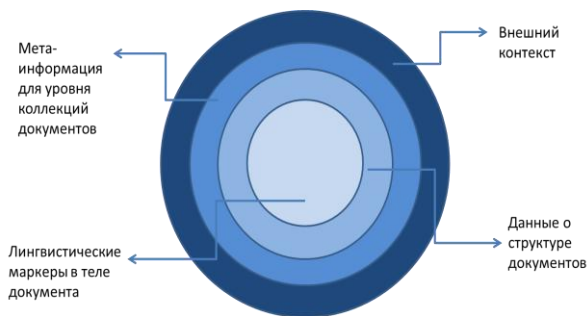


Рисунок 1 – Схема системы эвристик для выявления технологических трендов

3.2. Лингвистические маркеры в теле документа

Ключевым типом индикаторов для объектов НТП являются лингвистические маркеры в теле документа. Такого рода маркеры формируют семантические поля, каждое из которых соответствует определенному фрагменту системы онтологических моделей (например, для онтологии «Исследования и разработки» фиксируются стадии научно-исследовательского и разработческого процессов, основные типы получаемых результатов и т.п.).

Некоторые примеры семантических полей, формирующих группы индикаторов, представлены ниже.

4. Лингвистические шкалы

Одним их ключевых индикаторов релевантности фрагмента документа предметной области НТП является наличие в тексте упоминания движения того или иного параметра по лингвистической шкале, т.е. по шкале оценок типа «хорошо/плохо», «быстро/медленно», «сильно/слабо», «много/мало» и др. Это обусловлено тем, что при описании инновации, новой технологии, технологического тренда и т.п., обязательно фиксируются характеристики объектов, состояние которых изменяется в результате такого развития, указываются параметры, значение которых удалось улучшить, и т.п.

В результате анализа были выделены два основных супертипа объектов, в отношении которых предполагается проводить анализ оценочной информации в текстах, основанный на использовании лингвистических шкал:

- Потребности (человека, общества);
- Собственно объекты НТР (направления исследований и разработок, технологии, продукты, рынки и т.п.), т.е. объекты, которые создаются или трансформируются в результате НТР.

Более детально соответствующая проблематика освещена в работе [Ефименко, 2012].

Примеры, приведенные на рисунке 2, показывают, что лингвистические шкалы используются в текстах для концептов различных типов, в частности:

- Решаемые «общечеловеческие» проблемы, потребности;
- Ресурсы, ценности;
- Проблемы научного и технологического характера;
- Технологические параметры.

Все они, так или иначе, сводятся к характеристикам двух основных супертипов, представленных выше (потребности человека и общества; объекты НТР).

При этом, например, для технологических параметров выстраивается следующая модель: собственно технологические параметры (пример выделен на рисунке голубым), внешние характеристики, на достижение которых направлена технология (пример выделен зеленым), характеристики, апеллирующие к ценностям, к потребностям, ради удовлетворения которых создается технология (пример выделен красным).

- ❖ The US Export-Import Bank is to sign a \$2 billion deal with South Africa to fund a green energy scheme in the electricity-short country.
- ❖ Defects that are of zero or very small volume, known as kissing defects, are much harder to locate.
- ❖ There is considerably less understanding about the performance of IBs.
- ❖ Remarkable reduction in tensile, flexural and interlaminar properties was noticed after 2 weeks of immersion for all three materials.
- ❖ When increasing the external magnetic field by only 50 mTesla the dynamic stiffness for isotropic samples increased by 100 % while the damping factor decreased by 17%.
- ❖ For the marine industry where unpredictable dynamic loading conditions are the case, MRE isolators could greatly decrease the level of vibrations transmitted from the machines to the shell of the ship and the opposite, resulting to smaller fatigue loads and a much more comfortable journey.

Рисунок 2 – Примеры лингвистических шкал в текстах научных статей

На рисунке также выделены слова и словосочетания (синим шрифтом), задающие оценку степени изменения, которые также могут быть использованы в качестве индикаторов (в сочетании с индикаторами, фиксирующими наличие движения по лингвистической шкале, или самостоятельно).

В зависимости от параметра, значение которого изменяется по лингвистической шкале, а также от нюансов семантики самого лингвистического маркера, движение по шкале в каждом из направлений может интерпретироваться и как положительный, и как отрицательный результат (снижение стоимости vs. снижение эффективности; impairment или depletion vs. remission), при этом интерес для анализа представляет каждый из вариантов. В общем случае, в первой категории (положительный результат) имеет место фиксация нового решения – продукта, технологии и т.п. Во втором варианте (отрицательный результат) можно говорить о возникновении новой потребности или (технологической) проблемы.

Использование лингвистических шкал может обеспечить выявление технологических трендов на

различной стадии их развития, в т.ч., на начальных этапах (алгоритм будет представлен в докладе).

5. Другие типы индикаторов

Другие типы индикаторов для объектов НТП, разработанные к настоящему моменту, объединены в следующие модели:

- «Исследования и разработки»;
- «Инновации»;
- «Внедрение и производство»;
- «Технологии»;
- «Лакуны в НТР»;
- «Ценности и потребности»;
- «Стандарты и регулирование»;
- «Уполномоченные органы, инстанции»;
- «Коммерческие аспекты»;
- «Мероприятия»;
- «Абстрактные объекты» (элементы

процесса рассуждения, объяснения, анализа, такие как причина, следствие, цель, влияние, неизвестность, противопоставления, ограничения и др.).

Примеры фрагментов текста, обрабатываемых системой в соответствии с представленными системами индикаторов, представлены ниже (фрагменты, являющиеся основой для применения индикатора той или иной группы, выделены подчеркиванием):

- The use of adhesive bonds in engineering and marine structures is currently hindered by a lack of knowledge of joint reliability;
- Initial experimental work is presented on... Tests were performed...;
- An extensive study on the capabilities of marine animals has been conducted in relation to the equivalent functionalities in AUVs;
- The principal focus is on biological solutions to depth, speed, agility and endurance capabilities;
- PPT has shown promise as an effective near surface non-destructive evaluation (NDE) technique in a range of applications; focuses on the implementation of PPT on solid materials with artificial defects.
- Many of these have capabilities and functionality which have much in common with the engineered capabilities required for underwater vehicles e.g. propulsion/locomotion, manoeuvrability/agility and the ability & resilience to operate at depth;
- Indeed, in many examples, it appears the biological solutions exhibit superior performance compared to the technological alternative, yet by different and diverse means;
- Results from PPT are compared to other NDE methods such as water coupled ultrasound to ensure the PPT could produce results that are comparable to a more established technique;
- To be ready for production by next year, the company will offer the NM82/1500 designed specifically for low and medium wind zone onshore installations. With noise becoming an issue for turbines

installed near built-up areas, and regulations already in place to limit that noise, the new design will have the ability to switch automatically to a quieter mode of operation to meet limits at certain periods of the day;

- The UK and the World can no longer afford to neglect the massive potential of wave and tidal energy, said Dr. Desmond Turner announcing the publication of a report on ocean energy prepared by the Science and Technology Committee of the British Parliament

Результаты исследования показывают, что различные типы индикаторов коррелируют с различными жанрами текстов. При этом ряд индикаторов имеют универсальный характер (например, лингвистические шкалы, ценности), другие являются жанрово зависимыми.

В дополнение к основным моделям вводятся вспомогательные, которые, в зависимости от решаемой задачи и конкретного шага алгоритма, могут быть использованы или для анализа широкого контекста объектов НТП (например, для элементов глобального контекста для дорожных карт, таких как «Барьер», «Окно возможностей» и т.п.; для нетехнологического – экономического, политического, социального – контекста технологических трендов), или в качестве источника стоп-слов и выражений (семантические стоп-поля). Примерами таких моделей являются:

- «Экономика»;
- «Политика»;
- «Общество» («Роли»);
- «Время»;
- «Единицы измерения».

Кроме того, при выявлении объектов НТП, относящихся к определенной стадии НТР, в качестве стоп-полей могут быть использованы множества концептов основных моделей, характеризующих другую стадию НТР (например, «Внедрение и производство» для «Исследований и разработок»).

Из вышесказанного следует, что удаление терминов, относящихся к стоп-полям, является опциональным.

Они могут быть также использованы позднее как фильтр для готовых результатов. При этом их включение в результаты предоставляет дополнительные возможности, в частности, их наличие:

- Потенциально свидетельствует о стадии НТР;
- Является вспомогательной информацией для развития модели, поскольку помогает решить задачу отображения индикаторов на этапы НТР (например, с использованием кривых Гартнера);
- Позволяет идентифицировать явления пограничного характера (см. выше).

Базовая версия разрабатываемого программного обеспечения основана на использовании индикаторов, являющихся общими для различных

предметных областей. При этом в качестве опциональной стадии можно ввести предобработку перечня индикаторов экспертами в предметной области, выбранной для анализа, с целью:

- Введения дополнительных, специфичных для предметной области терминов, которые могут играть роль индикаторов (включение на этом этапе более сложных конструкций, чем отдельные термины, требует лингвистической компетенции и представляется нецелесообразным; при этом допустимы многословные термины, для которых будет обеспечен морфологический анализ);
- Удаления индикаторов, являющихся общеупотребительными, но имеющих дополнительное, специфичное значение в рамках соответствующей предметной области, характеризующихся высоким уровнем многозначности.

В составе алгоритма также предусмотрены методы, обеспечивающие автоматизированный анализ синонимических рядов и родственных объектов (например, терминов, относящихся к различным явлениям, но характеризующих одну и ту же технологию).

В зависимости от особенностей коллекций могут быть применены отдельные типы индикаторов или их совокупности. При этом роль играет не только жанр, но и другие аспекты, например, объем коллекции. Так, на больших коллекциях представляется целесообразным анализ документов по отдельным группам индикаторов с последующим поиском пересечений (внутри типа и между типами).

Заключение

В качестве направлений дальнейшего исследования и разработки для развития методов и средств, представленных в статье (в т.ч., с выходом за рамки работ по созданию программного комплекса «Интерактивная дорожная карта с обратной связью»), планируются следующие:

- Анализ групп индикаторов с точки зрения их эффективности и значимости для различных жанров документов;
- Частотный анализ словосочетаний и шаблонов индикаторов применительно к различным объектам НТП, в частности, для различных слоев ДК;
- Доработка онтологических моделей, задающих семантические поля индикаторов и стоп-слов;
- Автоматизация анализа динамики изменения объектов НТП;
- Соотнесение типов индикаторов с этапами (стадиями) НТП, Рис. 3;

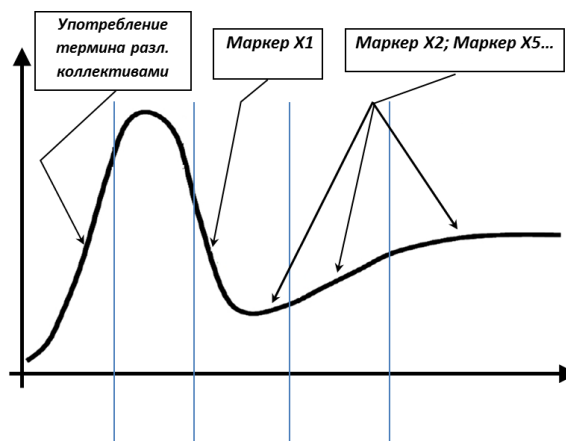


Рисунок 3 – Соотнесение индикаторов с этапами НТП (на примере кривой Гартнера)

- Уточнение формальной модели комплексных, слабо формализуемых объектов в области НТП, в т.ч., с учетом возможности формирования иерархий, декомпозиции, циклического характера развития;
- Разработка методов квантификации качественных оценок для времени, вероятности (plausibility), других типов оценок.

Благодарности. Работа выполнена при финансовой поддержке Минобрнауки России по государственному контракту от 16.05.2012 г. № 07.524.12.4018 в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Ефименко, 2012] Ефименко И.В. Модели использования и интерпретации оценочной информации в прогнозировании: время, состояние, вероятность. //Сборник трудов 13-й национальной конференции по искусственному интеллекту с международным участием. Том. 1, Белгород, 2012, С. 244-251.
- [Хорошевский, 2013] Хорошевский В. Ф. Автоматизация процессов выявления технологических трендов в системе АРМ «Тренд» (OSTIS-2013).

HYBRID APPROACH FOR IDENTIFYING COMPLEX CONCEPTS IN TECHNOLOGICAL FORESIGHT: BLACKBOX MODEL

Efimenko I. V.

*Center for Information Intelligence Applications of
Institute for Statistical Studies and Economics of
Knowledge, NRU HSE
Moscow, Russia*

iefimenko@hse.ru

The paper presents methods and algorithms for identifying complex concepts relevant for the domain of technological foresight within text collections. An approach based on the so called “black box” principle and combination of statistical and linguistic methods is proposed.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8+620

СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ НА ОСНОВЕ ИНТЕГРАЦИИ ОНТОЛОГИЧЕСКОГО, КОГНИТИВНОГО И СОБЫТИЙНОГО МОДЕЛИРОВАНИЯ

Массель Л.В., Массель А.Г.

Институт систем энергетики им. Л.А. Мелентьева СО РАН,

г. Иркутск, Россия

massel@isem.sei.irk.ru

amassel@gmail.com

В статье рассматриваются вопросы построения семантических технологий на основе интеграции онтологического, когнитивного и событийного моделирования. Рассматривается совместное использование семантического и математического моделирования в исследованиях проблем энергетической безопасности (ЭБ). Для поддержки семантических технологий описания знаний и управления знаниями разработана интеллектуальная ИТ-среда, интегрирующая инструментальные средства семантического моделирования: онтологического, когнитивного и событийного.

Ключевые слова: семантические технологии; онтологическое; когнитивное и событийное моделирование.

ВВЕДЕНИЕ

При решении прикладных задач, как правило, приходится сталкиваться с необходимостью интеграции разнородных информационных технологий. Авторам пришлось решать эту проблему при автоматизации исследований проблем энергетической безопасности России и ее регионов.

В исследованиях, выполняемых в Институте систем энергетики им. Л.А. Мелентьева СО РАН, энергетическая безопасность (ЭБ) рассматривается как составляющая национальной безопасности, в частности, как состояние защищенности граждан, общества, государства и экономики от угроз дефицита в обеспечении их обоснованных потребностей топливно-энергетическими ресурсами приемлемого качества в различных условиях [Бушуев и др., 1998].

В исследованиях проблем ЭБ, как правило, решается задача оценки состояния ТЭК России и ее регионов с использованием индикаторов энергетической безопасности (ЭБ) – набора параметров, характеризующих состояние ТЭК и энергетических систем. При этом традиционно решается общая задача линейного программирования, в которой используется технико-экономическая модель ТЭК большой размерности (до нескольких тысяч переменных и нескольких сотен уравнений).

Исследования проблем ЭБ носят многовариантный характер, т.к. рассматриваются и оцениваются возможные варианты направлений развития ТЭК с учетом различных сценариев, описывающих возмущения, вызванные угрозами ЭБ. В последнее время в исследованиях ЭБ принят комбинаторный подход, в результате чего может быть сгенерировано от одного до пяти миллионов вариантов. Для их оценки выбирается соответствующий «коридор» вариантов, что существенно снижает их количество, тем не менее, нагрузка на эксперта очень высока.

Для усовершенствования технологии и снижения нагрузки на эксперта авторами предложено перейти к двухуровневой технологии исследований с использованием методов ситуационного анализа, когда на первом, верхнем уровне используются качественные методы оценки для выбора интересующих эксперта вариантов [Массель Л., 2010], [Массель А., 2010], которые затем детально рассчитываются на втором уровне с использованием традиционных математических моделей. С этой целью разработаны методы построения и совокупность семантических моделей, которые будут рассмотрены ниже, а также инструментальные средства поддержки семантических технологий описания знаний и управления ими.

1. СЕМАНТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Термин «семантическая модель» впервые появился и активно используется в прикладном языкознании. В информатике этот термин начали применять в связи с построением моделей данных [Цаленко, 1989] и впоследствии он стал ассоциироваться с инфологической моделью, или моделью «сущность-связь» (моделью Чена). В области искусственного интеллекта появилась модель представления знаний «семантические сети», внешне похожая на модель «сущность-связь». В 90-х годах Т. Грубер предложил использовать для представления декларативных знаний онтологические модели знаний, или *онтологии* [Gruber, 1993], под которыми понимается множество концептов предметной области и отношений между ними. На основе анализа вышеперечисленных понятий можно сделать вывод, что в обобщенном виде *семантическая модель* – это информационная модель, отражающая понятия предметной области и отношения между ними. В настоящее время семантическое моделирование развивается в рамках одного из трендов направления «Семантические технологии» [Хорошевский, 2012], а именно – «Семантические технологии в приложениях».

В работах коллектива, представляемого авторами¹, к семантическим моделям относятся онтологические, когнитивные [Массель А., 2010] и событийные модели [Аршинский, 2010]. Под *когнитивным моделированием* понимается построение когнитивных моделей, или, иначе, когнитивных карт (ориентированных графов), в которых вершины соответствуют факторам (концептам), а дуги – связям между факторами (положительным или отрицательным), в зависимости от характера причинно-следственного отношения [Трахтенгерц, 1998]. Математическим аппаратом для построения когнитивных моделей является теория графов. В данном контексте когнитивное моделирование используется для ситуационного анализа [Макагонова и др., 2008] проблемы ЭБ и моделирования угроз ЭБ [Массель А., 2010], под которыми понимаются неблагоприятные для энергетики события, сгруппированные в семь видов угроз: техногенные, экономические, природные, социально-политические, внешнеэкономические и внешнеполитические, а также вызванные несовершенством управления (управленческо-правовые). В когнитивных картах, моделирующих угрозы ЭБ, отображаются основные факторы, влияющие на развитие ТЭК страны, выявляются факторы – угрозы и факторы – мероприятия по их предотвращению, а также причинно-следственные связи между этими факторами. Весовые коэффициенты связей назначаются экспертами, по умолчанию весовые коэффициенты составляют +1 и –1.

¹ Лаборатория информационных технологий в энергетике ИСЭМ СО РАН, возглавляемая Л.В. Массель

Под *событийным моделированием* понимается построение поведенческих моделей, причем в качестве объектов моделирования могут рассматриваться как люди, так и технические объекты [Столяров, 2004]. Сущность событийного метода моделирования заключается в отслеживании на модели последовательности событий в том же порядке, в каком они происходили бы в реальной системе. Задаваемые моделью последовательности реализаций событий – цепочки событий – описывают сценарии реакции системы на возникновение инициирующего события, стоящего в начале цепочки.

В качестве инструмента событийного моделирования используется аппарат Joiner-сетей – одной из разновидностей алгебраических сетей, предложенной в [Столяров, 2004]. Joiner-сети можно рассматривать как расширение сетей Петри, ориентированное на построение поведенческих моделей. В основе теории JN лежит описание логики взаимодействия асинхронных процессов в виде набора пусковых и флаговых функций, состоящих из булевых функций. Особенностью JN является то, что они предусматривают как графическое представление, так и описание в виде логических формул, обработку которых можно автоматизировать. Опыт применения Joiner-сетей для моделирования чрезвычайных ситуаций в энергетике рассмотрен, в частности, в [Аршинский, 2010], [Массель Л. и др., 2010].

2. ИНТЕГРАЦИЯ СЕМАНТИЧЕСКИХ ТЕХНОЛОГИЙ

Под семантической технологией в данном контексте понимается совокупность методов и инструментальных средств для выполнения действий, связанных с представлением знаний и управлением ими. Для представления знаний используются онтологические, когнитивные и событийные модели, а также фреймовые модели представления знаний.

Для поддержки двухуровневой технологии исследований, в рамках которой интегрируется математическое и семантическое моделирование, разработана многокомпонентная интеллектуальная ИТ-среда.

Интеллектуальная ИТ-среда определяется как $V_{IT} = \{O, E, M_C, M_S\} \cup T_V$, где $\{O\}$ – множество онтологий, $\{E\}$ – множество описаний прецедентов чрезвычайных ситуаций, $\{M_C\}$ – множество когнитивных моделей, $\{M_S\}$ – множество событийных моделей, T_V – инструментальные средства поддержки ИТ-среды, включающие описание знаний, представленных в виде онтологий, описаний прецедентов ЧС, когнитивных и событийных моделей и средства оперирования ими.

Таким образом, интеллектуальная ИТ-среда включает пространство знаний, интегрирующее: онтологические модели знаний в области исследований ЭБ, базу знаний о прецедентах ЧС в энергетике и базы знаний, содержащие когнитивные модели стратегических угроз ЭБ и событийные модели развития и последствий ЧС в энергетике (рисунок 1), а также инструментальные средства описания знаний и оперирования ими (рисунок 2).

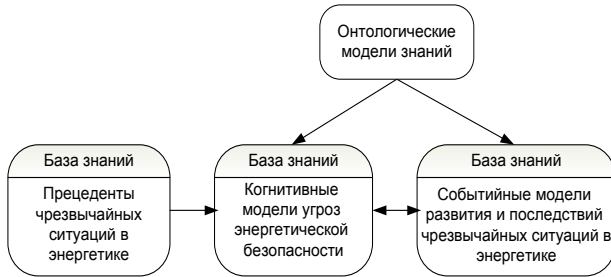


Рисунок 1 - Пространство знаний, поддерживаемое интеллектуальной ИТ-средой

Инструментальные средства включают библиотеки когнитивного и событийного моделирования (CogMap [Массель А., 2010] и EventMap [Аршинский, 2010]), экспертную систему «Emergency» для описания и анализа прецедентов чрезвычайных ситуаций в энергетике [Массель Л. и др., 2010] и многоагентный программный комплекс ИНТЭК-М [Фартышев, 2010]. В последнее время в состав интеллектуальной среды включена библиотека построения байесовских сетей доверия Bayesian Nets для оценки рисков угроз ЭБ [Массель

Л., 2012]. Все перечисленные инструментальные средства можно рассматривать как компоненты интеллектуальной ИТ-среды, интегрируемые с помощью специального Репозитория, разработанного в коллективе, представляемом авторами.

Для интеграции семантических и математических моделей и перехода от качественной оценки к традиционным расчетам используются дедуктивный синтез сценариев вычислительных экспериментов и автоматизированное построение экономико-математических моделей ТЭК [Массель Л., 2011]. Согласно этому подходу, исходные данные для моделирования представляются в виде XML-файлов. Исходные данные включают описания: объекта моделирования в виде XML-прототипа, структуры и форматов файлов математических моделей и средств содержательной интерпретации. Процессы преобразования данных в ходе вычислительного эксперимента представляются в виде декларативных описаний. Под ними понимаются формулы узкого исчисления предикатов, которые строятся по исходным XML-файлам. Обработка декларативных описаний позволяет синтезировать сценарий формирования файлов заданного формата для качественного анализа или численного расчета. Основанные на этом подходе средства дедуктивного синтеза программ обеспечивают связь между инструментами CogMap и EventMap и ПК ИНТЭК-М в рамках интеллектуальной ИТ-среды.

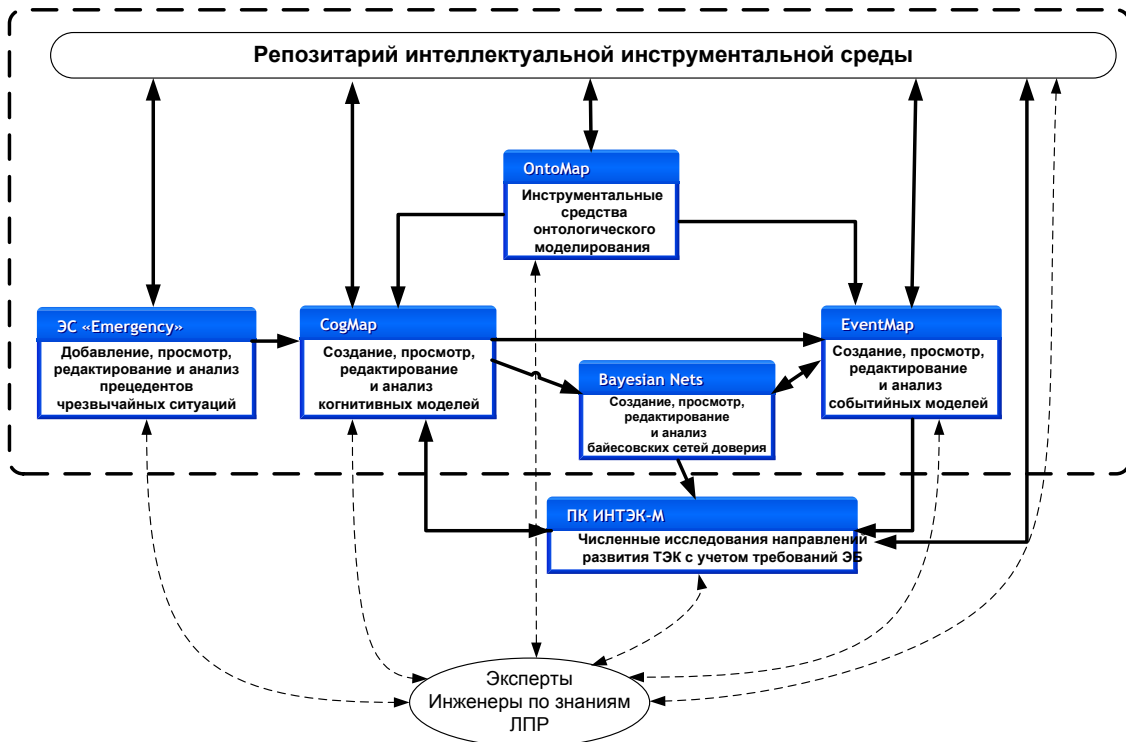


Рисунок 2 - Взаимосвязь инструментальных средств интеллектуальной ИТ-среды

ЗАКЛЮЧЕНИЕ

В статье рассмотрено построение семантических технологий на основе интеграции семантического и математического моделирования. Описана двухуровневая технология исследований проблем энергетической безопасности, базирующаяся на использовании семантических технологий, и многокомпонентная интеллектуальная ИТ-среда для их поддержки.

Результаты, представленные в статье, получены при частичной финансовой поддержке грантов РФФИ № 11-07-00192, № 12-07-00359, гранта Программы Президиума РАН №229 и гранта на выполнение интеграционного проекта СО РАН и НАН Беларуси №18 (2012-2014).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Бушуев и др., 1998] Энергетическая безопасность России / В.В. Бушуев, Н.И. Воропай, А.М. Мастепанов, Ю.К. Шафраник и др. – Новосибирск: Наука, 1998. – 302 с.

[Массель Л., 2010] Массель Л.В. Применение онтологического, когнитивного и событийного моделирования для анализа развития и последствий чрезвычайных ситуаций в энергетике / Проблемы безопасности и чрезвычайных ситуаций. - №2. - 2010. - С. 34-43.

[Массель А., 2010] Массель А.Г. Методологический подход к организации интеллектуальной поддержки исследований проблемы энергетической безопасности / «Информационные технологии». – №9. – 2010. – С. 32-36.

[Цаленко, 1989] Цаленко М.Ш. Моделирование семантики в базах данных. — М.: Наука. Гл. ред. физ.-мат.лит., 1989. - 288 с.

[Gruber, 1993] Gruber T.R. A Translational Approach to Portable Ontologies // Knowledge Acquisition. – 1993. – V. 5. – No. 2. – P. 199 – 220.

[Хорошевский, 2012] Хорошевский В.Ф. Семантические технологии: ожидания и тренды/ Труды II Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». – Беларусь, Минск: БГУИР, 2012. - С. 143-158.

[Трахтенгерц, 1998] Трахтенгерц Э.А. Компьютерная поддержка принятия решений. - М.: СИНТЕГ, 1998. - 376 с.

[Столяров, 2004] Столяров Л.Н., Новик К.В. Реализация параллельных процессов с помощью сетей Joiner-net // Информационные и математические технологии / Труды Байкальской Всероссийской конференции. – Иркутск: ИСЭМ СО РАН, 2004. – С. 11-14.

[Макагонова и др., 2008] Макагонова Н.Н., Массель Л.В., Бахвалов С.В. Применение когнитивного моделирования для ситуационного анализа проблемы энергетической безопасности // Вычислительные технологии, т. 13, ч. II, 2008. – С. 341-347.

[Массель А., 2010] Массель А.Г. Когнитивное моделирование угроз энергетической безопасности / Горный информационно-аналитический бюллетень (научно-технический журнал), отдельный выпуск №17. – М.: Изд-во «Горная книга», 2010. – С. 194 – 199.

[Аршинский, 2010] Аршинский В.Л. Событийное моделирование чрезвычайных ситуаций в энергетике / Труды Международной конференции «Информационные технологии в науке, образовании, телекоммуникации и бизнесе». Приложение к журналу «Открытое образование». – Украина, Гурзуф, 2010. – С. 299-301.

[Массель Л. и др., 2010] Массель Л.В., Аршинский В.Л., Массель А.Г. Интеллектуальные информационные технологии поддержки принятия решений в исследованиях и обеспечении энергетической безопасности/ Труды Международной конференции «Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта», т. 1. - Украина, Евпатория: ХНТУ, 2010. – С. 192-196.

[Фартышев, 2010] Фартышев Д.А. Разработка многоагентного ПК ИНТЭК-М для исследований проблемы

энергетической безопасности / Программные продукты и системы. - № 3. – 2010. – С. 126-129.

[Массель Л., 2012] Массель Л.В., Пяткова Е.В. Применение байесовских сетей доверия для интеллектуальной поддержки исследований проблем энергетической безопасности. – Вестник ИргТУ. - №2. - 2012. - С. 8-13

[Массель Л., 2011] Массель Л.В., Курганская О.В. Автоматизация вычислительного эксперимента на основе логических моделей – Вестник Иркутского государственного технического университета, №2 (49). – Иркутск: Издательство Иркутского государственного университета, 2011. – с. 8-14.

SEMANTIC TECHNOLOGIES BASED ON THE INTEGRATION OF THE ONTOLOGY, COGNITIVE AND EVENT MODELING

Massel L.V., Massel A.G.

* *Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences
Irkutsk, Russia*

massel@isem.sei.irk.ru

amassel@gmail.com

The article describes the construction of semantic technologies based on the integration of the ontological, cognitive and event modeling. The authors suggest sharing of semantic and mathematical modeling in the research of energy security problems. The Intelligent IT-environment is developed to support the description of knowledge space and knowledge management and to integrate tools for the ontological, cognitive and event modeling.

INTRODUCTION

The authors justify the need of heterogeneous information technologies integration for solving of applied problems. Specificity of studies problems of energy security is considered and two-level technology of research is proposed. It's proposed to use semantic technologies on the first level for the rapid analysis of the alternatives.

MAIN PART

The development of the semantic modeling is analyzed and a generalized notion of semantic model is introduced. Author considers these types of semantic modeling as an ontological, cognitive and event modeling. An approach to the integration of semantic and mathematical models is proposed and the intellectual IT-environment for support and integration of semantic and mathematical modeling is described.

CONCLUSION

The results presented in this paper were obtained with the partial financial support by grants RFBR № 11-07-00192, № 12-07-00359, by grant of RAS Presidium Program № 229 and by grant for the implementation of the integration project of SB RAS and the National Academy of Sciences of Belarus № 18 (2012-2014).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004:007:51-7

МОДЕЛИ СПОНТАННОЙ КОГНИТИВНОЙ ДИНАМИКИ

Прокопчук Ю.А.

*Институт технической механики НАНУ и НККАУ,
Украинский государственный химико-технологический университет
г. Днепропетровск, Украина*

Itk3@ukr.net

Исследование проводилось с целью построения формальных структур знаний для моделирования процессов эмпирического образного мышления в компьютерных технологиях. Предложена конструктивная версия механизма предельных обобщений и полимодельной дополнителности как парадигмального базиса и показана ее продуктивность при решении теоретико-прикладных проблем когнитивной науки. Построены формальные модели спонтанных идеальных процессов.

Ключевые слова: когнитивные архитектуры и процессы; парадигма предельных обобщений; модели идеальных процессов; модели знаний.

ВВЕДЕНИЕ

Исследование проводится с целью построения формальных структур знаний для моделирования процессов образного мышления в компьютерных технологиях [Валькман, 2009], [Голенков и др., 2001]. В сфере когнитивных наук продвижение в решении данной проблемы является актуальным для разработки партнерских систем, средств анализа и управления сложными системами, интерфейсов «мозг-компьютер» и других технологий, для которых критически важно максимальное соответствие закономерностям протекания когнитивных и эмоциональных процессов человека [Дубровский, 2007], [Chalmers, 1996].

В работе [Прокопчук, 2012] предложена конструктивная версия механизма предельных обобщений и полимодельной дополнителности как парадигмального базиса и показана ее продуктивность в исследованиях, связанных с анализом различных познавательных ситуаций и решением теоретико-прикладных проблем когнитивной науки. В настоящей работе строятся формальные модели спонтанных идеальных процессов.

1. Базовые сущности

1.1. Орграфы набросков, банк тестов

Предполагается, что произвольную ситуацию действительности (прецедент, систему, процесс) можно описать с помощью множества

элементарных тестов, представимых в виде «тест = значение». Мощность множества всех элементарных тестов совпадает с мощностью всех односложных вопросов вида «вопрос? ответ». Конкретный результат теста τ будем обозначать через $\underline{\tau}$. Результаты тестов могут выбираться (формироваться) из разных доменов (множеств значений). Для фиксации того, что в качестве множества результатов теста τ используется домен T , будем использовать нотацию: τ/T .

Правила пересчета значений теста из одного домена в другой задают *взвешенный орграф доменов теста* τ :

$$G(\tau) = \{T \rightarrow_e T'\}_\tau, \quad (1)$$

где e – структурная энергия (проводимость связи), которая может динамически изменяться. Домен слева от стрелки в фундаментальной триаде $(T \rightarrow_e T')$ будем называть доменом-предком (родителем), а справа от стрелки доменом-потомком. Совокупная смысловая область элементов домена–потомка полностью совпадает с совокупной смысловой областью элементов домена–предка. Без ограничений общности положим, что домены орграфа $G(\tau)$ состоят из альтернативных элементов (точечных или атомарных элементов) и атомарный элемент любого домена однозначным образом преобразуется в атомарные элементы доменов-потомков (если они существуют).

С помощью структурной энергии можно

моделировать такие эффекты памяти как распространение волновой активности, «забывание», «вспоминание» и т.д. Детализация структурной энергии на уровне домена позволяет рассмотреть динамические особенности обобщения каждого элемента каждого домена, а именно: $\forall a \in T(T.a, e_a \rightarrow T'.b)$.

Орграф имеет одну базовую вершину T_0 – базовый домен со значениями (элементами) максимально высокого уровня точности. В базовую вершину не входит ни одна дуга орграфа. Любой домен орграфа проецируется на весь базовый домен. На основе орграфов доменов могут быть построены структурно-завершенные орграфы доменов $G^+(\tau)$ путем автоматического порождения вершин-листьев вида $\{a; \neg a\}$ для каждого элемента дискретного домена [Прокопчук, 2012].

Конфигуратором теста называется процедурная реализация орграфа доменов. Ниже приведен пример конфигулятора теста «Возраст» (без структурной энергии):

Возраст {B3 {Молодой ^1; Немолодой ^2 3} B2 {Молодой ^1 [1; 33]; Средних лет ^2 (33; 60); Пожилой ^3 (60; 100)} B1 {[1; 100]}}.

$G(\text{Возраст}) = \{B1 \rightarrow B2 \rightarrow B3\}$.

Автоматизмами среды всегда порождается смысловая траектория обобщения любого значения любого домена, в частности базового. Пример: Возраст/B1? 77 \rightarrow Возраст/B2? Пожилой \rightarrow Возраст/B3? Немолодой.

Пример показывает, каким образом в орграфах доменов порождается смысл при движении информации от сенсориума (измерительной системы) к высшим отделам обработки и анализа информации. Следует отметить, что наличие знаков (термов) для элементов доменов не является обязательным. Пример:

Возраст {B3 {^1; ^2 3} B2 {^1 [1; 33]; ^2 (33; 60); ^3 (60; 100)} B1 {[1; 100]}}.

Все дальнейшие построения, включая модели знаний, работают и в этом случае, что позволяет рассматривать когнитивные процессы у животных. Более того, лишь малая часть тестов и доменов орграфов доменов тестов имеют знаковую (вербальную) интерпретацию в силу полимодальности любого образа. Это объясняет, почему традиционные лингво-логические интеллектуальные системы не обеспечивают однозначного сопоставления между когнитивным представлением и описанием в вербальной форме.

Орграф доменов теста является частным случаем базовой сущности «взвешенный орграф набросков»:

$$G_s(W) = \{P \rightarrow_e P'\}_W, \quad (2)$$

где W – произвольное явление действительности, образ; P, P' – наброски; e – структурная энергия (проводимость связи). Причем

$||I(P')| < |I(P)|$, где I – оператор вычисления информации (происходит диссипация информации). Любой набросок P при декогерентном рассмотрении является множеством значений тестов $\{\tau/T\}_P$. Для одного и того же образа W могут быть построены орграфы набросков разных типов. Каждому типу соответствует свой оператор I . При рекогерентном рассмотрении существует квантово-семантическое представление орграфов набросков [Прокопчук, 2012].

Орграфы набросков *структурно неустойчивы*: с ними связана «структурная энергия» и эволюция, выражающаяся в постоянном спонтанном изменении структуры при постоянстве внешних условий. В этой связи важно отметить, что система с переменной структурой элементов может обладать *самодвижением* и эволюцией. Слои набросков размещаются в разных отделах памяти (быстрые, медленные, сверхмедленные отделы), что обусловлено стремлением минимизировать структурную энергию.

Текст, речь, музыка – это последовательность сигналов-знаков $\langle \underline{a}/A \rangle$, которые призваны активировать у респондента определенную волновую активность (переживания). Для них, как и для любых сигналов, строятся орграфы набросков, модели знаний и т.д.

В совокупности орграфы доменов образуют *Банк тестов* $\{G(\tau)\}$. Различные комбинации доменов для всех тестов определяют различные уровни обобщенности описания ситуаций действительности. Банк тестов обеспечивает переход от физической реальности к феноменологическому пространству (ФП) и служит *системой координат* произвольного ментального многообразия ФП. Важно отметить, что разноуровневые описания образа, явления, ситуации существуют *одновременно*. Другими словами, имеет место неразделяемая *суперпозиция описаний* разного уровня обобщенности. Подобная суперпозиция препятствует вербализации (декогеренции). Общее количество *описаний* или *информационных экранов* определяется выражением:

$$M = \prod_{\tau \in \{\tau\}} |G(\tau)|, \quad (3)$$

где $|G(\tau)|$ – число доменов в орграфе $G(\tau)$.

ФП является объединением ментальных многообразий (ММ). Каждое ММ имеет в общем случае свою систему координат. Одно и то же явление или образ может быть представлено в разных ММ. В общем случае, в качестве системы координат ММ выступает произвольный *Банк образов* $\{G_s(W)\}$. Поскольку любой набросок P на нижнем уровне описывается с помощью $\{G(\tau)\}$, то $\{G_s(W)\}$ является системой координат более высокого уровня общности (порождение образов из

образов). В этом смысле можно говорить об иерархии систем координат ММ по степени обобщенности.

Будем говорить, что домен T' доминирует домен T в рамках орграфа $G(\tau)$, если они не совпадают и существует путь $T \rightarrow \dots \rightarrow T'$. Доминирование обозначим нотацией: $T < T'$. Нестрогое доминирование (домены могут совпадать) обозначим нотацией: $T \leq T'$. Если один домен доминирует другой, то он имеет более высокий уровень обобщения. Ясно, что все терминальные вершины орграфа $G(\tau)$ являются недоминируемыми.

1.2. Системопаттерны

Любые преобразования, движения, вывод, импульсы, вычислительные модели в системе координат $\{G(\tau)\}$ можно описать с помощью динамических системопаттернов (или просто системопаттернов) вида [Прокопчук, 2012]:

$$f/\mu: \{a/A\}, e/E \rightarrow \{b/B\}, \mu \in \{\mu\}_f, \quad (4)$$

где $\{a/A\}$ – входные тесты; $\{b/B\}$ – выходные тесты; e/E – требуемая структурная энергия, ресурсы; μ – механизм реализации. С помощью структурной энергии, в частности, может передаваться возбуждение (активность) в вычислительной среде. Преобразования между доменами ($T \rightarrow_e T'$) и набросками ($P \rightarrow_e P'$) являются разновидностями системопаттернов.

Для любого системопаттерна строится орграф набросков, что позволяет системно реализовать метамоделирование (любой набросок системопаттерна является системопаттерном). Важным свойством системопаттернов является их активность: если системопаттерн запущен, то он самостоятельно может искать значения входных тестов $\{a/A\}$. Реальная скорость выполнения системопаттерна зависит от наличной структурной энергии (ресурсов). На основе системопаттернов можно строить среды радикалов (СР), интеллектуальные среды, банки математических моделей, функциональные системы (ФС) [Прокопчук, 2012].

2. Производные сущности

2.1. Банк прецедентов, идеальные закономерности

Для решения той или иной когнитивной задачи (Z -задачи) формируется множество прецедентов с известными исходами $\Omega = \{\alpha(\{\underline{\tau}/T\}, \underline{z}/Z)\}$, где $Z = \{1, \dots, N\}$ – множество заключений (диагнозов, прогнозов, управлений); $\{\underline{\tau}/T\}$ – множество значений тестов. Без потери общности примем, что каждый тест входит в описание прецедента (ситуации действительности) один раз. Кроме того,

будем рассматривать описания прецедентов с полной информацией (имеются значения всех тестов из $\{G(\tau)\}$). Через $\Omega(\{\tau/T_0\})$ обозначим априорные описания прецедентов.

Зафиксируем уровень общности $\{\tau/T\}$. Описание базы прецедентов $\Omega(\{\tau/T\}, Z)$ назовем конфликтным, если существуют хотя бы два прецедента $\alpha(\{\underline{\tau}/T\}_\alpha, z_\alpha)$ и $\beta(\{\underline{\tau}/T\}_\beta, z_\beta)$ такие, что $\{\underline{\tau}/T\}_\alpha = \{\underline{\tau}/T\}_\beta$, но $z_\alpha \neq z_\beta$. Будем предполагать, что априорное описание $\Omega(\{\tau/T_0\})$ бесконфликтно.

Будем говорить, что описание $\{\tau/T'\}$ доминирует описание $\{\tau/T\}$, если $\forall \tau T \leq T'$ и $\exists \tau: T < T'$. Доминирование описаний будем обозначать нотацией: $\{\tau/T\} < \{\tau/T'\}$.

Предложение 1. Если описание базы прецедентов $\Omega(\{\tau/T\}, Z)$ бесконфликтно, то бесконфликтны также все описания $\Omega(\{\tau/T'\}, Z)$ такие, что $\{\tau/T'\} < \{\tau/T\}$. Если описание базы прецедентов $\Omega(\{\tau/T\}, Z)$ конфликтно, то конфликтны также все описания $\Omega(\{\tau/T'\}, Z)$ такие, что $\{\tau/T\} < \{\tau/T'\}$.

Описание базы прецедентов назовем критическим – $\{\tau/T\}^*$, если оно бесконфликтно, но любое доминирующее описание конфликтно. Остальные бесконфликтные описания назовем докритическими. Все конфликтные описания назовем надкритическими [Прокопчук, 2012].

Множество всех описаний базы прецедентов образует орграф набросков $\Omega(Z)$, в котором $\Omega(\{\tau/T\}, Z)$ – отдельный набросок, а множество критических описаний образует экстремальный пограничный слой набросков. Можно также рассмотреть орграф набросков каждого прецедента α , тогда описание $\alpha(\{\underline{\tau}/T\}, \underline{z}/Z)$ – это отдельный набросок. Для каждого прецедента можно найти критические наброски максимального уровня общности, которые позволяют отличить данный прецедент от всех остальных прецедентов из Ω . Надкритические наброски служат основой для проведения аналогий и переноса.

Контекстом Z -задачи назовем кортеж $K = \langle \Omega(Z), \{G(\tau)\} \rangle$. Заданием K любой прецедент погружается в контекст K . Пусть $\{\tau\}$ – полный набор тестов в рамках Банка тестов $\{G(\tau)\}$.

Идеальной закономерностью V в рамках контекста K назовем произвольную совокупность значений тестов, позволяющую однозначно установить заключение:

$$V = (\{\underline{a}/A\} \rightarrow \underline{z}/Z), \quad (5)$$

$$\exists \alpha(\{\underline{\tau}/T\}_\alpha, \underline{z}/Z) \in \Omega(Z) : \{\underline{a}/A\} \subseteq \{\underline{\tau}/T\}_\alpha.$$

Компактная запись: $V(\{\underline{a}/A\}, \underline{z}/Z)$. «Идеальность» означает субъективный характер закономерности. Закономерность является разновидностью системопаттерна. Закономерность может быть избыточна, что подтверждает следующее предложение.

Предложение 2. Если $V(\{\underline{a}/A\}, \underline{z}/Z)$ закономерность в рамках контекста K , то любое множество значений тестов $\{\underline{a}/A\} \cup \{\underline{b}/B\}$ такое, что $\{a\} \cap \{b\} = \emptyset$ образует закономерность $V'(\{\underline{a}/A\} \cup \{\underline{b}/B\}, \underline{z}/Z)$ в рамках контекста K при выполнении условия:

$$\begin{aligned} \exists \alpha(\{\underline{\tau}/T\}_\alpha, \underline{z}/Z) \in \Omega(Z) : \\ \{\underline{a}/A\} \cup \{\underline{b}/B\} \subseteq \{\underline{\tau}/T\}_\alpha. \end{aligned} \quad (6)$$

Предложение 3. Закономерность $V(\{\underline{a}/A\}, \underline{z}/Z)$ применима ко всем описаниям прецедентов $\{\alpha(\{\underline{a}/A'\} \cup \{\underline{b}/B\}, \underline{z}/Z)\}$ таким, что: 1) $\{a\} \cap \{b\} = \emptyset$, $\{a\} \cup \{b\} = \{\tau\}$; 2) $\{a/A'\} \leq \{a/A\}$.

Действительно, любой набросок $\alpha(\{\underline{a}/A'\} \cup \{\underline{b}/B\}, \underline{z}/Z)$ переходит в набросок $\alpha(\{\underline{a}/A\} \cup \{\underline{b}/B\}, \underline{z}/Z)$, для которого применяется закономерность $V(\{\underline{a}/A\}, \underline{z}/Z)$.

Формальным синдромом S (или просто *синдромом*) в рамках контекста K назовем неизбыточную идеальную закономерность. Другими словами, ни один тест из описания $S(\{\underline{a}/A\}, \underline{z}/Z)$ убрать нельзя без потери однозначности заключения.

Будем говорить, что закономерность $V'(\{\underline{a}/A'\}, \underline{z}/Z)$ доминирует закономерность $V(\{\underline{a}/A\}, \underline{z}/Z)$, если $\{a'\} \subseteq \{a\}$, $\forall a \in \{a'\}$: $A \leq A'$ и выполняется хотя бы одно из условий: а) $|\{a'\}| < |\{a\}|$; б) $\exists a \in \{a'\}$: $A < A'$.

Предложение 4. Для любого синдрома S доминирующей закономерностью может быть только синдром с тем же составом тестов.

Предельным синдромом S^* в рамках контекста K назовем синдром, у которого отсутствуют доминирующие закономерности.

Множество всех закономерностей в рамках контекста K обозначим через $\{V\}_{Full}$, множество всех синдромов обозначим через $\{S\}_{Full}$, а множество всех предельных синдромов обозначим $\{S^*\}_{Full}$. Ясно, что $\{S^*\}_{Full} \subseteq \{S\}_{Full} \subseteq \{V\}_{Full}$.

2.2. Модели знаний

Моделью знаний назовем произвольное множество закономерностей $\{V\}$, которое позволяет установить заключение как минимум для каждого прецедента из $\Omega(\{\tau/T_0\})$.

Будем говорить, что модель знаний $\{V'\}$ доминирует модель знаний $\{V\}$, если она применима к большему числу набросков прецедентов из $\Omega(Z)$, включая и те наброски, к которым применима $\{V\}$. Факт доминирования будем отражать нотацией $\{V'\} > \{V\}$. Будем говорить, что модели знаний $\{V'\}$ и $\{V\}$ эквивалентны в плане доминирования, если они применимы к одному и тому же множеству набросков прецедентов. Факт эквивалентности в плане доминирования будем отражать нотацией $\{V'\} \sim \{V\}$. Если для модели знаний $\{V\}$ не существует доминирующая модель, то будем говорить, что она *недоминируема*. Если между двумя моделями знаний $\{V'\}$ и $\{V\}$ не выполняется отношение доминирования и они не эквивалентны в плане доминирования, то будем говорить, что такие модели знаний *несравнимы между собой в плане доминирования*. Данный факт будем отражать нотацией $\{V'\} >< \{V\}$.

Предложение 5. В рамках любого критического $\{\tau/T\}^*$ или докритического описания базы прецедентов можно построить модели знаний, но в рамках любого надкритического описания $\{\tau/T\} > \{\tau/T\}^*$ такой модели построить нельзя.

Предложение 6. Модели знаний $\{V\}_{Full}$, $\{S\}_{Full}$ и $\{S^*\}_{Full}$ недоминируемы и эквивалентны в плане доминирования.

Большой практический интерес представляет поиск всех минимальных (неизбыточных) по составу моделей знаний, которые принадлежат $\{S^*\}_{Full}$ и эквивалентны по доминированию $\{S^*\}_{Full}$. Такие модели знаний обозначим $\{S^*\}_{Full-Min}$. Их поиск может быть выполнен методом исключения части синдромов $\{S^*\}_{Full}$.

«Материализация» минимальных моделей в виде СР приводит к ФС когнитивно-поведенческого уровня - « $\{S^*\}_{Full-Min}$ - СР» и описывает процесс формирования имплицитного опыта: обучение начинается на когнитивном уровне, а результат переходит на уровень рефлекторных реакций. Развиваемый формализм помогает осознать имплицитный опыт и перевести его в эксплицитный опыт.

3. Динамика формирования моделей знаний

Пусть $V(\{a/A\}, \underline{z}/Z)$ и $V'(\{b/B\}, \underline{z}/Z)$ – две закономерности. Элементарной операцией обобщения Φ^1 назовем переход $V \rightarrow V'$, в котором обобщается значение лишь одного из тестов закономерности V по схеме $A \rightarrow A'$ (родитель – потомок). Верхний индекс означает тип операции (1 – операция обобщения). Состав тестов закономерности V при операции Φ^1 не изменяется. Ясно, что при данной операции происходит диссипация информации. Общее количество различных операций типа Φ^1 определяется выражением:

$$|\{\Phi^1\}| = \sum_{\tau \in \{\tau\}} |\{T \rightarrow_e T'\}_{\tau}|. \quad (7)$$

Элементарной операцией редукции Φ^2 назовем переход $V \rightarrow V'$, при котором исключается один из тестов закономерности V . При данной операции также происходит диссипация информации. Общее количество различных операций типа Φ^2 совпадает с $|\{\tau\}|$. Операции $\{\Phi^v\}$ представляют собой элементарные акты познания.

Пусть фиксирован контекст K и множество операций $\{\Phi^v\}$. Тогда нелинейную дискретную хаотическую информационно-диссипативную систему будем описывать следующим рекуррентным соотношением:

$$\begin{aligned} \{V\}_{n+1} &= \Phi_{\sigma_n}(\{V\}_n, \Omega(Z), \{G(\tau)\}), \\ n=0, 1, \dots, \quad \Phi_{\sigma_n} &\in \{\Phi^v\}, \end{aligned} \quad (8)$$

$$|\{\Phi^v\}| = \sum_{\tau \in \{\tau\}} |\{T \rightarrow_e T'\}_{\tau}| + |\{\tau\}|,$$

$$\{V\}_0 = \{\{\underline{\tau}/T\}_{\alpha} \rightarrow z_{\alpha} \mid \alpha \in \Omega(\{\tau/T_0\})\},$$

где Φ_{σ_n} выбирается с вероятностью p_{σ_n} (сумма всех вероятностей равна единице). На каждом шаге процесса случайно выбранная операция Φ^v из множества операций $\{\Phi^v\}$ применяется к случайно выбранной закономерности из $\{V\}_n$. Если к выбранной закономерности нельзя применить Φ^v , то случайным образом выбирается другая закономерность из оставшихся и т.д. Если выбранную операцию Φ^v нельзя применить ни к одной закономерности из $\{V\}_n$, то случайным образом выбирается другая операция из оставшихся. Процесс (8) останавливается только тогда, когда к $\{V\}_n$ нельзя применить ни одну операцию из $\{\Phi^v\}$. Аттрактором процесса является неподвижная точка – некоторое множество закономерностей $\{V\}_{fin}$.

Во многом процесс (8) напоминает систему

случайных итеративных функций (ССИФ) – сжимающих отображений, порождающих фракталы [Макаренков, 2002]. Этот процесс чаще всего происходит в диссипативных системах, траектории которых заполняют низкоразмерное инвариантное притягивающее подмножество – аттрактор в фазовом пространстве. Если W – оператор Хатчинсона СИФ, то неподвижная точка A такая, что $W(A) = A$, называется аттрактором СИФ или фракталом [Макаренков, 2002]. В нашем случае, аналогом оператора Хатчинсона является $\{\Phi^v\}$ (сохраним обозначение W):

$$W(\{V\}) = \{W(V) \mid V \in \{V\}\}. \quad (9)$$

Пусть $H(\{V\}_{Full})$ – множество непустых подмножеств $\{V\}_{Full}$, тогда в отличие от (8) когнитивный аналог оператора Хатчинсона W описывает детерминированную дискретную динамическую систему с пространством состояний $H(\{V\}_{Full})$ и преобразованием W . Если W^{on} – композиция порядка n оператора W , то последовательность множеств, полученную в результате итерирования $\{V\}$, т.е. $\{\{V\}, W(\{V\}), W^2(\{V\}), \dots, W^{on}(\{V\}), \dots\}$, назовем *смысловой орбитой* $\{V\}$. Совокупность всех слоев орбиты образует полный *орграф набросков* $\{V\}$.

Траекторию $\{V\}_0 \rightarrow \{V\}_1 \rightarrow \dots \rightarrow \{V\}_{fin}$ назовем *хаотической траекторией* множества закономерностей $\{V\}_0$. «Созревание» орграфов набросков происходит путем наращивания хаотических траекторий. Отметим, что множество $\{V\}_0$ является моделью знаний по определению.

Предложение 7. Любое множество закономерностей $\{V\}_n$ ($n=0, 1, \dots, fin$), порождаемое процессом (8), является моделью знаний. Справедливо также: $\{V\}_0 \leq \{V\}_1 \leq \dots \leq \{V\}_{fin}$.

Важно отметить, что, несмотря на диссипацию информации, доминирование не уменьшается, а наоборот, как правило, увеличивается.

Предложение 8. Аттрактор динамического процесса (8) принадлежит множеству предельных синдромов $\{S^*\}_{Full}$. Объединение аттракторов всех процессов (8) в точности совпадает с $\{S^*\}_{Full}$. Другими словами

$$\lim_{n \rightarrow \infty} W^{on}(\{V\}_0) = \{S^*\}_{Full}. \quad (10)$$

Неподвижная точка $A = \{S^*\}_{Full}$ является аттрактором множества операций $\{\Phi^v\}$.

Модели (8), (10) описывают когнитивную метаэволюцию или *информогенез* предельных моделей знаний при фиксированном контексте K .

Смысловая орбита образует оргграф набросков первичной сенсорной информации $\Omega(\{\tau/T_0\})$, т.е. модели знаний $\{V\}_0$. Важно отметить, что оргграф набросков может формироваться длительное время на основе хаотических смысловых траекторий, обуславливая структурную неустойчивость применяемых моделей знаний и спонтанное самодвижение. Ситуация усложняется, если изменяется контекст K .

Введем в рассмотрение третью элементарную операцию - Φ^3 , суть которой состоит в исключении какой-либо закономерности V из модели знаний $\{V\}$ при условии, что $\{V\}' = \{V\} \setminus V$ эквивалентна в плане доминирования $\{V\}$. Ясно, что операция Φ^3 также приводит к диссипации информации. Количество возможных операций - $|\{V\}|$.

Рассмотрим вторую (эволюционную) фазу нелинейного хаотического процесса когнитивной самоорганизации, а именно:

$$\begin{aligned} \{V\}_{n+1} &= \Phi_{\sigma_n}(\{V\}_n, \Omega(Z), \{G(\tau)\}), \\ n=0,1,\dots, \quad \Phi_{\sigma_n} &\in \{\Phi^3\}_n, \quad |\{\Phi^3\}_n| = |\{V\}_n|, \quad (11) \\ \{V\}_0 &= \{S^*\}_{Full}, \end{aligned}$$

где Φ_{σ_n} выбирается из $\{\Phi^3\}_n$ с вероятностью p_{σ_n} (сумма всех вероятностей равна единице). Принцип построения процесса (11) совпадает с принципом построения процесса (8).

Предложение 9. Аттрактором динамического процесса (11) является неизбыточная предельная модель знаний $\{S^*\}_{Full-Min}$. Совокупность аттракторов всех процессов (11) совпадает с полным конечным набором всех неизбыточных предельных моделей знаний $\{\{S^*\}_{Full-Min}\}$.

Модели динамики вида (8) – (11) могут быть построены для обработки любого образа [Прокопчук, 2012]. Хаотические смысловые траектории заканчиваются финитными набросками, а орбита образует оргграф набросков.

Модели (8) - (11) должны быть дополнены динамикой контекста K (изменяется Банк тестов, включая структурную энергию, и/или база прецедентов). Так динамику изменения базы прецедентов можно представить соотношением:

$$\Omega(t+1) = \Omega(t) \cup \{\alpha\}_t, \quad (12)$$

где $\{\alpha\}_t$ – новые ситуации, t - время. После каждого значимого изменения контекста K эволюция (8) – (11) запускается заново (аналог эволюции после когнитивной катастрофы).

ЗАКЛЮЧЕНИЕ

При построении моделей когнитивных архитектур и процессов выдвинуто предположение о существовании единого универсального принципа структурирования информационно-энергетических потоков (этот принцип закладывался в “устройство” мышления с эволюционным возникновением самого мышления). Следствием принципа являются декогерентные и рекогерентные модели трех взаимосвязанных базовых информационно-энергетических сущностей: оргграфа набросков (целое представляется суперпозицией набросков), динамического системопаттерна и структурной энергии.

Элементы-сущности находятся в постоянном движении из-за своей структурной неустойчивости (структурного разнообразия), что создает внутренние движущие силы и обеспечивает самодвижение. Трехсущностные взаимодействия порождают виртуальную сплошную среду, которая может служить моделью субъективной реальности.

С формализацией описания спонтанных идеальных процессов фактически вводится новое естественнонаучное представление о феномене идеального, лежащего в основе организации любой деятельности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Валькман, 2009] Валькман Ю. Р. Модельно-параметрическое пространство – средство представления знаний исследователей сложных систем / Ю. Р. Валькман, А. Ю. Рыхальский // УСИМ. – 2009. – №1. – С. 20 – 30.
- [Голенков В.В. и др., 2001] Голенков В.В. Представление и обработка знаний в графодинамических ассоциативных машинах / В.В.Голенков, О.Е.Елисева, В.П.Ивашенко и др.: Под ред. В.В.Голенкова. – Мн.: БГУИР, 2001. – 412 с.
- [Дубровский, 2007] Дубровский Д.И. Сознание, мозг, искусственный интеллект. – М.: Стратегия-Центр, 2007. - 272 с.
- [Макаренков, 2002] Макаренков Н.Г. Фракталы, аттракторы, нейронные сети и все такое // Лекции по нейроинформатике. Часть 2. – М.: МИФИ, 2002. – С. 121 – 168.
- [Прокопчук, 2012] Прокопчук Ю. А. Принцип предельных обобщений: методология, задачи, приложения. Монография. – Дн-вск: ИТМ НАНУ и НККАУ, 2012.- 384 с.
- [Chalmers, 1996] Chalmers D. The Conscious Mind: In Search of a Fundamental Theory. - Ox-ford University Press, 1996.

MODEL OF SPONTANEOUS COGNITIVE DYNAMICS

Prokopchuk I.A.

*Institute of Technical Mechanics of the NASU,
Dnepropetrovsk, Ukraine*

Itk3@ukr.net

This study has been aimed at constructing formal knowledge structures to simulate empirical imaginative thinking processes in computer technologies. A constructive version of the mechanism of limiting generalizations and polymodel complementarity as a paradigmatic basis is proposed, and its productivity in the solution of fundamental and applied problems in cognitive science is shown. Formal models of spontaneous ideal processes are constructed.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89

СЕМАНТИЧЕСКИЕ МОДЕЛИ ПРЕДСТАВЛЕНИЯ РЕШАТЕЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ

Шалфеева Е.А.*

* *Институт автоматизации и процессов управления,
г. Владивосток, Россия*

shalf@iacp.dvo.ru

В работе предложен набор семантических представлений разрабатываемых интеллектуальных программных систем, их содержание и структура. Предложен метод конструирования моделей решателя системы этапов анализа и проектирования на основе моделей этапа системного анализа. Эти модели необходимы для построения управляемых интеллектуальных систем, а методы являются базой для создания инструментария для поддержки разработчиков.

Ключевые слова: интеллектуальная программная система, знания, решатель, постановка задачи, онтология, модель подзадач, архитектурная модель, семантическая сеть.

ВВЕДЕНИЕ

В настоящее время продолжают работы по методам автоматизации интеллектуальной деятельности и созданию инструментария, позволяющего комплексно автоматизировать сферы экономики, процессы в которых связаны с использованием постоянно обновляемых *знаний*. Вместе с тем отмечается, что из-за «отсутствия стандартов проектирования и разработки интеллектуальных систем» проблемы, связанные с использованием прикладных моделей представления знаний и моделей обработки знаний, нуждаются в комплексной теоретической и практической доработке [Голенков и др., 2012].

Переход к декларативному представлению знаний в *интеллектуальных программных системах* (ИПС) уже внес свой вклад в сопровождаемость баз знаний. Отвечающая современным требованиям *база знаний* не только отделена в самостоятельный переносимый компонент, но и является *концептуальной*, понятной для сопровождающего ее эксперта [Грибова и др., 2011].

На успех комплексной автоматизации интеллектуальной профессиональной деятельности (ИПД) существенно влияют результаты ее системного анализа. Идентифицированы создаваемые на этом этапе модели предметной области, в числе которых и онтологии знаний и данных, и модель деятельности, и концептуальная модель системы [Клещев и др., 2012].

Представление баз знаний и данных в форме семантических сетей достаточно распространено

[Гулякина и др., 2012, Грибова и др., 2011]. Развиваются языки программирования, ориентированные на обработку таких сетей, представление данных и программ в единообразном виде [Гулякина и др., 2012].

Известны и преимущества декларативного представления программ, по сравнению с императивным: более простое написание, более легкое их понимание программистами и модифицирование, возможность замены сопровождения программы управлением ею. Предложена концепция проектирования интеллектуальных систем, все компоненты (данные, знания, решатель задач с пользовательским интерфейсом) которого имеют единые принципы для их формирования, доступа и модифицирования [Клещев и др., 2010; Грибова и др. 2012]. Разрабатываемая по этой концепции технология создания является развитием технологии построения мультиагентных систем «в сторону» улучшения их сопровождаемости. Самостоятельным компонентом ИПС с концептуальной базой знаний является решатель задач, использующий базу знаний при оказании поддержки в принятии решения.

Целью исследования является разработка метода построения ключевых моделей процесса проектирования решателей интеллектуальной системы с использованием семантических моделей системного анализа автоматизируемой профессиональной деятельности.

1. Семантические модели системного анализа

Проектирование эффективных информационных систем невозможно без использования результатов информационного моделирования и стратегического планирования [Штрик, 1993]. При автоматизации ИПД, когда проектируется наращиваемый «долгоживущий» комплекс средств поддержки принятия решения и средств доступа к знаниям и другой информации, этот первый этап является более содержательным по составу работ [Клещев и др., 2012].

Системный анализ ИПД подразумевает выработку следующих моделей (представлений предметной области и разрабатываемой в ней системы).

1. *Схема описания информационного объекта* - объекта автоматизируемой деятельности (примером такого артефакта для задачи медицинской диагностики является схема истории болезни (ИБ) пациента [Клещев и др., 2012]).

2. *Профессиональная терминология* – понятия, которые используют специалисты при описании наблюдаемой действительности и знаний о происходящих в ней процессах или существующих связях (пример терминов для медицины: наблюдения, признаки, события, особенности, измерения, возможные значения, диагнозы, планы лечения и т.д. [Клещев и др., 2005, Черняховская и др., 2007]).

3. *Перечень формируемых в процессе ИПД документов* (например: ИБ, назначения на обследования, протокол лечения, рецепты на лекарственные средства, назначение даты посещения врача).

4. *Схема решаемых задач* – (графовая) модель, упорядочивающая задачи специалистов в рассматриваемой сфере ИПД (на рис. 1 графовая модель упорядочивает интеллектуальные задачи врача – задачи описания, обследования, постановки диагноза, назначения лечения, наблюдения за состоянием – и прочие задачи медицинского персонала.

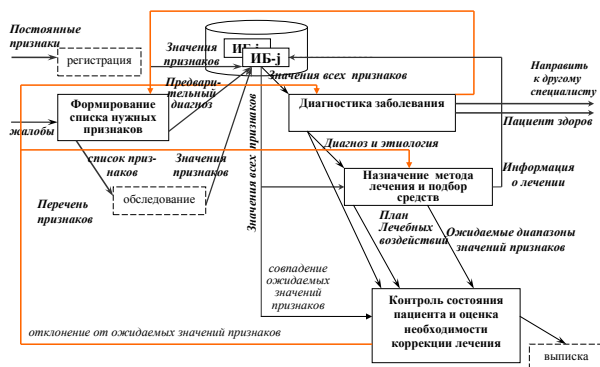


Рисунок 1 – Схема медицинских задач

5. Множество онтологий задачи - своя для

каждой интеллектуальной задачи¹ из схемы решаемых задач.

Каждая онтология задачи имеет следующие составляющие.

O1. *Онтология наблюдений* – терминология, лежащая в основе описания наблюдаемых\управляемых объектов задачи и описания знаний о процессах, в которые эти объекты вовлечены (например, терминология для истории наблюдений и знаний о диагностике, содержащая такие термины, как наблюдение, множество значений наблюдения и т.д.) [Клещев и др., 2005].

O2. *Онтология действительности* – описание структуры наблюдаемых элементов действительности, их области возможных значений; структуры принимаемого специалистом решения и возможные значения решения (например, для медицины - все, что принято включать в карту ИБ пациента).

Например, часть структуры ИБ может быть такой:

- паспортная часть,
- особенности пациента,
- (жалобы,
- опрос,
- осмотр)²
- название признака
- значение
- момент наблюдения)
- предварительный диагноз...

O3. *Онтология знаний* задает структуру хранимых знаний о протекающих процессах, об изменении состояний объекта и влияющих на это факторах и прочих внутренних и внешних связях объекта и закономерностях предметной области.

Например, часть структуры знаний может быть такой:

- ...Клиническая картина
- (главные жалобы,
- дополнительные жалобы,
- наружный осмотр,
- лабораторные исследования)
- (простой признак,
- составной признак
- характеристика признака)
- название признака
- необходимость
- варианты динамики признака

¹ Автоматизация неинтеллектуальных задач (в рамках данного исследования) считается успешно решаемой, и подсистемы, реализующие их, будут интегрироваться в общую систему на соответствующем этапе разработки.

² Здесь скобки группируют понятия одного уровня с одинаковой структурой, а «(*)» указывает на множественное вхождение понятия на этом уровне.

- (*) вариант
 - (*) период динамики
 - номер периода
 - значение признака
 - длительность периода...

04. *Соглашения о связях действительности и знаний* (иногда часть онтологии знаний) - формализация знания о зависимости результата решения от различных элементов знаний; каждый элемент онтологии действительности (получаемых результатов) должен быть связан с элементами онтологии входных данных или некоторых «промежуточных», отражающих их причинно-следственные связи [Клещев и др., 2006].

05. *Онтология документов* - архивных, отчетных и печатных, которая задает структуру этих документов (например, структура медицинского архива, т.е. всех ИБ, формат описаний схем лечения и структура рецептов).

06. *Онтология объяснения* результата решения, которая задает структуру получаемой консультации, «прописывая» требования к содержимому объяснения. Например, часть структуры объяснения - сам диагноз плюс все возможные схемы развития во времени процессов в организме пациента, соответствующие диагнозу (заболеванию):

- название заболевания
 - (*) признак наличия заболевания
 - название признака
 - ссылка на вариант динамики
 - объяснение развития этого варианта
 - (*) период развития
 - номер периода динамики этого варианта
 - значение признака, произошедшее в этот период
 - момент его наблюдения [Москаленко, 2006; Клещев и др., 2006].

6. Множество *моделей знаний* (баз знаний или наборов формализованных знаний).

Знания для каждой интеллектуальной задачи – формально представленные знания (под управлением онтологии).

Пример. При описании знаний о заболеваниях дается описание нормы: каждому наблюдению из базы наблюдений сопоставляются нормальные значения – собственное подмножество множества возможных значений этого наблюдения в простой базе наблюдений. Описание каждого заболевания состоит из его клинической картины (см. рис. 2), содержащей описания клинических проявлений, названиями которых являются названия некоторых наблюдений в простой базе наблюдений. Описание каждого клинического проявления состоит из конечного множества вариантов, не имеющих названий, а описание каждого варианта – из конечного упорядоченного множества периодов

динамики, не имеющих названий. Описание каждого периода динамики состоит из области значений и границ длительности [Черняховская и др., 2007].



Рисунок 2 – Фрагмент редактируемого описания заболевания

7. Множество *постановок задач* - своя для каждой интеллектуальной задачи (в терминах ее онтологии).

Постановка задачи включает 1) название задачи, 2) текстовое описание задачи, 3) названия и форматы (ссылки на фрагменты онтологии действительности) входных данных задачи, 4) знания о решении интеллектуальной задачи; 5) что требуется найти (в том числе объяснение решения) со ссылками на область значений самого решения (в модели знаний) и на структуру (онтологию) объяснения и сопутствующих документов; 6) описание или указание метода решения.

Пример постановки задачи диагностики заболевания» [Москаленко, 2011]:

1) Задача медицинской диагностики

2) Задача состоит в определении всех возможных альтернативных диагнозов больного на основе знаний предметной области и данных его обследования, к которым относят значения признаков (в моменты их наблюдения), значения анатомо-физиологических особенностей (постоянные во времени) и значения произошедших событий (в моменты, когда они происходили).

3) Наблюдавшиеся признаки³, моменты их наблюдения и их значения в эти моменты;

³ а также произошедшие события и наблюдавшиеся особенности

формат: фрагмент Онтологии действительности (см. О2 выше).

4) Знания о наблюдениях (признаках, событиях, анатомо-физиологических особенностях пациента) и их возможных значениях + знания о заболеваниях и длительностях их периодов развития⁴ + знания о значениях признаков при отсутствии заболеваний - знания о нормальных реакциях и знания о реакциях на воздействие событий.

5-1) все альтернативные диагнозы пациента;

формат: множество названий известных заболеваний.

5-2) объяснение каждого диагноза через объяснение всех наблюдаемых значений признаков;

формат: онтология объяснения (см. О6).

6) «Естественным методом решения частной обратной задачи медицинской диагностики является перебор всех возможных значений диагнозов, при котором для каждого заболевания выполняется решение прямой задачи - построение всех возможных схем развития ...» [Москаленко, 2011].

Наличие в постановке «Описания метода решения» увеличивает вероятность реализуемости соответствующей подсистемы для рассматриваемой задачи.

8. Множество алгоритмов решения задач - свой для каждой интеллектуальной задачи.

Алгоритм решения задачи на этапе системного анализа не обязателен для «обычных» задач. Технология систем рекомендует построение алгоритмов для решения задач на более ранних этапах для получения гарантии реализуемости соответствующей подсистемы для рассматриваемой задачи [Martin, 1989; Данилов, 2006]. Алгоритмы могут быть представлены математически, структурно, на псевдо-языке, графически. Но, как минимум, эти алгоритмы содержат информацию об отдельных решаемых подзадачах (разного уровня вплоть до элементарных) информацию об их связях, нужную для создания архитектурных проектов подсистем).

9. Концептуальная архитектура системы - схема взаимодействия подсистем – решателей задач с пользователями, с общими информационными компонентами и друг с другом. Для каждой интеллектуальной задачи требуется подсистема поддержки (решатель и пользовательский интерфейс), которая вырабатывает соответствующее объяснение своего заключения.

Пример. Врач подтверждает результат задачи либо формирует свое решение. Решения задач обследования, диагностики, лечения и коррекции являются элементами ИБ, поэтому подсистема ее

редактирования доступна врачу (рис. 3). Каждый случай расхождения решения врача с автоматическим решением сохраняется в архиве с тем, чтобы стать одним из прецедентов для будущих научных исследований по формированию новых знаний. Знания, общие для задач прогноза и лечения - фармакологические знания; знания о реакциях на воздействие событий используется в задачах мед диагностики, назначения лечебных мероприятий, мониторинга.

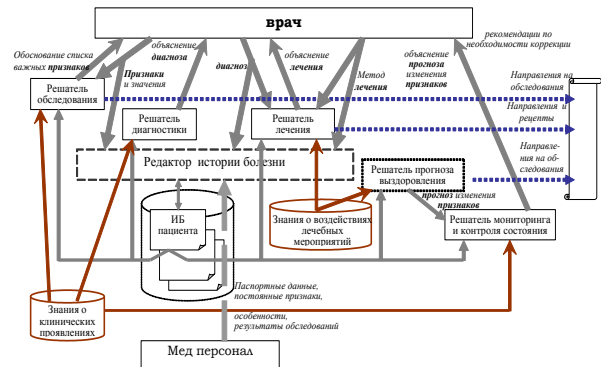


Рисунок 3 – Вариант концептуальной архитектуры медицинской системы

Структурное описание такой архитектурной модели требует несколько видов узлов и соединений между ними:

- (*) узел-подсистема;
- (*) узел-пользователь;
- (*) узел-хранилище (или адресуемый элемент хранилища);
- узел-устройство печати;
- (*) связь по информации пользователь-подсистема;
- (*) связь по информации подсистема - подсистема;
- (*) связь по информации подсистема - хранилище.

Структурированное содержание каждой модели, разрабатываемой при системном анализе позволяет иметь представление такой модели в виде семантической сети.

9. Множество документов «Пользовательские требования к решателю» или единый документ (если разные задачи ИПД решаются одним и тем же специалистом).

Ввиду того, что для формирования предыдущих моделей привлекались эксперты предметной области, которые в общем случае могут не быть конечными пользователями, задача формирования пользовательских требований к подсистемам-консультантам (решателям задач) остается почти такой же, как в традиционных разработках, за исключением того, что главные функции подсистем уже выявлены.

Документ включает «описания назначения решателя» (возможно совпадающее с названием задачи или ее текстовым описанием из Постановки

⁴ а также знания об этиологиях и знания об осложнениях, знания о причинно-следственных связях между заболеваниями и наблюдениями - знания о клинических проявлениях и т.д.

задачи) и списка сформулированных требований к интерфейсу, дополнительным функциям, эксплуатационным характеристикам.

2. Формирование функциональных моделей решателя по ранним семантическим представлениям системы

Этап детального анализа функциональности каждой разрабатываемой подсистемы (приложения) позволяет выработать одно или несколько представлений программных подсистем, на основе которых может быть создан проект качественной реализации каждой подсистемы и проект интеграции подсистем в единую развивающуюся систему.

Наиболее распространенное представление функциональности приложения (функциональный взгляд на требования к решателю) - *модель подзадач*. (Представление процесса решения в виде системы взаимодействующих объектов не является естественным, т.к. причинно-следственные связи – не объекты и не атрибуты объектов). На основе описания назначения решателя, постановки задачи решателя и пользовательских требований к решателю аналитик строит модель разбиения функциональности решателя на более мелкие подфункции (подзадачи), возможно учитывая при этом структуру (онтологию) знаний и данных.

Структура обрабатываемых данных действительности и структура знаний о связи решения (искомого результата) с элементами наблюдаемой действительности (т.е. онтология предметной области) играют здесь определяющую роль (оставляя, впрочем, «место» для творчества аналитика при выборе им необходимого числа подзадач). Тем не менее, чем больше узлов в сети-онтологии, тем больше узлов в модели подзадач.

Например, в соответствии с фрагментом онтологии знаний «Если в ситуации присутствует некоторая нормальная реакция, то в множестве знания о нормальных реакциях присутствует такой элемент, у которого следствие совпадает со следствием нормальной реакции, в множестве вариантов нормы этого элемента содержится вариант нормы из нормальной реакции и для него выполнено условие на воздействующие факторы.» (из [Клещев и др., 2005]), алгоритм должен включать шаг (*подзадачу*) «Проверить гипотезу о том, что пациент здоров», организовав Цикл по наблюдавшимся признакам и для наблюдавшегося признака проверить гипотезу о том, что все его наблюдавшиеся значения могут иметь место у здорового пациента.

Универсальным представлением для алгоритма и функционального разбиения задачи может считаться иерархия подзадач с возможностью элементу иерархии иметь полустепень захода больше единицы. Элемент иерархии – подзадача некоторого уровня (последовательная или с-ветвлением).

Элемент иерархии верхнего уровня имеет в подчинении (чтобы обращаться к ним) элементы иерархии нижнего уровня, некоторые из которых состоят в «простом» подчинении, а некоторые – в «цикличном» подчинении (как тело цикла):

```
задача {
  название;
  (*) подзадача {~alt
    подзадача в «простом» подчинении
  OR
    подзадача в «цикличном» подчинении
    (название;
    [(*) подзадача]
  });
};
```

Если на более ранних этапах математик или аналитик или инженер знаний строит *алгоритм* на основе «метода решения задачи» и связей знаний с действительностью (онтологии знаний), то специальное разбиение на подзадачи не нужно, достаточно рассмотреть альтернативное представление (в виде семантической сети) алгоритмов подзадач.

Например, в соответствии с методом решения задачи «перебор всех возможных значений выходных данных (диагнозов – отдельных заболеваний). Для каждого заболевания выполняется решение прямой задачи – построение всех возможных вариантов развития причинно-следственных связей (ПСС)... и поиск среди них всех тех, которым соответствуют все наблюдаемые значения признаков пациента» (из [Москаленко, 2006]) модель подзадач (сформулированных упрощенно) в виде сети может выглядеть так⁵:

постановка и объяснение диагноза

Получить данные наблюдений пациента (В)

Проверить гипотезу о том, что пациент здоров (П)

проработать каждый признак (Ц)

проверить гипотезу о соответствии

значений признака норме (П)

запомнить значения признака,

опровергающие гипотезу (П)

подытожить гипотезу о том, что пациент здоров. (В)

проработать каждое заболевание (Ц)

проверить выполнение необходимого условия для текущего заболевания (В)

проверить гипотезу о заболевании(П)

попытаться отвергнуть гипотезу о заболевании по не-КК-признаку (П)

проработать каждый КК-признак (Ц)

проверить возможность наблюдавшихся значений (В)

проработать каждый вариант проявления КК-признака (Ц)

составить динамику знаний варианту ПСС (П)

добавить анализ этого варианта в объяснение (П)

⁵ каждый сдвиг означает здесь подчиненность подзадачи, а буквами размечены подзадачи последовательные (П), с ветвлением (В) и в цикличном подчинении (Ц);

подытожить результаты анализа всех вариантов в объяснении КК-признака (П)

подытожить результаты анализа всех признаков в объяснении заболевания (П)

выдать результат – множество диагнозов и их объяснение (П).

Для построения этой семантической сети использованы сети этапа системного анализа. Для каждой подсистемы из *концептуальной архитектуры системы* строится своя «модель задач».

Традиционно вместе с разбиением функциональности подсистемы на подфункции (подзадачи) выполняется специфицирование входов и выходов каждой подфункции [Pressman, 2001]. *Расширенная модель подзадач* [Шалфеева, 2011] позволяет охватить информацию о каждой подзадаче и ее связях с обрабатываемыми данными (в виде сетей).

Например, подзадаче «ввод истории болезни» требуется записывать информацию от пользователя - паспортные данные, жалобы, текущее наблюдение и др. – в информационную структуру «история болезни». Подзадача формирования объяснения должна записать в соответствующий информационный ресурс (ИР) объяснение подтверждаемых и опровергаемых заболеваний. Подзадаче «Оценка гипотезы здоров» – требуется обращаться к ИБ за множеством пар <название признака, значение признака>. Подзадаче «проверить гипотезу о соответствии значений признака норме» надо сопоставлять один элемент этого множества – хранимым знаниям о нормальных значениях и т.д.

Расширенная модель подзадач добавляет в *модель подзадач* дуги-ссылки от подзадачи к компоненту обрабатываемой информации. Указаны связи подзадач «*проработать каждое...*» с компонентом-множеством, по которому организуется цикл (связь этой подзадачи с «циклично» подчиняемыми) [Шалфеева, 2011].

Пример. Узел «проработать каждый признак» имеет связь с узлом-множеством «наблюдения» структуры ИБ, узел «*проверить гипотезу о соответствии значений признака норме*» имеет связь получения информации из подструктуры «нормальные значения», узел «*подытожить гипотезу о том, что пациент здоров*» имеет связь записи информации в подструктуру «объяснение».

При разработке решателей (подсистем) для каждой интеллектуальной задачи эти модели являются ключевыми, т.к. по ним определяется внутренняя архитектура решателей и требования (спецификации) к разработке отдельных компонентов решателей. Сложность модели может быть «перенесена» на архитектуру, по мере необходимости и возможности либо архитектура либо модель подзадач может быть оптимизирована (например, для задачи диагностики может быть предложен метод сокращения списка гипотез для

проверки за счет предварительной проверки наблюдавшихся признаков в клинических картинах заболеваний-гипотез).

Единое декларативное представление всех моделей создаваемой системы [Грибова и др. 2012] позволяет модели функциональности решателя формировать в виде семантических сетей.

3. Формирование архитектурных представлений решателя по моделям анализа функциональности

Ключевыми моделями для обеспечения сопровождаемости и управляемости ИПС являются архитектурные модели. К ним, в частности, относятся: архитектура каждого решателя, архитектура хранимой информации, проекты программных единиц. Достаточной для производства проектных представлений программных подсистем является *расширенная модель подзадач*.

Построение первой версии архитектуры решателя целесообразно по более компактной модели - *модели подзадач*, оно осуществляется по схеме: 1) сопоставить подзадачам – программные единицы (ПЕ) отвечающие за выполнение соответствующей подфункции; 2) сопоставить связям подчинения среди подзадач – передачи управления между ПЕ. В рамках технологии [Грибова и др. 2012] ПЕ - агенты (или блоки агентов), среди которых можно различать обрабатывающие, сопоставляемые обычно подзадачам в «листовых» узлах семантической сети, и управляющие ПЕ, сопоставляемые часто подзадачам в узлах с ненулевой полустепенью исхода.

Параллельно с архитектурой решателя строятся «внешние» спецификации каждой ПЕ этой архитектуры. *Спецификация ПЕ* показывает входные параметры (например, требуется передать как параметр элемент множества, для которого должна быть решена «циклично-подчиненная» подзадача) и необходимость формирования ответной информации о результате решения подзадачи. Кроме того, важной составной частью *спецификации* является указание входных и выходных нелокальных данных. Эта зависимость ПЕ от хранимых данных уже прописана в *расширенной модели подзадач*.

Проект (архитектура) хранимого ИР – описание его структуры и набор спецификаций операций, достаточных для манипулирования ими. Такой вариант абстрактного типа данных позволяет иметь гибкость при дальнейшей реализации системы (удобен, как для объектно-ориентированного, так и для процедурного и агентного подходов). Проект каждого типа обрабатываемого ресурса представляется как семантическая сеть.

Обрабатывающие агенты взаимодействуют с ИР через операции. Например, подзадаче «*Получить*

данные наблюдений пациента» – соответствует обрабатывающий агент, который реагирует на сообщение-задание «ввести» с параметром «ссылка на ИБ». Агент должен записывать информацию от пользователя в ИР, для этого необходим набор операций, таких как *записать паспортные данные*, *записать жалобы*, *записать текущее наблюдение* и др.

Подзадаче формирование объяснения – соответствует агент или блок «Начать формирование объяснения», который вносит в ИР-результат ссылку на ИБ (или пациента), чей диагноз начинает устанавливаться в виде объяснения (возможных невозможных заболеваний). Требуется операция над ИР «объяснение» «добавить ссылку на ИБ» с параметрами «ссылка на ИБ» (или номер ИБ или уник идентификатор пациента), «ссылка на выходной ИР – объяснение».

Достаточность представляемых (проектируемых) наборов операций над ресурсами проверяется их присутствием в спецификациях множества проектируемых ПЕ.

Для операций (вновь разрабатываемых абстрактных типов данных) формируются спецификации на разработку.

Семантические сети продолжают оставаться средством представления все этих архитектурных моделей.

Пример архитектуры решателя. В примере упрощенной агентной версии архитектуры решателя задачи диагностики (на рис. 4) подзадаче «Проверить гипотезу о том, что пациент здоров» – соответствует блок реагирования на сообщение-задание «оценить норму» агента «Оценка гипотезы здоров» и блок реагирования на сообщение-обратная связь (того же агента).

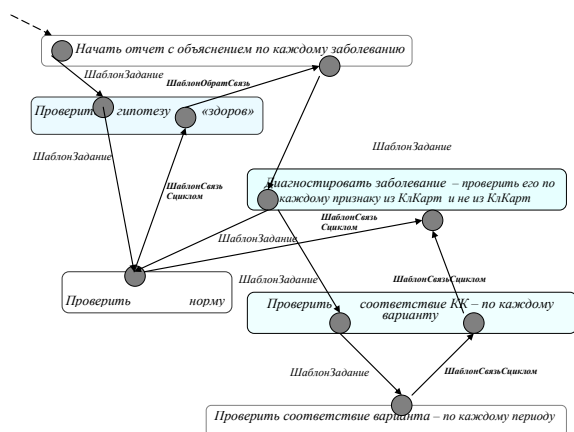


Рисунок 4 – Вариант архитектуры решателя задачи диагностики

Пример. Спецификация агента «проверить норму признака»:

«проверить норму признака» [название]

«проверяет, все ли значения указанного признака соответствуют норме» [описание]

Множество блоков продукции

Шаблон-задание [шаблон входного сообщения]

форматы используемых ИР

Онтология ИБ [формат первого используемого ИР]

Онтология описания нормы [формат второго используемого ИР]

форматы изменяемых ИР

шаблон-обратная связь [Шаблон выходного сообщения]

Пример архитектуры хранимого ИР «история болезни». Модель содержит структуру (см. выше) и список операций доступа:

«записать паспортные данные»,

«записать жалобы»,

«записать текущее наблюдение»,

«взять названия всех признаков»,

«взять уникальный идентификатор пациента»,

«взять все наблюдения указанного признака,

...

«записать предварительный диагноз».

Пример спецификации операции чтения над ресурсом ИБ:

«Взять названия всех признаков»

онтология ИБ [Формат входного ИР];

множество названий [Формат результата].

4. Заключение

Разработка описанных моделей и методов является продолжением работы по формированию методологии системного анализа и моделирования произвольных сфер деятельности с интеллектуальными процессами и концепции комплексной автоматизации для систематической поддержки принятия ответственных решений специалистами.

Предложенный метод построения ключевых моделей процесса проектирования решателей интеллектуальных программных систем состоит в разработке трех групп семантических сетей с учетом их взаимовлияния друг на друга. Предложен необходимый набор моделей каждой группы и структуры семантических сетей для их представления. Эти модели обеспечивают необходимый минимум для построения управляемых интеллектуальных систем. Предложены методы перехода от предшествующей модели к последующей. Они являются базой для создания инструментария для автоматизации интеллектуальной профессиональной деятельности.

Работа выполнена при поддержке грантов РФФИ № «12-07-00179-а «Интернет-сервисы для преобразования семантических сетей» и ДВО РАН № 12-III-A-01И-019 «Мониторинг концептуальных информационных ресурсов для управления интеллектуальными системами».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Голенков и др, 2012] Голенков В. В., Гулякина Н. А., Колб Д. Г. унификация подходов к разработке интеллектуальных систем в internet // // Материалы 22-ой международной - Крымской конференции «СВЧ-техника и телекоммуникационные технологии» (КрыМиКо'2012). Севастополь, 10—14 сентября 2012 г. Севастополь: Вебер, 2012. Т.2 С. 28-31.

[Грибова и др, 2012] Грибова В.В., Клещев А.С., Крылов Д.А., Москаленко Ф.М., Тимченко В.А., Шалфеева Е.А. Агентный подход к разработке интеллектуальных интернет-сервисов // Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'12». М.: Физматлит, 2012 –Т.1. с. 218-223.

[Грибова и др. 2011] Грибова В.В., Клещев А.С. Проблемы создания жизнеспособных интеллектуальных систем и методы их решения // International Journal "Information Technologies & Knowledge". Bulgaria. Sofia: ITHEA. 2011. Vol. 5, № 3. P. 250-258.

[Гулякина и др, 2012] Гулякина Н.А., Пивоварчик О.В., Лазуркин Д.А. Языки и технологии программирования, ориентированные на обработку семантических сетей // OSTIS-2012, стр. 221-228, Минск, БГУИР

[Данилов, 2006] Данилов Н. Н. Курс математической экономики: учеб. пособие / Н. Н. Данилов. - М. : Высшая школа, 2006. - 407 с.

[Клещев и др., 2012] Клещев А.С., Шалфеева Е.А. Системный анализ при автоматизации интеллектуальной профессиональной деятельности // XIII Национальная конференция по искусственному интеллекту с международным участием «КИИ-2012», 16–20 октября 2012 г. ISBN: 978-5-361-00182-8 Труды конференции, т.2. Белгород: Изд-во БГТУ, 2012. С.128–135.

[Клещев и др., 2010] Клещев А.С., Грибова В.В., Управление интеллектуальными системами. Известия РАН. Теории и системы управления. 2010. № 6. С. 122-137.

[Клещев и др., 2005] Клещев А.С., Черняховская М.Ю., Москаленко Ф.М. Модель онтологии предметной области «Медицинская диагностика». Часть 1. Неформальное описание и определение базовых терминов / Журнал НТИ - Серия 2. #12, 2005.

[Клещев и др., 2006] Клещев А.С., Черняховская М.Ю., Москаленко Ф.М. Модель онтологии предметной области "медицинская диагностика". Часть 2. Формальное описание причинно-следственных связей, причин значений признаков и причин заболеваний / Журнал НТИ, Серия 2, №2, 2006.

[Москаленко, 2011] Москаленко Ф.М. Задача медицинской диагностики, методы и алгоритм её решения // Владивосток: ИАПУ ДВО РАН, 2011. 40с.

[Москаленко, 2006] Москаленко Ф.М. Алгоритм диагностики, основанный на реальной онтологии медицины, для многопроцессорной ЭВМ / Доклад РАСО2006, 2006. Издатель: Институт проблем управления им. В.А. Трапезникова РАН

[Черняховская и др., 2007] Черняховская М. Ю., В. Я. Мельников, Л. П. Догадова, В. И. Негода, Москаленко Ф.М. Формальное представление знаний о конъюнктивитах // ИАПУ ДВО РАН, 2007. 51с.

[Шалфеева, 2011] Шалфеева Е.А. Использование онтологий при разработке управляемых интеллектуальных систем. Владивосток: ИАПУ ДВО РАН, 2011.

[Штрик, 1993] Штрик А.А. Эволюция методологий разработки программного обеспечения, поддерживаемых зарубежными CASE-системами. 1993 г.

[Martin 1989] Martin, J. Information Engineering: Book I - Introduction. Englewood, NJ: Prentice Hall, 1989.

[Pressman, 2001] Pressman R.S. Software Engineering: Practitioner's Approach. Fifth edition. McGraw-Hill Inc., 2001. 860 p.

* *The Institute of Automation and Control Processes, Vladivostok, Russia*

shalf@iacp.dvo.ru

The set of semantic representations of **intelligent** software, their contents and structure is offered. The construction method for the problem solver analysis and design models on the basis of domain models is offered. These models are necessary for development of managed **intelligent** software. These methods form the base for development of tools supporting of **intelligent** software developers.

INTRODUCTION

Nowadays the methods of automation of intellectual activity and of the help in acceptance of crucial decisions are being developed. In such spheres of economy decision-making is carried out with use of constantly updated knowledge.

The success of complex automation of professional activity is influenced essentially by the next results of its system and domain analysis: knowledge' and data' ontologies, activity model, and conceptual model of system.

The research objective is development of a method of creation of main designing models of **intelligent** software' problem solver with use of semantic models of the system analysis of professional activity being automated.

MAIN PART (REPORT THESES)

First all main semantic models of the domain add system analysis are presented. The formation of functional models of a problem solver on the base of early semantic representations of intelligent system is shown in the second part. The formation of a problem solver' architectural representations from the models of the analysis of its functionality is shown In the third part.

CONCLUSION

The offered method of creation of key models of designing of intelligent software' problem solver consists in development of three groups of semantic networks. The necessary set of models of each group and structure of semantic networks for their representation are offered. These models provide a necessary minimum for creation of managed intelligent software. The methods for transformation of previous model to the subsequent are offered.

THE SEMANTIC MODELS FOR REPRESENTATION OF INTELLIGENT SOFTWARE' PROBLEM SOLVER

Shalfееva E. *



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПОДХОД К ИНТЕГРАЦИИ РАЗНОРОДНЫХ МЕТОДОВ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ СЛОЖНЫХ ЗАДАЧ

Загорулько Г.Б., Загорулько Ю.А.

*Институт систем информатики им. А.П. Ершова Сибирского отделения
Российской академии наук, г. Новосибирск, Россия*

gal@iis.nsk.su

zagor@iis.nsk.su

В докладе рассматриваются подход к интеграции разнородных методов поддержки принятия решений в оболочке для построения интеллектуальных СППР. Благодаря использованию онтологии как в качестве средства представления знаний о задачах и методах поддержки принятия решений, так и в качестве универсального формата представления знаний и данных в системе, удалось не только облегчить и сделать более эффективным труд разных типов специалистов, участвующих в процессе создания и использования СППР, но и значительно упростить и унифицировать обмен информацией между разнородными компонентами и модулями СППР, реализующими различные методы поддержки принятия решений.

Ключевые слова: методы поддержки принятия решений, онтология, интеллектуальная СППР, оболочка СППР.

ВВЕДЕНИЕ

Поддержка принятия решений (ППР) является в настоящее время одним из *mainstream* как в прикладных, так и в теоретических исследованиях [Петровский, 2009]. С одной стороны, велика практическая потребность в системах поддержки принятия решений (СППР), с другой стороны, разработано много методов поддержки принятия решений (МППР), созданы инструментарии для создания СППР, развивается и сама теория принятия решений. Несмотря на очевидные успехи в этой области, практическое использование полученных в ней результатов затруднено в связи с тем, что нет целостного описания разработанных методов и средств поддержки принятия решений, как и нет единых формализмов и языков для описания МППР. Поэтому потенциальному пользователю трудно ориентироваться в многообразии имеющихся средств, а чтобы иметь возможность использовать тот или иной метод для решения своих задач, он должен вникать в тонкости его реализации. Вот почему как никогда становится актуальной задача создания единой концептуальной основы, которая помимо решения перечисленных выше проблем послужила бы базисом для решения сложных практических задач путем интеграции всех требующихся для этого методов.

В качестве такой концептуальной основы

предлагается использовать онтологию задач и методов ППР [Guarino. 1998]. Эта онтология должна иметь так называемую «вертикально-горизонтальную» структурную организацию, что позволит представлять интересующую область знаний как бы в двух измерениях. Первое измерение («вертикальная структуризация») представляется в виде традиционной иерархической структуры, на каждом уровне которой интересующая область знаний описывается с разной степенью детализации. Второе измерение («горизонтальная структуризация») задает описание области знаний с точки зрения разных типов специалистов, участвующих в процессе создания и использования СППР: лиц, принимающих решение (ЛПР), инженеров знаний, экспертов и программистов. Такая онтология позволит систематизировать многочисленные разнородные методы и средства ППР, что, в свою очередь, позволит заинтересованным лицам ориентироваться в имеющихся методах и средствах и выбирать наиболее подходящие для решения своих задач.

Для интеграции различных методов ППР в рамках одной СППР, одной онтологии задач и методов недостаточно. Для обмена данными между методами, точнее, реализующими их программными модулями и подсистемами, требуется некоторый общий формализм, на котором бы описывались входные и выходные данные различных методов. Как показала практика [Загорулько и др., 2012] в

качестве такого формализма удобно использовать также онтологию, а именно: онтологию предметной области (ПО). Онтология задач и методов ППР и онтология ПО вместе образуют концептуальный и технологический базис для интеграции различных методов ППР, тем самым создавая реальные предпосылки для создания массовой семантической технологии компонентного проектирования и разработки интеллектуальных систем такого класса, как СППР [Голенков и др., 2011].

1. Онтология задач и методов ППР

В качестве концептуального базиса, на котором строится онтология задач и методов поддержки принятия решений, была предложена метаонтология задач и методов. Эта онтология содержит описание таких базовых понятий ППР, как Задача, Метод, Модуль, Решатель, Входные данные, Результат, Ситуация, Проблемная ситуация, Альтернатива, Этап принятия решений, а также отношения между ними (см. рисунок 1).

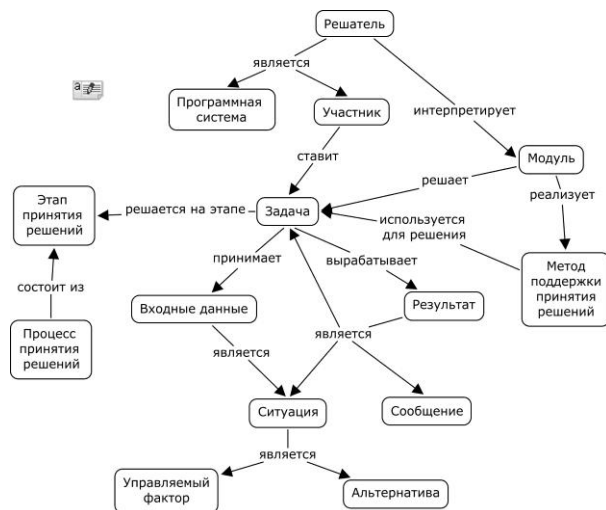


Рисунок 1 – Фрагмент метаонтологии задач и методов

Рассмотрим подробнее эти понятия.

Согласно теории [Кузнецов, 2009; Кулагин, 2001; Петровский, 2009;], процесс принятия решений состоит из нескольких этапов. На каждом этапе решаются свои задачи. Задача принимает Входные данные и вырабатывает некоторый Результат. Входными данными для задач являются Ситуации, каждая из которых представляет собой набор связанных отношениями объектов предметной области (ПО). Выделяется класс проблемных ситуаций (ПС), т.е. ситуаций, в которых значения атрибутов некоторых объектов выходят за область нормальных значений, либо критически близко подходят к ее границам. Результатом решения задачи может быть Сообщение, Ситуация или Задача. Сообщение – это окончательный результат решения задачи, который пользователь принимает к сведению. Ситуация – это результат, который может быть подвергнут дальнейшему анализу. В зависимости от того, решалась ли прямая задача или обратная, Ситуация может представлять собой

последствия принимаемых решений или же начальные решения, которые должны привести к желаемым результатам. Если в качестве решения получено несколько ситуаций – Альтернатив, то может быть сгенерирована новая Задача, которая будет оценивать полученные Альтернативы и выбирать из них наиболее приемлемые.

Для решения Задач используются различные Методы поддержки принятия решений. Некоторые из них могут иметь компьютерную реализацию, т.е. могут быть реализованы в некоем программном Модуле, который, в свою очередь, интерпретируется тем или иным Решателем. Другие методы не имеют программной поддержки. Реализующий их Модуль представляет собой текстовое (возможно, формализованное) описание данного метода, а в качестве решателя, интерпретирующего такой модуль, выступает человек – Участник процесса принятия решений. Участниками могут быть ЛПР (лицо, принимающее решение), Владельцы проблемы, различные Активные группы [Петровский, 2009], Эксперты и специалисты по принятию решений.

С использованием описанной выше метаонтологии был построен верхний уровень онтологии задач и методов поддержки принятия решений.

Как известно, основу любой онтологии составляет классификация ее основных сущностей [Гаврилова и др., 2006]. В настоящее время не существует общепринятой классификации методов поддержки принятия решений. В литературе по теории принятия решений [Орлов, 2002], как правило, дается классификации методов в рамках отдельных групп методов. При построении данной онтологии в качестве основания классификации были выбраны этапы принятия решений, а к этапам уже «привязываются» решаемые на них задачи и используемые для этого методы.

Всего в онтологии представлено 7 этапов принятия решений: Выявление проблемной ситуации, Диагностика проблемной ситуации, Постановка цели, Анализ факторов, Разработка альтернатив, Оценивание альтернатив, Выбор альтернативы.

Например, на этапе Диагностика проблемной ситуации устанавливаются причины, приведшие к ее возникновению, т.е. ставится диагноз. А для решения задачи диагностики применяются такие методы, как рассуждения на основе экспертных правил [Загоруйко и др., 1995], рассуждения на основе прецедентов [Варшавский и др., 2006], ИАД [Zagoruiiko, 2007] и др.

2. Онтология предметной области

Для того чтобы СППР была настраиваема на любую ПО, онтологии, описывающие разные ПО, должны быть одинаково устроены. Это задается путем использования в качестве основы для

построения онтологии конкретной ПО метаонтологии, включающей базовые понятия, общие для всех ПО.

Разработанная метаонтология ПО включает следующие базовые понятия и отношения: Объект управления, Идеальный объект, Состояние, Результат (рисунок 2). Объекты могут находиться в том или ином состоянии, для описания которого вводится класс Состояние. Для описания штатного состояния объекта управления, в котором объект управления должен находиться и на достижение которого должна быть направлена вся деятельность ЛПР, вводится специальное понятие Идеальный объект. В качестве результатов работы СППР могут быть: Диагноз текущего состояния объекта (объектов) управления, Рекомендация для ЛПР, Прогноз изменения состояния объекта и т.п. Отдельным результатом может быть Задача, которую требуется решить, чтобы получить дополнительную информацию для принятия решения.

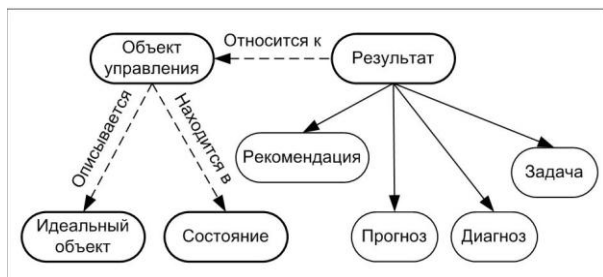


Рисунок 2 – Фрагмент метаонтологии предметной области

Так как метаонтология ПО определяет только понятия самого верхнего уровня, характерные для всех СППР, независимо от решаемого класса задач, то при специализации СППР на определенную проблемную область и для облегчения работы экспертов и инженеров знаний могут разрабатываться онтологии базового уровня, которые ориентированы на определенную проблемную область и могут использоваться для построения онтологии конкретной ПО.

Например, оболочка для построения СППР, ориентированных на поддержку принятия решений в технических областях, может снабжаться базовой онтологией диагностики и мониторинга производственных объектов [Zagorulko et al., 2010], которая включает помимо введенных в метаонтологии ПО базовых понятий Объект управления, Состояние, Результат, Идеальный объект, понятия, полученные путем специализации перечисленных выше понятий. Так, в классе Объект управления могут выделяться подклассы Оборудование, Подвижные объекты, Дорожная сеть. В свою очередь класс Состояние может иметь следующие подклассы: Неисправность, Поломка, Предаварийное состояние и др.

3. Основные принципы интеграции разнородных методов ППР

Как уже было сказано выше, в соответствии с предлагаемым подходом интеграция МППР выполняется на основе концептуального базиса задаваемого онтологиями задач и методов ППР и предметной области.

Онтология задач и методов ППР содержит описание каждого метода, показывает его место в задаче (системе задач) ППР и задает его взаимосвязи с другими методами. Благодаря таким свойствам при решении задачи интеграции различных методов эта онтология может использоваться как экспертами и инженерами знаний, осуществляющими выбор подходящих методов, так и программистами (конструкторами СППР), устанавливающими порядок взаимодействия методов.

Онтология ПО задает структуры для описания входных и выходных данных методов и отслеживает их логическую целостность. Эти структуры, представляемые понятиями онтологии ПО, являются универсальными и поэтому «понятными» для всех методов. Благодаря этому, при решении задачи интеграции различных методов данная онтология используется для организации обмена данными между методами. В частности, в СППР все данные, как входные и выходные, должны быть описаны в терминах понятий онтологии.

Возможны разные способы интеграции методов ППР: начиная с интеграции на уровне модулей и подсистем, реализующих эти методы ППР, и кончая интеграцией на уровне единого формализма (модели) [Загорулько, 2013].

Интеграция на уровне компонентов безусловно сужает класс потенциально-решаемых задач. Однако такой способ интеграция дает возможность более быстрого практического решения многих сложных задач, поскольку имеются уже готовые компоненты для решения их отдельных подзадач.

При интеграции МППР на уровне компонентов методы будут работать последовательно, принимая входные данные и передавая результат своей работы другим методам. При этом для организации работы неважно, как реализован и на каких принципах основан метод. Важно только уметь его запустить, правильно передать ему исходные данные и выгрузить результат.

Так как у каждой интегрируемой подсистемы (модуля), реализующей тот или иной метод, согласно [Guarino, 1998] есть своя онтология, каждая такая подсистема имеет свой формат представления данных. В связи с этим требуется организовать отображение входных данных, представленных в онтологии ПО, в формат онтологии интегрируемой системы, а затем (после отработки метода) обеспечить отображение полученных выходных данных в формат, заданный онтологией ПО. Поэтому для каждого метода

должен быть создан конвертор, переводящий данные в его внутренний формат и наоборот.

ЗАКЛЮЧЕНИЕ

Предложен подход к интеграции разнородных методов поддержки принятия решений для сложных задач, базирующийся на системе онтологий. Благодаря использованию онтологии как в качестве средства представления знаний о задачах и методах поддержки принятия решений, так и в качестве универсального формата представления знаний и данных в системе, удалось не только облегчить и сделать более эффективным труд разных типов специалистов, участвующих в процессе создания и использования СППР, но и значительно упростить и унифицировать обмен информацией между разнородными компонентами и модулями СППР, реализующими различные методы поддержки принятия решений.

Работа выполнена при финансовой поддержке Президиума РАН (интеграционный проект СО РАН № 15/10 «Математические и методологические аспекты интеллектуальных информационных систем»).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Гаврилова и др., 2006] Гаврилова, Т.А. Модели и методы формирования онтологий / Т.А. Гаврилова, Д.В. Кудрявцев, В.А. Горовой // Научно-технические ведомости СПбГПУ, № 4, 2006. – С.21-28.

[Голенков и др., 2011] Голенков, В.В. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н.А. Гулякина // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2011): мат. Междунар. научн.-техн. конф. – Минск: БГУИР, 2011. – С. 21-58.

[Варшавский и др., 2006] Варшавский, П.Р. Методы правдоподобных рассуждений на основе аналогий и прецедентов для интеллектуальных систем поддержки принятия решений / П.Р. Варшавский, А.П. Еремеев // Новости Искусственного Интеллекта, № 3, 2006, С. 39-62.

[Загорулько и др., 2012] Загорулько, Ю.А. Онтологический подход к разработке системы поддержки принятия решений на нефтегазодобывающем предприятии / Ю.А. Загорулько, Г.Б. Загорулько // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2012. Том.10, выпуск 1. –С. 121-128.

[Загорулько, 2013] Загорулько, Ю.А. Технологии разработки интеллектуальных систем, основанные на интегрированной модели представления знаний / Ю.А. Загорулько // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013): мат. Междунар. научн.-техн. конф. – Минск: БГУИР, 2013.

[Загорулько и др., 1995] Загорулько, Ю.А. Интегрированная технологическая среда для создания систем обработки знаний / Ю.А. Загорулько, И.Г. Попов, В.В. Щипунов // Известия РАН. Теория и системы управления. –1995. – № 5. – С. 210–213

[Кузнецов, 2009] Кузнецов, О.П. Интеллектуализация поддержки управляющих решений и создание интеллектуальных систем / О.П. Кузнецов // Теория и методы разработки программного обеспечения систем управления. Спецвыпуск журнала «Проблемы управления» №3.1, 2009.

[Кулагин, 2001] Кулагин О.А. Принятие решений в организациях. – СПб.: Издательский дом «Сентябрь», 2001.

[Петровский, 2009] Петровский А.Б. Теория принятия решений. М.: Издательский центр «Академия», 2009.

[Guarino, 1998] Guarino, N. Formal Ontology in Information Systems / N. Guarino // Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, 6–8 June 1998. – Amsterdam: IOS Press, 1998. – P. 3–15.

[Zagoruiko, 2007] Zagoruiko, N.G. Ontology of the Data Mining Subject Domain / N.G. Zagoruiko, S.E. Gulyaevskii, and B.Ya. Kovalerchuk // Pattern recognition and image analysis. Vol. 17. No. 3. 2007.

[Zagorulko et al., 2010] Zagorulko, Yury. An Approach to Development of the Decision Support System for Enterprise with Complex Technological Infrastructure / Yury Zagorulko, Galina Zagorulko. // Bulletin of NCC. – Issue 31.– 2010.–P. 195–207.

APPROACH TO INTEGRATION OF HETEROGENEOUS METHODS OF DECISION SUPPORT FOR COMPLICATED PROBLEMS

Zagorulko G.B., Zagorulko Yu.A.

*A.P. Ershov Institute of Informatics Systems
Siberian Branch of the Russian Academy of
Sciences, Novosibirsk, Russia*

zagor@iis.nsk.su

gal@iis.nsk.su

The paper discusses an approach to integration of heterogeneous methods of decision support for program shell intended for development of intelligent decision support systems (IDSS).

In the frameworks of this approach an ontology is used both as means for representation of knowledge about tasks and decision support methods, and as a universal format for knowledge and data representation in IDSS. Due to this fact, we managed both to make a labour of all types of specialists, participating in process of creating and using the IDSS, more easy and effective, and considerably simplify and unify an information exchange between heterogeneous subsystems and modules implementing various decision support methods.

The approach is based on two ontologies: the subject domain ontology (SD ontology) and the task and decision support methods ontology (TDSM ontology).

The TDSM ontology includes descriptions of all well-known methods. It shows place of each method in system of decision support tasks and defines its interconnections with other methods. Due to these properties of the TDSM ontology, it can be used during integrating heterogeneous methods of decision support as experts and knowledge engineering selecting a appropriate method as programmers (IDSS designers) establishing cooperation of methods.

The SD ontology defines structures for describing input and output data of methods and keeps track of their integrity. These structures represented by concepts of the SD ontology are universal and, therefore, clear for all methods. Due to this fact during integrating heterogeneous methods of decision support this ontology is used for providing data exchange between heterogeneous methods. In particular, all data to be used in the IDSS could be described in terms of the SD ontology concepts.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

МОДЕЛИ И СРЕДСТВА КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ МАШИН ОБРАБОТКИ ЗНАНИЙ НА ОСНОВЕ СЕМАНТИЧЕСКИХ СЕТЕЙ

Шункевич Д.В.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

shu.dv@tut.by

В данной работе рассматриваются проблемы существующих методов, средств и технологий построения машин обработки знаний и ставится проблема отсутствия средств, позволяющих относительно неподготовленному разработчику в удовлетворительные сроки проектировать машины обработки знаний для прикладных интеллектуальных систем различного назначения. Далее рассматривается технология, призванная решить поставленную проблему путем интеграции различных методов и способов решения задач на общей формальной основе.

Ключевые слова: интеллектуальные системы; машины обработки знаний; решатель задач; компонентное проектирование; многоагентные системы.

ВВЕДЕНИЕ

В настоящее время особенно актуальными становятся проблемы обработки знаний в интеллектуальных системах. Вопросы представления знаний различного вида на настоящее время рассматриваются достаточно широко, существует большое количество языков представления знаний различной мощности и сложности, а также моделей представления знаний [Гаврилова и др., 2001].

Машина обработки знаний, включающая информационно-поисковую машину, интеллектуальный решатель задач и набор служебных операций обработки знаний (операции сборки мусора, выявления противоречий в базе знаний и т.д.), является важнейшей частью любой интеллектуальной системы, т.к. именно возможностями машины обработки знаний определяется функционал системы в целом, возможность давать ответы на нетривиальные вопросы пользователя и способность решать различные задачи.

Однако большинство прикладных интеллектуальных и экспертных систем [Гаврилова и др., 2001] имеют один и тот же недостаток – они не позволяют в должной мере обеспечить обработку тех знаний, которые в них содержатся.

Те же прикладные системы, которые обладают встроеной машиной обработки знаний,

предоставляют пользователю жестко ограниченный функционал, заданный разработчиком на этапе проектирования системы. Примером таких машин обработки знаний может служить машина дедуктивного вывода, представленная в ряде экспертных систем [Ефимов, 1982], или машина нечеткого вывода.

Машина обработки знаний каждой конкретной системы во многом зависит от назначения данной системы, множества решаемых задач, предметной области и другими факторами. Например, в системе, решающей задачи по геометрии, химии и другим естественным наукам обоснованным будет использование дедуктивных методов вывода, поскольку решение задач в таких предметных областях основывается только на достоверных правилах. В системах же медицинской диагностики, к примеру, постоянно возникает ситуация, когда диагноз может быть поставлен только с некоторой долей уверенности и абсолютно достоверным ответ на поставленный вопрос быть не может. В связи с этим **возникает необходимость использования различных машин обработки знаний в различных системах**, при этом состав и возможности машины обработки знаний в конкретной системе определяется не только непосредственно разработчиком, а требует консультаций с экспертами в данной предметной области.

При проектировании машин обработки знаний интеллектуальных систем, как и при проектировании любых программных систем,

возникает ряд трудностей, связанных с переносимостью разработанного программного обеспечения на различные платформы, обеспечению возможности его последующей доработки (в том числе и сторонними разработчиками), универсализации методов принципов решения поставленных задач.

Основная проблема, рассматриваемая в данной работе, заключается в отсутствии средств, позволяющих относительно неподготовленному разработчику в удовлетворительные сроки проектировать машины обработки знаний для прикладных интеллектуальных систем различного назначения. Под неподготовленным разработчиком здесь понимается лицо, не имеющее специальной подготовки непосредственно в области разработки машин обработки знаний, однако имеющее представление об особенностях текущей предметной области и обладающее базовыми техническими навыками в работе с современными компьютерными средствами. Примером может служить эксперт-профессионал, тесно связанный с предметной областью, для которой разрабатывается система.

В связи с этим возникает необходимость создания универсальной технологии проектирования машин обработки знаний, обладающей следующими свойствами:

- **Универсальность.** Проектируемая технология должна обеспечивать возможности для обработки знаний и решения произвольных классов задач в различных предметных областях, не требуя при этом вмешательства пользователя данной разработки в ее внутреннее устройство.

- **Модульность и расширяемость.** Проектируемая технология должна предоставлять возможность расширения функционала системы, без изменения базовой модели машины обработки знаний.

- **Кроссплатформенность.** Проектируемая технология не должна зависеть от операционной системы и аппаратной архитектуры устройства, на котором предполагается работа информационной системы.

- **Параллельность.** Проектируемая технология должна обеспечивать возможность параллельного использования различных способов решения задач в рамках решения одной задачи, а также возможность параллельного решения сразу нескольких задач. При этом необходимо обеспечить согласованность и интегрируемость результатов применения различных методик решения задач.

- **Обоснованность.** Машина обработки знаний, в частности интеллектуальный решатель задач, построенный на базе предлагаемой технологии должен в случае необходимости указать пользователю правила вывода, на которых базируется решение той или иной задачи, другими словами построить алгоритм решения поставленной задачи в виде, понятном пользователю.

В данной работе рассматриваются основные принципы построения универсальной семантической технологии проектирования машин обработки знаний интеллектуальных систем, обладающей всеми описанными выше достоинствами. Данная технология является частью открытой семантической технологии проектирования интеллектуальных систем OSTIS [OSTIS, 2012].

1. Анализ существующих машин обработки знаний различного рода

В качестве наиболее заметных представителей машин обработки знаний в интеллектуальных системах можно указать следующих:

- GPS (General Problem-Solver)
- QA3
- STRIPS (Stanford Research Institute Problem Solver)
- ПРИЗ (Пакет прикладных инженерных задач)
- ППР (Программа принятия решений)
- УДАВ (Универсальный делатель алгоритмов Варламова)

1.1. Анализ системы GPS (General Problem-Solver)

При разработке GPS авторов в основном интересовали вопросы, связанные с поисковой деятельностью человека, решающего задачи. Это привело к созданию известной эвристической стратегии поиска решений, используемой в различных дальнейших модификациях решателей. Однако, стремясь создать теорию мышления на подобной основе, авторы не уделили должного внимания другому важному аспекту теории — представлению знаний. В результате GPS не оказался универсальным решателем задач, на что надеялись его создатели. Решатель по существу имел процедурный язык низкого уровня, на котором, как показали, например, шахматы, далеко не всегда оказалось возможным эффективное описание сложных сред в терминах априори упорядоченных различий, таблиц связей, операторов и других элементов проблемной среды GPS. Поэтому, несмотря на довольно эффективную саму по себе стратегию поиска (анализ целей и средств, планирование и др.), система решала задачи медленно. Здесь сказались нерешенность проблемы совмещения эффективной стратегии поиска с эффективным представлением знаний.

1.2. Анализ вопросно-ответной системы QA3

Вопросно-ответная система QA3 может быть также названа многоцелевой системой решения задач или общим решателем задач. Она рассчитана на произвольную предметную область и произвольные вопросы, ее действие основано на автоматическом доказательстве теорем с использованием принципа резолюций. Но так как в

рамках формализма метода резолюций оказалось затруднительным описание эвристик, то это обстоятельство заставило отказаться в QA3 от эвристического поиска. Таким образом, попытка построить дедуктивный решатель, используя в полной степени формализм принципа резолюции, оказалась, как показала система QA3, также unsuccessful.

1.3. Анализ системы STRIPS (Stanford Research Institute Problem Solver)

Система STRIPS (Stanford Research Institute Problem Solver), использует декларативно-процедуральное представление знаний в сочетании с эвристическим поиском. Эта особенность в сочетании с использованием макрооператоров, формируемых на основе обучения решателя STRIPS, позволила значительно повысить его эффективность. Улучшив, таким образом, стратегию поиска решений, авторы STRIPСа тем не менее не сумели решить ряд возникших на их пути проблем. Наиболее серьезной из них оказалась проблема так называемых побочных эффектов. Оказалось, что принципиально невозможно, оставаясь в рамках подобного описания действий, априори предусмотреть и описать полный эффект действий, т. е. что действительно меняется в результате применения данного оператора к конкретной ситуации. [Ефимов, 1982]

1.4. Анализ системы ПРИЗ (Пакет прикладных инженерных задач)

Ядром системы ПРИЗ (Пакет прикладных инженерных задач) служит организующая программа, не ориентированная априори на какую-либо предметную область. В наиболее общем режиме решатель по задаче, заданной текстом, формирует ее описание и далее составляет и исполняет решение задачи. Знания о предметной области составляют содержание пакета системы ПРИЗ и в процедурной форме представляют собой множество вычислительных моделей и программных модулей. Система ПРИЗ не планирует вычислительный процесс, составляющий решение заданной задачи, в полном объеме. Обычно в текстовом описании задачи содержится информация, по которой формируется управляющая программа, представляющая собой последовательность требуемых подзадач. Таким образом, ПРИЗ планирует решения только типовых подзадач при заданном скелете решения задачи в целом. [Ефимов, 1982], [Кахро и др., 1988]

1.5. Анализ системы ППР (Программа принятия решений)

В системе ППР (Программа принятия решений) знания о предметной области представлены в пространстве признаков в виде растущих пирамидальных сетей (РПС), которые строятся автоматически. С помощью таких сетей удается хранить в системе необходимую информацию в компактном виде (общие для нескольких объектов

признаки соответствуют одной вершине РПС), естественным образом организовать процедуру обучения системы в пространстве признаков и формировать понятия, характеризуемые своим объемом. В ППР поиск решений включает в число процедур построение дерева возможностей, эвристический поиск на дереве наилучшей ветви, анализ достижимости целей и механизм возврата в случае неудачи. Для увеличения эффективности поиска введены: двунаправленный поиск; представление в виде РПС знаний, описываемых на языке предикатов; процедура формирования рабочей информации в зависимости от решаемой задачи и процедуры формирования и применения макрооператоров. При всем при этом ППР представляет одноуровневую систему планирования и не использует процедурные языки, что не позволяет считать успешно решенной в этой системе проблему эффективного поиска. [Ефимов, 1982]

1.6. Анализ программного комплекса УДАВ (Универсальный делатель алгоритмов Варламова)

В программном комплексе «УДАВ» реализован «универсальный делатель алгоритмов Варламова». Этот метод базируется на миварной логической сети правил и представляет возможность активного обучения логического вывода, управляемого потоком данных, со снижением вычислительной сложности с $N!$ (факториал) до линейной. «Универсальный делатель алгоритмов Варламова» работает со знаниями, представленными в виде продукционных правил и процедур. [Владимиров и др., 2010]

Следует отметить, что ни один из описанных примеров существующих машин обработки знаний не удовлетворяет всем требованиям, предъявленным к машинам обработки знаний во введении к данной статье.

Одним из основных преимуществ предлагаемой технологии является ее ориентация на параллельную обработку знаний. Широкие возможности для реализации параллелизма обусловлены следующими моментами:

- Основными компонентами решателя являются sc-операции, по сути представляющие собой автономные самостоятельные агенты над общей памятью;
- Процедуры, реализующие операции решателя могут быть описаны как параллельные программы. Внутренний язык программирования SCP [Голенков и др., 2001], являющийся основным языком реализации процедур решателя, изначально является языком параллельного программирования.

Сама концепция использования графодинамической ассоциативной памяти как среды взаимодействия операций предоставляет широкие возможности для параллелизма. Единственным условием в данном случае является

наличие в реализации памяти стандартных механизмов синхронизации, например, таких как блокировки.

Более подробно структура предлагаемой модели машины обработки знаний описана в следующем разделе данного проекта.

2. Общая модель и структура машин обработки знаний, построенных на основе технологии OSTIS

2.1. Унифицированная модель машин обработки знаний, разрабатываемых на основе технологии OSTIS.

В предлагаемом подходе к построению машин обработки знаний сама машина рассматривается в неклассическом варианте. В данном случае машина обработки знаний представляет собой графодинамическую sc-машину (память в качестве модели представления знаний использует семантическую сеть), состоящую из двух частей:

- графодинамическая sc-память;
- система sc-операций (sc-агентов).

Система операций является агентно-ориентированной и представляет собой набор sc-операций, условием инициирования которых является появление в памяти системы некоторой определенной конструкции. При этом операции взаимодействуют между собой через память системы посредством генерации конструкций, являющихся условиями инициирования для другой операции. При таком подходе становится возможным обеспечить гибкость и расширяемость функционала системы путем добавления или удаления из ее состава некоторого набора операций.

Отличительной особенностью машины обработки знаний как многоагентной системы в рамках данного подхода является принцип взаимодействия операций-агентов. По сути, предлагаемый подход реализует принцип «доски объявления», рассматриваемый в теории многоагентных систем [Тарасов, 2002]. Агенты обмениваются сообщениями исключительно через общую память путем использования соответствующего языка взаимодействия (языка вопросов-ответов, рассматриваемого далее), в отличие от большинства классических МАС, в которых агенты обмениваются сообщениями непосредственно друг с другом. В рассматриваемом подходе каждый агент, формулируя вопросную конструкцию в памяти, априори не знает, какой из агентов будет обрабатывать указанную конструкцию, а лишь дожидается появления в памяти факта окончания обработки вопроса. При этом в решении поставленной таким образом задачи может принимать участие целый коллектив агентов. Аналогичным образом, реагируя на появление некоторой конструкции в памяти, агент в общем случае не знает, кто из его коллег поставил данный

вопрос, а лишь может проверить соответствие сгенерированной конструкции своему условию инициирования. В случае наличия такого соответствия, агент начнет обработку указанного вопроса (решение поставленной задачи), и в результате работы сгенерирует некоторый ответ на поставленный вопрос.

Проверка соответствия сгенерированного вопроса условиям инициирования агентов происходит следующим образом: автору вопроса после его формулирования необходимо инициировать данный вопрос (включить его во множество инициированных вопросов). После инициирования вопроса каждый из агентов, работающих в памяти, переходит в активное состояние и начинает проверку условия инициирования. При этом проверка начинается с наиболее уникальных фрагментов условия (например, типа вопроса) с целью оптимизации данного процесса. В случае установления факта изоморфности вопросной конструкции и условия инициирования агент начинает решение поставленной задачи, в противном случае агент переходит в состояние пассивного ожидания.

Описанная модель взаимодействия агентов в общей памяти позволяет обеспечить максимальную расширяемость системы агентов и предельно упростить процесс добавления новых агентов в уже имеющийся коллектив.

Следует также отметить немаловажный момент: для описания процедур, реализующих принципы работы того или иного агента (т.е. программ агента [Рассел, Норвиг]) используется специализированный язык SCP, построенный на базе SC-кода, как и в случае с представлением знаний, предназначенных для обработки. Такой подход имеет ряд преимуществ:

- И программа агента, и обрабатываемые знания, по сути, представлены на одном и том же языке. В связи с этим преобразование восприятий агента в его действия, описываемое функцией агента [Рассел, Норвиг 2006], значительно упрощается, т.к. отсутствует необходимость дополнительных преобразований во внутреннее представление агента;
- Так как алгоритм работы агента описан на том же языке, что и другие знания в системе, то появляется возможность модифицировать сам алгоритм того или иного агента прямо в процессе его работы. Это предоставляет широкие возможности для построения принципиально нового класса программ и, соответственно, агентов, способных к самоконфигурированию в процессе работы.

2.2. Иерархическая структуризация машин обработки знаний, разрабатываемых на основе технологии OSTIS.

Для определения структуры рассматриваемой модели машины обработки знаний рассмотрим

более подробно процесс поиска ответа на вопрос интеллектуальной системой.

Определим два основных понятия, используемых ниже.

Под *стратегией решения задачи* понимается общий, недетализированный план решения задачи, способ достижения поставленной цели

Стратегия решения задает принцип и порядок обхода объектов в рамках семантической окрестности вопроса.

Под *операцией логического вывода* понимается некоторый *sc*-агент, который получает на вход теоретико-множественную пару $\{S, O\}$, где

S - логическое утверждение произвольной конфигурации

O - совокупность объектов, в семантической окрестности которых необходимо применить утверждение *S*.

Целью такого агента является генерация в памяти новых знаний на основании уже имеющихся, т.е. по сути, применение утверждения *S*.

Указанный процесс поиска ответа можно разделить на следующие этапы:

- Этап работы поисковых операций.

Вне зависимости от типа поставленного вопроса всегда имеется вероятность того, что данная задача уже была решена системой ранее или системе уже откуда-либо известен ответ на поставленный вопрос. На данном этапе работу осуществляет коллектив поисковых операций, каждая из которых, как правило, соответствует некоторому классу решаемых задач. Если ответ найден, подсистема обработки знаний прекращает свою работу. В противном случае происходит переход на следующий этап решения.

- Этап применения стратегий решения задач.

На данном этапе осуществляется выбор между различными стратегиями решения задач, и, при необходимости, параллельный запуск различных стратегий. Целью работы каждой из стратегий является получение набора пар, связывающих некоторое множество объектов и логическое утверждение из базы знаний, которое справедливо для классов, которым принадлежат эти объекты в рамках некоторой теории. Впоследствии при рассмотрении каждого утверждения осуществляется попытка применить его в рамках некоторой семантической окрестности рассматриваемых объектов, для чего осуществляется переход на следующий этап решения.

- Этап применения правил логического вывода.

На данном этапе происходит попытка применения утверждения, полученного на

предыдущем шаге, с целью генерации в системе новых знаний. Если такое применение справедливо (например, посылка истинна) и имеет смысл (в результате применения будут сгенерированы новые знания), то осуществляется генерация новых знаний на основе одного из правил логического вывода. При этом применение происходит в контексте объекта, рассматриваемого на предыдущем этапе (в общем случае – ряда объектов). Если в данном контексте вывод на основе данного утверждения невозможен или нецелесообразен, решение возвращается на предыдущий этап. В случае успешного применения утверждения происходит переход к следующему этапу решения.

- Этап верификации и оптимизации сгенерированных знаний и сборки мусора.

На данном этапе происходит интерпретация арифметических отношений, сгенерированных в процессе решения на предыдущем этапе, то есть попытка вычисления недостающих значений компонентов связок арифметических отношений (например, сложение величин и произведение величин) на основе имеющихся значений. Если вычислить все недостающие значения не представляется возможным, то все знания, сгенерированные на предыдущем этапе, уничтожаются и решение переходит на этап применения стратегий. В таком случае применение логического вывода для рассматриваемого на предыдущем шаге утверждения считается не целесообразным. Также на данном этапе происходит устранение синонимии, если таковая появилась на предыдущем этапе решения, например, сгенерирована связка отношения совпадения между некоторыми объектами. В конечном итоге происходит удаление конструкций, ставших ненужными и по каким-либо причинам не удаленных на предыдущих этапах решения.

Если все этапы решения выполнены успешно, то решение возвращается к первому этапу, и в случае, если ответ не получен, процесс повторяется еще раз. Стоит отметить, что в процессе решения один и тот же объект или одно и то же высказывание могут быть использованы многократно, если это целесообразно. Однако, очевидно, что применение одного и того же утверждения для одного объекта несколько раз не имеет смысла, при условии, что нужные знания из памяти не удаляются в процессе решения какими-либо сторонними операциями.

Следует учитывать тот факт, что агенты сборки мусора, устранения синонимии и верификации знаний могут оказаться полезными и необходимыми не только на завершающем этапе работы интеллектуального решателя задач. В этом смысле 4-ый этап является несколько размытым и может быть частично интегрирован с какими-либо из предыдущих.

Таким образом, в структуре описываемой модели можно выделить 4 логических уровня, на

каждом из которых возможно использование методов параллельной обработки информации.

Следует также отметить, что использование такой многоуровневой модели позволяет улучшить производительность системы в целом за счет попеременного «выключения» некоторых уровней в процессе решения сложной задачи, т.е. искусственному переводу агентов, соответствующих некоторому уровню в неактивное состояние. К примеру, на этапе применения правил логического вывода могут быть отключены агенты, реализующие стратегии решения задач. Это уменьшит число ненужных срабатываний и проверок условий инициирования агентов при возникновении в памяти некоторого события. Однако такие действия могут негативно сказаться на возможностях системы в целом, к примеру, станет невозможным одновременное решение нескольких задач, поскольку разные задачи могут на один и тот же момент времени находиться на разных этапах решения, что делает отключение какого-либо этапа недопустимым.

Структуру такой параллельной асинхронной модели машины обработки знаний можно наглядно изобразить в виде следующей диаграммы:

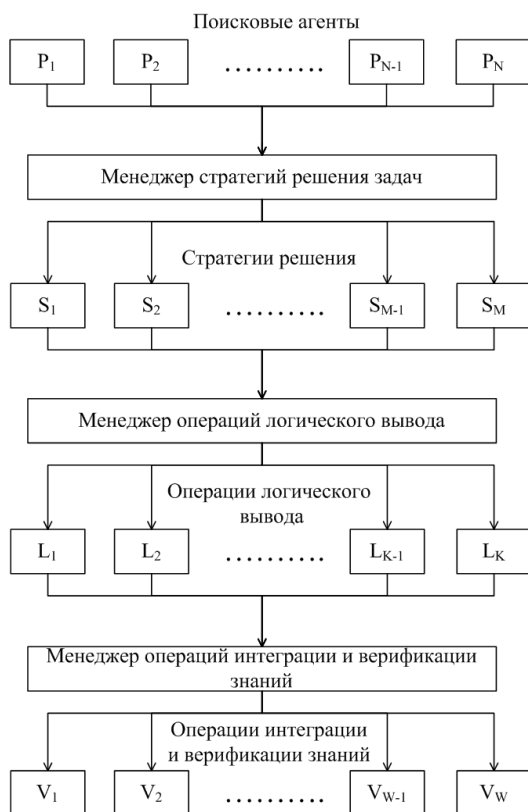


Рисунок 1 – Структура модели обработки знаний

В рамках каждого менеджера возможно использование принципа векторного параллелизма. Действительно, набор стратегий решения или операций логического вывода, по сути, представляет собой вектор. Каждый элемент данного вектора может быть рассмотрен по отдельности и активирован независимо от других в асинхронном режиме. Это позволяет говорить о

возможности параллельного использования в рамках решения одной задачи нескольких стратегий, операций логического вывода, операций верификации и интеграции знаний. Таким образом, можно считать, что каждый из менеджеров реализует концепцию агента-супервизора, рассматриваемого в теории многоагентных систем, т.е. некоторого метаагента, основной задачей которого является координация действий других агентов. [Тарасов, 2002]

3. Классификация агентов

Совокупность агентов, осуществляющих обработку знаний в интеллектуальной системе, можно декомпозировать на основе различных критериев. В данной работе мы рассмотрим два основных способа классификации агентов.

По функциональному назначению агенты можно декомпозировать следующим образом:

- **Поисковые агенты.** Основной задачей данного класса агентов, как следует из названия, является осуществление попытки поиска готового ответа на поставленный вопрос. Таким образом, для каждого класса решаемых системой задач необходимо наличие хотя бы одного соответствующего поискового агента.

- **Агенты, реализующие стратегии решения.** К данному классу относятся агенты, которые реализуют принципы решения задач, заложенные в ту или иную стратегию решения. Для каждой имеющейся в системе стратегии необходимо наличие хотя бы одного агента, реализующего данную стратегию.

- **Агенты логического вывода.** К данному классу относятся агенты, предназначенные для генерации новых знаний на основе некоторых логических утверждений. Количество и разнообразие таких агентов зависит от типологии логических утверждений, которые предполагается использовать в прикладной интеллектуальной системе.

- **Агенты интерпретации программ.** Агенты данного класса предназначены для интерпретации программ, записанных как на внешних (с точки зрения системы), так и на внутренних языках системы. Количество таких агентов в конкретном случае зависит от интенсивности использования готовых программ в решении задачи и может равняться нулю, если готовые программы не предполагается использовать по каким-либо причинам.

- **Агенты-мусорщики.** Агенты данного класса предназначены для удаления из памяти системы информационного мусора, т.е. конструкций, сгенерированных ранее каким-либо агентом, и потерявших актуальность либо семантическую ценность. Агенты-мусорщики могут ориентироваться либо на типовые конструкции,

либо на нетипичные конструкции, помеченные каким-либо специальным образом.

- **Агенты интеграции знаний и устранения противоречий в базе знаний.** Данный класс агентов предназначен для обеспечения корректной интеграции в системе новых знаний с уже существующими, а также для верификации знаний. Агенты данного класса могут активироваться не в процессе решения системой конкретной задачи, а в произвольный момент при возникновении необходимости верификации знаний, как это указано в разделе 2.

На основании внутренней структуры агенты можно декомпозировать следующим образом:

- **Атомарные агенты.** Под атомарным агентом понимается агент, не содержащий в своем составе других агентов.

- **Составные агенты.** Под составным агентом понимается агент, в составе которого можно выделить ряд более простых агентов. Примером может служить агент, реализующий некоторую стратегию решения задач и имеющий в своем составе несколько программных агентов, выполняющих конкретные функции.

Следует отметить, что деление агентов на атомарные и неатомарные является достаточно условным и зависит от уровня детализации при анализе структуры машины обработки знаний.

4. Семантический унифицированный язык описания вопросов

Язык описания вопросов (или просто язык вопросов) предназначен для формулирования вопросов системе сторонними субъектами, каковыми могут быть как пользователи системы, так и сторонние интеллектуальные системы. Также и в рамках самой системы агенты обработки знаний могут формулировать вопросы, предназначенные для других подобных агентов.

Каждый вопрос в памяти системы описывается некоторой вопросной конструкцией, содержащей сам узел вопроса и дополнительную информацию, необходимую для поиска ответа на заданный вопрос – тип вопроса, аргументы и т.д.

Ниже приводятся описания ключевых узлов языка описания вопросов в рамках технологии OSTIS.

- Ключевой узел **вопрос** - знак множества вопросов произвольного вида, без уточнения конкретной семантики вопроса.

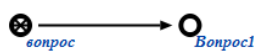


Рисунок 2 – Вопрос

- Семейство знаков множеств частных вопросов.

Примерами знаков множеств частных вопросов могут служить **запрос значения величины**, **запрос истинности** и т.д.

Принадлежность какому-либо классу частных вопросов уточняет семантику данного конкретного вопроса. Классы вопросов можно разделить на предметно-зависимые (запрос яркости небесного тела) и предметно-независимые (запрос значения величины). Однако такое деление достаточно условно, потому как, например, запрос температуры объекта или запрос скорости объекта могут быть использованы далеко не только в системе по физике.

Любой класс частных вопросов является подмножеством множества вопросов.

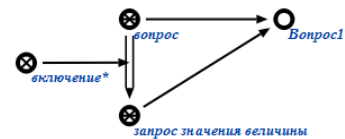


Рисунок 3 – Частный вопрос

- Аргументы вопроса.

Аргументами вопроса могут быть любые элементы базы знаний системы. В зависимости от конкретного класса вопроса их количество может варьироваться от 0 до десятка. Аргумент вопроса с теоретико-множественной точки зрения является его элементом, т.е. любой вопрос представляет собой множество аргументов (иногда – пустое множество).

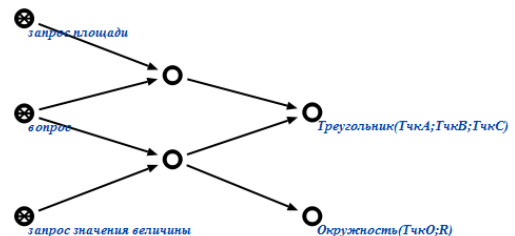


Рисунок 4 – Аргументы вопроса

При необходимости роль каждого аргумента может уточняться при помощи соответствующих ролевых отношений:

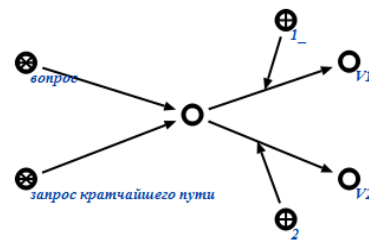


Рисунок 5 – Роль аргументов вопроса

- Ключевой узел **автор***.

Является знаком отношения, связывающего вопрос автора данного вопроса (т.е. субъекта, сгенерировавшего в памяти соответствующую вопросную конструкцию).

Указание автора является необходимым в ряде случаев, например, когда вопрос задан пользователем через интерфейс, и ответ также должен быть выдан в нужное окно пользовательского интерфейса.

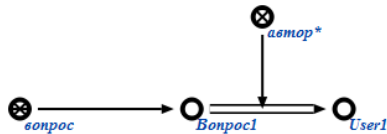


Рисунок 6 – Автор вопроса

- Ключевой узел **ответ***.

Является знаком отношения, связывающего конкретный вопрос и некоторую конструкцию, являющуюся ответом на данный вопрос.

Вид этой конструкции определяется конкретным классом вопроса. Наличие ответа на вопрос (конечно, при условии возможности его получения) является необходимым независимо от автора, класса вопроса и прочих параметров.

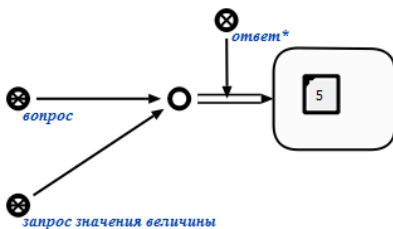


Рисунок 7 – Ответ на вопрос

- Ключевой узел **решение***.

Является знаком отношения, связывающего конкретный вопрос решением. Решение представляет собой набор связей, каждая из которых содержит информацию о том, какое утверждение было использовано на данном шаге решения для какого объекта, и какая конструкция получена в итоге.

На указанных связках задано отношение строго порядка, обеспечивающее возможность определения последовательности шагов решения конкретной задачи.

В некоторых случаях решение может отсутствовать даже в случае успешного ответа на вопрос, к примеру, в том случае, если решение задачи ограничено простым поиском.

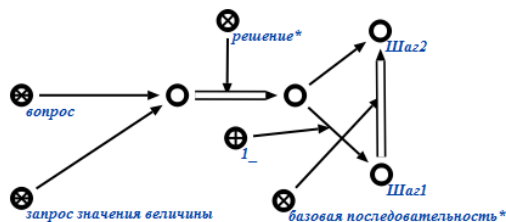


Рисунок 8 – Решение задачи

- Ключевой узел **иницированный вопрос**.

Является знаком множества инициированных вопросов. Тот факт, что вопрос инициирован, свидетельствует о том, что вопросная конструкция полностью сформирована и машина обработки знаний может приступать к поиску ответа на поставленный вопрос. Таким образом, генерация дуги принадлежности вопроса множеству инициированных вопросов должна являться завершающим шагом построения любой вопросной конструкции.

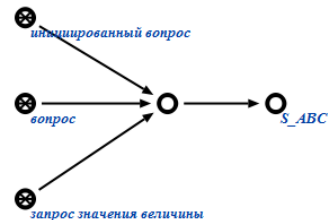


Рисунок 9 – Иницированный вопрос

- Ролевое отношение **присутствует ответ'**.

Используется для указания того факта, что системе удалось найти ответ на поставленный вопрос. Роль указывается для вопроса в рамках некоторого класса частных вопросов.

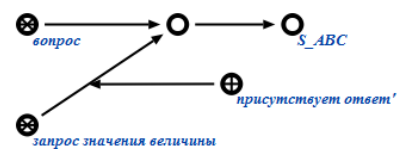


Рисунок 10 – Присутствие ответа на вопрос

Предполагается, что факт присутствия ответа указывается после того, как ответная конструкция полностью сформирована и сгенерирована связка отношения **ответ***.

- Ролевое отношение **отсутствует ответ'**.

Используется для указания того факта, что ответ на поставленный вопрос на настоящий момент найден быть не может. Роль указывается для вопроса в рамках некоторого класса частных вопросов.

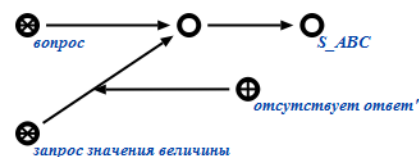


Рисунок 11 – Отсутствие ответа на вопрос

Для дополнительной синхронизации между агентами машины обработки знаний факт отсутствия ответа может уточняться более частными ролевыми отношениями, например, **ответ отсутствует в явном виде'**.

5. Семантический унифицированный язык спецификации агентов машин обработки знаний

Семантическая спецификация агента машины обработки знаний позволяет менеджеру агентов определить необходимость запуска того или иного агента для решения конкретной задачи (подзадачи), а также установить порядок запуска агентов для решения одной и той же задачи для обеспечения большей эффективности.

Ниже приводятся описания ключевых узлов языка спецификации агентов машин обработки знаний в рамках технологии OSTIS.

- Ключевой узел **стратегия решения задач**.

Является знаком множества стратегий решения задач. Используется менеджером стратегий для доступа к возможным стратегиям решения задач.

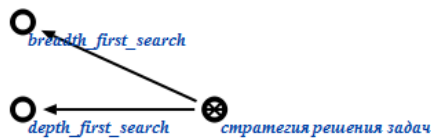


Рисунок 12 – Стратегии решения задач

- Ключевой узел **операция логического вывода**.

Является знаком множества операций логического вывода. Используется менеджером операций логического вывода к возможным операциям логического вывода.

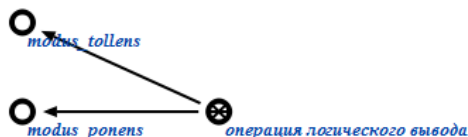


Рисунок 13 – Операции логического вывода

- Ключевой узел **условие инициирования***.

Является знаком отношения, связывающего конкретную sc-операцию и условие ее инициирования, т.е. шаблон конструкции наличие которой в памяти системы является критерием активации данной sc-операции. Условие инициирования используется соответствующим менеджером операций для определения целесообразности запуска той или иной операции в данный момент.

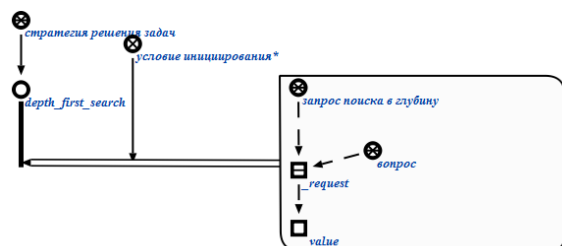


Рисунок 14 – Условие инициирования

- Ключевой узел **результат работы***.

Является знаком отношения, связывающего конкретную sc-операцию и множество результатов ее выполнения. Указанное множество, как правило, представляет собой связку отношения **строгая дизъюнкция***, поскольку несколько результатов работы не могут появиться в результате одного запуска sc-операции.

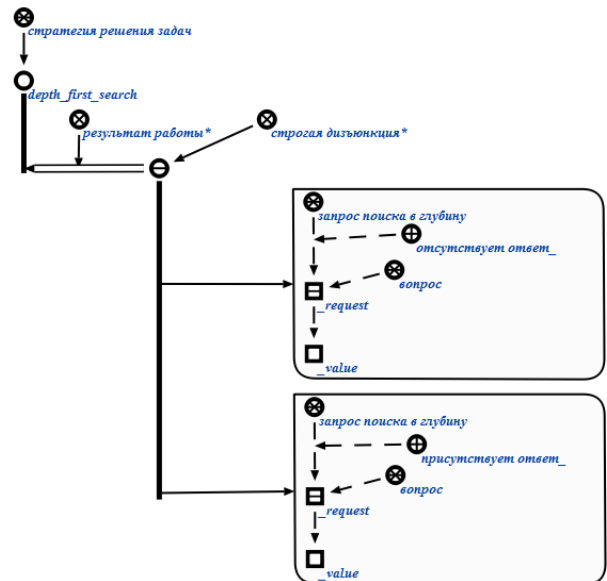


Рисунок 15 – Результат работы

Результат работы представляет собой шаблон, изоморфный конструкции, которая может быть сгенерирована агентом машины обработки знаний в семантической окрестности вопроса в процессе работы.

Результаты работы также могут быть использованы менеджерами операций для установления целесообразности запуска той или иной sc-операции в данный момент.

- Ключевой узел **приоритет запуска***.

Является знаком отношения связывающего различные sc-операции с целью указания того факта, какая из операций является более приоритетной для запуска.

Отношение, как вытекает из определения, является транзитивным, антисимметричным и антирефлексивным, т.е. есть отношением строго порядка [Кузнецов, 2009].

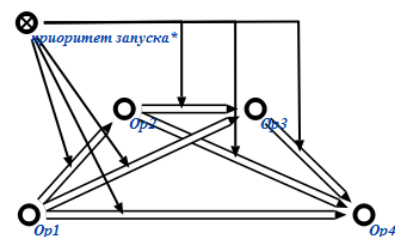


Рисунок 16 – Приоритет запуска агентов

Как видно, множество агентов, связанных данным отношением, образуют стандартный граф предпочтений [Поспелов, 1994], в котором узлам

соответствуют sc-операции, а дуги устанавливают приоритет запуска.

6. Классификация методов решения задач в интеллектуальных системах

Проблема автоматического решения задач достаточно давно рассматривается в работах по искусственному интеллекту. Приведем краткую классификацию существующих методов решения задач, рассмотренных в литературе:

- Классический дедуктивный вывод.

Классический дедуктивный вывод является наиболее популярным при построении автоматических решателей задач, так как всегда дает достоверный результат. Дедуктивный вывод включает в себя прямой и обратный и логический вывод (принцип резолюции, процедуру Эрбрана и др.) [Вагин и др., 2008], все виды силлогизмов [Малыхина, 2002] и т.д. Основной проблемой дедуктивного вывода является невозможность его использования в ряде случаев, когда отсутствуют достоверные правила вывода.

- Индуктивный вывод.

Индуктивный вывод предоставляет возможность в процессе решения использовать различные предположения, что делает его удобным для использования в слабоформализованных и трудноформализуемых предметных областях, например при построении систем медицинской диагностики. Подробно принципы индуктивного вывода рассмотрены в [Кулик, 2001], [Пойа, 1975].

- Абдуктивный вывод.

Под абдуктивным выводом в искусственном интеллекте, как правило, понимается вывод наилучшего абдуктивного объяснения, т.е. объяснения некоторого события, ставшего неожиданным для системы. Причем «наилучшим» считается такое объяснение, которое удовлетворяет специальным критериям, определяемым в зависимости от решаемой задачи и используемой формализации. Абдуктивный вывод подробно рассматривается в [Вагин и др., 2008].

- Нечеткие логики.

Теория нечетких множеств и, соответственно, нечетких логик, также применяется в системах, связанных с трудноформализуемыми предметными областями. Подробнее теория нечетких логик рассматривается в [Поспелов, 1989], [Батыршин, 2001], [Деменков, 2001] и других изданиях.

- Логика умолчаний.

Логика умолчаний применяется, в том числе, для того, чтобы оптимизировать процесс рассуждений, дополняя процесс достоверного вывода вероятностными предположениями в тех случаях, когда вероятность ошибки крайне мала. Подробнее

логика умолчаний рассмотрена в статье [RRIAI, 2012].

- Темпоральная логика.

Применение темпоральной логики является очень актуальным для нестатичных предметных областей, в которых истинность того или иного утверждения меняется со временем, что существенно влияет на ход решения какой-либо задачи. Следует отметить, что используемый в данной работе язык представления знаний предоставляет все необходимые возможности для описания таких динамических предметных областей. Более подробно темпоральная логика рассмотрена в работе [Еремеев, 1997]

В заключение данного раздела, следует отметить один из важнейших принципов данной работы. Данная работа не ставит своей целью разработку нового метода решения задач, нового класса логик или отрицание существующих достижений в данной области. Целью работы является разработка технологии, позволяющей интегрировать *любые модели решения задач и принципы логического вывода* для решения задач в интеллектуальных системах на основе общей формальной модели. Для того, чтобы использовать какую-либо новую или существующую модель, необходимо привести ее предлагаемому в данной работе формализму, что позволит интегрировать и синхронизировать ее с уже имеющимися в соответствующей библиотеке совместимых компонентов.

7. Библиотека ip-компонентов машин обработки знаний

Центральным элементом всей технологии OSTIS и, соответственно, любой более частной технологии является библиотека совместимых ip-компонентов.

Рассмотрим подробнее структуру этой библиотеки, от общего к частному.

- Библиотека готовых машин обработки знаний

В данную библиотеку целиком попадают самодостаточные машины обработки знаний, подходящие для какой-либо предметной области. Примерами могут служить:

- Машина обработки знаний для статичных предметных областей (геометрия, алгебра)
- Машина обработки знаний для динамичных предметных областей (физика, химия, многие гуманитарные области)
- Машина обработки знаний для систем классификации на основе признаков (медицинская диагностика, биология, другие классификационные области)
- И другие

- Библиотека готовых подсистем машин обработки знаний

В данную библиотеку попадают взаимосвязанные коллективы агентов,

соответствующие одному уровню в структуре машины обработки знаний, описанной выше.

Таковыми, например, являются:

- Поисковая машина, адаптированная под какой-либо класс предметных областей.
- Совокупность стратегий решения, адаптированных под какую-либо предметную область.
- Машина дедуктивного логического вывода
- Машина индуктивного логического вывода
- Машина логического вывода на основе темпоральной логики
- И другие

- Библиотека агентов машин обработки знаний

В данную библиотеку попадают все возможные реализации агентов, предназначенных для обработки знаний в интеллектуальных системах.

Данная библиотека может быть декомпозирована различными способами. Рассмотрим некоторые из них.

По функциональному назначению агентов:

- Библиотека агентов информационного поиска
- Библиотека агентов, реализующих стратегии решения задач
- Библиотека агентов логического вывода
- Библиотека агентов сборки мусора
- Библиотека агентов верификации знаний и устранения противоречий

По языку реализации агентов:

- Библиотека агентов, реализованных на внутреннем языке SCP
- Библиотека агентов, реализованных на внешнем языке программирования (может быть декомпозирована далее в зависимости от конкретных языков реализации).

По степени универсальности агентов (декомпозиция является достаточно условной, т.к. зависит от критериев универсальности):

- Библиотека агентов, ориентированных на решение частной задачи (поиск площади объекта)
- Библиотека агентов, ориентированных на решение некоторого класса задач (применение имплицитного утверждения, применение утверждения об эквивалентности)

- Библиотека программ и процедур машин обработки знаний

В данную библиотеку попадают все возможные программы и процедуры, использованные для реализации агентов обработки знаний.

Так же как и библиотека агентов, данная библиотека может быть декомпозирована различными способами по схожим критериям. Рассмотрим некоторые из них.

По функциональному назначению программ:

- Поисковые процедуры
- Процедуры генерации заданных конструкций
- Процедуры сравнения заданных конструкций
- Процедуры удаления заданных фрагментов конструкций
- Базовые теоретико-множественные процедуры (объединение, пересечение и т.д.)
- Системные процедуры, предназначенные, например, для предварительной обработки знаний

Другие критерии классификации процедур и программ полностью аналогичны рассмотренным в библиотеке агентов, поэтому подробное их описание приводить здесь не будем.

8. Автоматические средства поддержки проектирования машин обработки знаний

Рассмотрим ряд средств, обеспечивающих дополнительные возможности при проектировании машин обработки знаний на основе библиотек, а также собственно компонентов, входящих в состав каждой из библиотек.

- Среда программирования для языка SCP.

Данная среда программирования должна обладать всеми стандартными возможностями, предоставляемыми средами разработки для существующих языков программирования. Среди таких возможностей можно отметить

- Отладчик, поясняющий ошибки времени выполнения в виде, понятном пользователю, и позволяющий осуществлять пошаговую отладку программы с просмотром значений переменных и т.п.
- Редактор исходных текстов программ, имеющий функции автодополнения, подсветки синтаксиса, выявления синтаксических ошибок, поиска необъявленных переменных и т.д.
- Транслятор реального времени, позволяющий протранслировать составленную программу в память системы прямо в процессе ее работы для последующего тестирования и использования.

- Профайлер графодинамической памяти.

Задача профайлера состоит в подсчете объема памяти, используемого тем или иным агентом или группой агентов. Объем памяти оценивается соответственно на основе количества созданных узлов и дуг, с возможностью уточнения типа учитываемых элементов. Такой профайлер позволит отслеживать и предупреждать накопление информационного мусора, ложные срабатывания агентов обработки знаний и т.д.

- Визуализатор графодинамической памяти.

Задача визуализатора состоит в отображении в реальном времени средствами пользовательского интерфейса некоторого фрагмента памяти (например, семантической окрестности некоторого узла). Это позволяет разработчику лучше понять

процессы, происходящие в памяти, оценить правильность работы спроектированных программ и выяснить, действительно ли все происходит так, как он планировал. Использование визуализатора наиболее целесообразно одновременно с отладчиком в режиме пошаговой отладки, т.к. разработчик сможет визуально отследить реальный результат каждого шага составленной программы.

- Среда проектирования коллективов агентов над общей памятью.

В мире существуют средства, позволяющие осуществлять проектирование многоагентных систем. Наиболее значимые представители рассмотрены в статье А.В. Нарушева и В.Ф. Хорошевского [Нарушев, Хорошевский, 2000]. Однако все указанные средства ориентированы на непосредственное взаимодействие агент-агент, в то время как в рамках рассматриваемого подхода к проектированию машин обработки знаний взаимодействие агентов осуществляется через общую память.

Рассматриваемая среда проектирования коллективов агентов позволяет разработчику отслеживать срабатывания агентов при иницировании вопросов в памяти системы, минимизировать количество ложных или нецелесообразных срабатываний, выбрать наиболее рациональный порядок активации агентов в каждом случае.

Таким образом, речь идет о создании интегрированной среды разработки машин обработки знаний на основе мультиагентного подхода.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [OSTIS, 2012] Проект OSTIS [Электронный ресурс]. Минск, 2012. – Режим доступа: <http://ostis.net/>. – Дата доступа: 30.11.2012.
- [RRIAI, 2012] Искусственный интеллект. Системы и модели [Электронный ресурс]. Минск, 2012. – Режим доступа: <http://www.rriai.org.ru/logika-kosvennogo-opisaniya-i-logika-umolchaniya.html>. – Дата доступа: 30.11.2012.
- [Батыршин, 2001] Батыршин И.З. Основные операции нечеткой логики и их обобщения / И.З. Батыршин; – Казань : Отечество, 2001.
- [Вагин и др., 2008] Вагин В.Н. Достоверный и правдоподобный вывод в интеллектуальных системах / Вагин В.Н. [и др.]; – М. : ФИЗМАТЛИТ, 2008.
- [Владимиров и др., 2010] Владимиров А.Н., Варламов О.О., Носов А.В., Поталова Т.С. Программный комплекс “УДАВ”: практическая реализация активного обучаемого логического ввода с линейной вычислительной сложностью на основе миварной сети правил //Труды научно-исследовательского института радио. - 2010.- №.1. С. 108-116.
- [Гаврилова и др., 2001] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. Учебник / Гаврилова Т.А.. [и др.]; – СПб. : Изд-во «Питер», 2001.
- [Голенков и др., 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах / Голенков В.В. [и др.]; под ред. В.В. Голенкова – Минск, 2001.
- [Голенков и др., 2001] Программирование в ассоциативных машинах / Голенков В. В. [и др.]; под ред. В. В. Голенкова – Минск, 2001.
- [Деменков, 2001] Деменков Н.П. Нечеткое управление в технических системах / Н.П. Деменков; – М : Изд. им. Баумана, 2005.

[Еремеев, 1997] Еремеев А.П. Построение решающих функций на базе тернарной логики в системах принятия решений в условиях неопределенности // А.П. Еремеев Известия академии наук. Теория и системы управления, 1997. №5.

[Ефимов, 1982] Ефимов, Е. И. Решатели интеллектуальных задач / Е. И. Ефимов; - М. : Наука, 1982.

[Кахро и др., 1988] Инструментальная система программирования ЕС ЭВМ (ПРИЗ) / М.В. Кахро, А.П. Калья, Э.Х. Тыугу;– М., Изд-во «Финансы и статистика», 1988.

[Кузнецов, 2009] Кузнецов О.П. Дискретная математика для инженера. Изд. 6, стереотипное. / О.П. Кузнецов; - СПб, Лань, 2009.

[Кулик, 2001] Кулик, Б. А. Логика естественных рассуждений / Б. А. Кулик; - СПб.: Изд-во «Невский диалект», 2001.

[Малыхина, 2002] Малыхина Г.И. Логика / Г.И. Малыхина; – Мн. : Высшая школа, 2002.

[Нарушев, Хорошевский, 2000] Нарушев Е.С., Хорошевский В.Ф. AgSDK: Инструментарий разработки мультиагентных систем // Труды 7-ой Национальной конференции по искусственному интеллекту с международным участием, Переславль Залесский, 24-27 октября 2000 – Москва: ИФМЛ, том 2.

[Пойа, 1975] Пойа Д. Математика и правдоподобные рассуждения / Пойа Д.; – М. :Изд-во «НАУКА», 1975.

[Поспелов, 1989] Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов / Д.А.Поспелов; – М. :Изд-во «Радио и связь», 1989.

[Поспелов, 1994] Поспелов Д.А. Информатика. Энциклопедический словарь. / Д.А.Поспелов; – М. : «Просвещение», 1994.

[Рассел, Норвиг 2006] Рассел С., Норвиг П. Искусственный интеллект. Современный подход / Рассел С., Норвиг П. ; - М. : Вильямс, 2006.

[Тарасов, 2002] Тарасов, В.Б. От многоагентных систем к интеллектуальным организациям / В.Б. Тарасов; – М. :Изд-во УРСС, 2002.

MODELS AND MEANS OF KNOWLEDGE PROCESSING MACHINES COMPONENT DESIGN ON BASIS OF SEMANTIC NETWORKS

Shunkevich D.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

shu.dv@tut.by

This work is devoted to considering of problems of existing methods, means and technologies of knowledge processing machines design. The article describes the problem of lack of means, which allow relatively inexperienced developer to design knowledge processing machine for applied intelligent systems for different purposes. Next part of the paper considers the technology which was designed to solve given problem in the way of integration different problem situations solution methods and ways on the common formal basis.

Keywords: intelligent systems; knowledge processing machine; problem situations solver; component design; multiagent systems.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

АВТОМАТИЗАЦИЯ ПОИСКА ТРАНСФОРМАЦИЙ СХЕМ ОЦЕНКИ УВЕРЕННОСТИ

Моросанова Н.А.

*Московский государственный университет имени М.В. Ломоносова, факультет ВМК
г. Москва, Россия*

nmosanova@gmail.com

При совместном использовании систем, использующих различные схемы оценки уверенности, возникает задача равномерного накопления и обмена информацией между схемами. Эта задача может быть решена с помощью трансформаций, сохраняющих операцию комбинирования. В работе рассматриваются возможности автоматизации поиска таких трансформаций. Приводятся результаты работы предложенного алгоритма поиска на примерах трансформаций схем, основанных на коэффициентах уверенности и байесовском подходе, а также для функций комбинирования, полученных аксиоматическим методом.

Ключевые слова: изоморфизм, операция комбинирования, оценка уверенности.

ВВЕДЕНИЕ

Схема оценки уверенности есть некоторый способ описания информации, определяемый при создании экспертной системы. Эта схема используется для составления базы знаний и организации логического вывода. Существование трансформации, устанавливающей соответствие между двумя схемами, определяется свойствами операций в этих схемах.

Выбор конкретной схемы оценки уверенности является подзадачей, возникающей при разработке экспертных систем. При этом существует возможность выбирать сразу несколько различных схем для различных категорий пользователей, либо различных задач, решаемых с помощью этой системы. В таком случае требуется некоторый способ преобразования данных, представленных с помощью различных схем оценки уверенности.

1. Подходы к трансформации оценок уверенности

Задача трансформации оценки уверенности возникает при наличии применяющих различные схемы систем, которые необходимо соединить в одну неоднородную систему [Luo et al., 2001]. Задача состоит в обеспечении обмена информацией, представленной в терминах различных схем. Можно выделить два основных подхода к решению этой задачи:

1. построение некоторого обобщения всех или нескольких используемых схем;
2. построение преобразований между парами схем.

К первому подходу можно отнести построение различных обобщающих схем на основе нечетких логик [Аршинский, 2007, Аверкин и др., 2000, Hunter et al., 2006], теории Демпстера-Шафера [Hajek et al., 2008].

Ко второму подходу относятся установленные соответствия между распространенными схемами оценки уверенности [Heckerman, 1990, Groszof, 1986, Daniel, 2004].

Выбор подхода зависит от свойств конкретной задачи, например, количества различных схем оценки уверенности. Первый подход позволяет получать многократно используемые результаты, второй подход направлен на решение конкретного частного случая. Тем не менее, рассмотрение таких частных случаев может дать дополнительную информацию для построения обобщающей схемы, либо ее выбора из уже имеющихся обобщений.

2. Возможности автоматизации трансформаций

В случае задачи трансформации систем, применяющих различные схемы оценки уверенности, системы работают отдельно друг от друга, но могут обмениваться информацией. Для этого необходим изоморфизм операций комбинирования.

Будем рассматривать задачу поиска трансформации для пары схем, сохраняющей результат операции комбинирования.

Будем считать, что для каждой схемы определены:

- множество значений, которые описывают степень уверенности в факте;
- «особые» значения этого множества: «истина», «ложь», «отсутствие информации»;
- допустимые операции над этим множеством;
- отношение порядка на множестве значений.

Трансформация есть отображение множества значений одной схемы во множество значений другой схемы. Будем обозначать его, как $h(x)$. Рассмотрим основные свойства, которые могут быть у $h(x)$:

- сохранение соответствия «особых» значений двух множеств;
- существование обратного преобразования $h^{-1}(x)$;
- установление соответствия между операциями комбинирования;
- монотонность: если $x_1 > x_2$, то $h(x_1) > h(x_2)$.

Обозначим множества значений и операции комбинирования двух схем, как X , Y и cmb_1 , cmb_2 , соответственно. Предположим, что обе операции дизъюнктивны, то есть

$$\begin{aligned} \text{cmb}_1(x_1, x_2) &> \max(x_1, x_2), \\ \text{cmb}_2(y_1, y_2) &> \max(y_1, y_2), \end{aligned} \quad (1)$$

а также что X и Y замкнуты и в них выделены особые значения x_{\min}, x_{\max} , и y_{\min}, y_{\max} , соответствующие друг другу:

$$\begin{aligned} h(x_{\min}) &= y_{\min}, \\ h(x_{\max}) &= y_{\max}. \end{aligned} \quad (2)$$

Будем искать монотонную трансформацию $h(x)$, такую, что

$$h(\text{cmb}_1(x_1, x_2)) = \text{cmb}_2(h(x_1), h(x_2)) \quad (3)$$

в виде некоторого численного приближения $h^*(x)$.

3. Алгоритм поиска

Пусть на X и Y заданы две однородные сетки $\{x_i\}$ и $\{y_j\}$ размера n . Будем считать, что в переменных $listx$ и $listy$ накапливаются элементы множеств X и Y , соответствующие друг другу. Изначально $listx = (x_{\min}, x_{\max})$ и $listy = (y_{\min}, y_{\max})$, а очередь q_1 содержит все элементы $\{x_i\}$. Приведем

алгоритм получения $h^*(x)$ в указанных условиях.

3.1. Базовый алгоритм

1) Если очередь q_1 не пуста, то перейти на п.2, иначе на п.4

2) Для очередного x_i выполнить поиск $h^*(x_i)$:

3) Для каждого y_j :

a. Фиксировать y_j как кандидата;

b. Построить последовательность точек $\{z_k\}$:

$$\begin{aligned} z_1 &= x_i, \quad z_k = \text{cmb}_1(z_{k-1}, x_i), \quad k \geq 2, \quad k \in \square, \\ \text{пока } |z_k - z_{k-1}| &> \varepsilon_1; \end{aligned} \quad (4)$$

c. Построить соответствующую последовательность точек $\{u_k\}$:

$$\begin{aligned} u_1 &= y_j, \quad u_k = \text{cmb}_2(u_{k-1}, y_j), \quad k \geq 2, \quad k \in \square, \\ \text{пока } |u_k - u_{k-1}| &> \varepsilon_2; \end{aligned} \quad (5)$$

d. Добавить $\{z_k\}$ к $listx$ и $\{u_k\}$ к $listy$, если при этом не нарушается их «монотонность»:

$$\begin{aligned} listx(pos_1) &> listx(pos_2) \\ \Rightarrow listy(pos_1) &> listy(pos_2), \end{aligned} \quad (6)$$

в противном случае вернуться к п. 2;

4) По полученным точкам $h^*(x_i)$ построить $h^*(x)$ (например, с помощью полиномиальной регрессии);

5) Если полученная $h^*(x)$ удовлетворяет требованиям монотонности и неравенства

$$\frac{\sum_{i=1}^n |h^*(\text{cmb}_1(x_i, x_i)) - \text{cmb}_2(h^*(x_i), h^*(x_i))|}{n} < \varepsilon_3, \quad (7)$$

где ε_3 – заранее заданный порог, то $h^*(x)$ считать приближением $h(x)$.

Важно отметить, что если $h(x)$ не единственна, то может быть найдено некоторое из всех подходящих отображений. Также приближение $h^*(x)$ может быть найдено в тех случаях, когда точное решение не существует.

3.2. Модификации алгоритма

Так как алгоритм переборный, то для его эффективной работы требуются эвристики, позволяющие на ранней стадии отбрасывать неперспективные варианты. Одна из них, погрешность (7), может использоваться не только при окончательном решении, но и на промежуточных этапах, когда длина $listx$ больше некоторого числа, например, половины размера сетки.

Другое требование к $h(x)$ – сохранение разницы пороговых значений, может интерпретироваться как достаточный наклон $h(x)$, то есть $h(x)$ не должна быть константой, в то время как именно константа лучше всего минимизирует (7). Тогда можно ввести значение

$$\frac{\sum_{i=2}^m (listy(i) = listy(i-1))}{m} = \varepsilon_4, \quad (8)$$

где m – длина $listx$, а $listy$ упорядочен по возрастанию соответствующих элементов $listx$. Это значение, в свою очередь, не должно превышать заранее заданного порога, например 0,5 или 0,3. Для константы это значение равно 1.

Поскольку, как правило, подходящие трансформации составляют целое семейство, требуется дополнительная информация для получения конкретной $h(x)$. Могут быть использованы значения «отсутствие информации», которые часто являются нейтральными элементами для комбинирования:

$$cmb(x, e) = x \quad \forall x. \quad (9)$$

Тогда трансформация $h(x)$ должна устанавливать соответствие таких значений. По имеющейся операции комбинирования cmb значение e можно найти выбором значения сетки x_j , минимизирующего значение

$$\sum_{i=1}^n |cmb(x_i, x_j) - x_i|. \quad (10)$$

3.3. Примеры работы алгоритма

Рассмотрим примеры работы алгоритма для описанного в [Luo et al., 2001] изоморфизма между схемами Байеса и Шортлиффа.

В качестве способа приближения (п. 4 в описании алгоритма) используется квадратичная регрессия. Перед использованием алгоритма производится поиск нейтральных элементов для операций комбинирования с помощью (9)-(10). Алгоритм применяется отдельно для интервалов значений больше и меньше найденного нейтрального элемента.

В таблице приведены результаты для двух вариантов отображения [Luo et al., 2001], так как в [Luo et al., 2001] приводится семейство отображений, зависящее от параметра $p(H)$ – априорной вероятности гипотезы. Для трансформации $p(H)$ соответствует $CF = 0$.

Таблица 1 – Результаты поиска $h^*(x)$

Вариант	$p(H) = 0,5$	$p(H) = 0,33$
---------	--------------	---------------

$h(x)$	$x < 0$	$\frac{x+1}{x+2}$	$\frac{x+1}{x+3}$
	$x \geq 0$	$\frac{1}{2-x}$	$\frac{1}{3-2x}$
$h^*(x)$	$x < 0$	$0,5 + 0,35x - 0,16x^2$	$0,33 + 0,23x - 0,1x^2$
	$x \geq 0$	$0,5 + 0,25x + 0,25x^2$	$0,33 + 0,23x + 0,44x^2$
ε_1		0,1	0,1
ε_2		0,1	0,1
ε_3		0,007	0,007
ε_4		0,3	0,3

Рассмотрим пример применения алгоритма в том случае, когда точного решения не существует. Для этого пусть

$$X = [0, 1], \quad Y = [0, 1],$$

$$cmb_1(x, y) = x + y - xy, \quad (11)$$

$$cmb_2(x, y) = x + y - xy(x + y - xy).$$

Здесь cmb_1 – положительная часть функции комбинирования Шортлиффа, а cmb_2 – функция комбинирования на $[0, 1]$, полученная аксиоматическим методом [Johnson et al., 1989].

Точное решение $h(x)$ уравнения $h(cmb_1(x_1, x_2)) = cmb_2(h(x_1), h(x_2))$ не существует, поскольку cmb_1 ассоциативна, а cmb_2 – нет.

При $\varepsilon_3 = 0,008$ получено решение

$$h^*(x) = 0,45x + 0,55x^2. \quad (12)$$

На рисунке 1 приведено графическое изображение функции

$$cmb_2(h^*(x), h^*(y)) - h^*(cmb_1(x, y)). \quad (13)$$

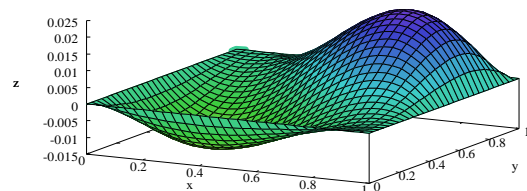


Рисунок 1 – Трансформация $h^*(x) = 0,45x + 0,55x^2$.

Для сравнения на рисунке 2 приведена исходная разность значений между cmb_1 и cmb_2 .

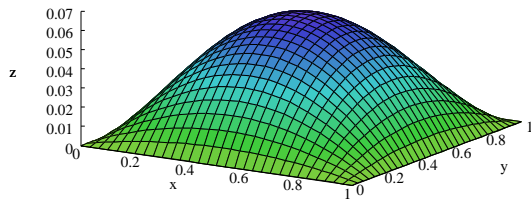


Рисунок 2 – Исходная разность значений

Видно, что с помощью применения алгоритма удается получить трансформацию, переводящую cmb_1 в приближение cmb_2 . При этом максимальная разность их значений уменьшается на порядок.

ЗАКЛЮЧЕНИЕ

В тех случаях, когда операции не дизъюнктивны, либо имеют различные свойства на подмножествах своего множества определения, задача усложняется: поиск трансформации выполняется отдельно на каждом таком подмножестве. Таким образом, требуется решить дополнительную задачу определения интервалов значений $\text{cmb}(x, y)$, на которых ее поведение:

- дизъюнктивно: $\text{cmb}(x, y) > \max(x, y)$;
- конъюнктивно: $\text{cmb}(x, y) < \min(x, y)$
- компромиссно:

$$\text{cmb}(x, y) \leq \max(x, y),$$

$$\text{cmb}(x, y) \geq \min(x, y).$$

Также интересной задачей может стать обобщение алгоритма для работы с оценками уверенности, которые описываются не одним числом, а набором чисел.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Daniel, 2004] Daniel, M. Algebraic structures related to the combination of belief functions / Daniel, M. // *Scientiae Mathematicae Japonicae*. – 2004 – № 60(2), p. 245-256.
- [Groszof, 1986] Groszof, B. Evidential confirmation as transformed probability / Groszof, B. // *Uncertainty in Artificial Intelligence*. – 1986 – p.153-166.
- [Hajek et al., 2008] Hajek, P. and Valdes, J.J. An analysis of MYCIN-like expert systems/ Hajek, P. and Valdes, J.J. // *Mathware & soft computing*. – 2008 – № 1(1), p. 45-68.
- [Heckerman, 1990] D. Heckerman. Probabilistic interpretations for MYCIN's certainty factors. In *Readings in uncertain reasoning*, Glenn Shafer and Judea Pearl (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA 298-312.
- [Hunter et al., 2006] Hunter A., Liu W. Fusion rules for merging uncertain information / Hunter A., Liu W. // *Information Fusion*. – 2006 – № 7(1), p. 97-134.
- [Johnson et al., 1989] Johnson N. L., Kotz S. Axiomatic approaches to formulas for combining likelihoods or evidence / N. L., Kotz S. // *Journal of Statistical Computation and Simulation*. – 1989 – № 31(1).
- [Luo et al., 2001] Luo, X. and Zhang, C. and Leung, H. Information sharing between heterogeneous uncertain reasoning models in a multi-agent environment: a case study / Luo, X. and Zhang, C. and Leung, H. // *International journal of approximate reasoning*. – 2001 – № 27(1), p. 27-59.

[Аверкин и др., 2000] Аверкин А.Н., Костерев В.В. Тriaнгуляpные нормы в системах искусственного интеллекта / Аверкин А.Н., Костерев В.В. // *Известия академии наук. Теория и системы управления*. – 2000 – № 5, с. 116-128.

[Аршинский, 2007] Аршинский Л. В. Векторные логики: основания, концепции, модели: монография // Л.В. Аршинский. – Иркутск: ВСИ МВД России, 2007.

AUTOMATED SEARCH FOR UNCERTAIN REASONING MODELS TRANSFORMATIONS

Morosanova N.A.

Lomonosov Moscow State University, CMC department, Moscow

nmorosanova@gmail.com

A problem of information sharing between different uncertain reasoning models can occur in heterogeneous multi-agent systems. The solution for this problem may be obtained by the use of the uncertain reasoning models' transformations. This study explores the possibilities of an automated search for such transformations. The results of the algorithm proposed for the case with the combination operations isomorphism are presented and discussed.

INTRODUCTION

Uncertain reasoning model specifies the way to describe uncertainty, including the value set and the combination operation. For the information sharing between systems, using several different models the transformation should preserve the result of combination operations.

MAIN PART

We seek to find transformation $h(x)$ that preserves the result of combination. An algorithm receives two combination operations and their value sets. It is based on a recursive exhaustive search with backtracking and a number of heuristics to cut the search space. The results are given in Table 1 for the Shottliffe CF – Bayes transformation. Fig. 1 and 2 show the results of an algorithm for the case where only pseudo-solution exists.

CONCLUSION

The proposed algorithm proved to find the transformation for the MYCIN-like combination operations. The next step should consist in using several values instead of only one for the uncertainty representation. Also the intervals of value set where the operation is disjunctive, conjunctive or compromise should be determined in order to apply the search algorithm more effectively.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89: 519.179

ИНТЕГРИРОВАННЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ В ГИБРИДНЫХ СИСТЕМАХ

Комарцова Л.Г.

*Московский государственный технический университет им. Н.Э. Баумана (Калужский филиал)
г. Калуга, Россия*

lkomartsova@yandex.ru

В докладе рассматриваются проблемы создания гибридных интеллектуальных систем поддержки принятия решений, что позволит более эффективно выполнять задачи, связанные с распознаванием, предсказанием, планированием действий и т.д. в режиме реального времени. Предложены новые гибридные модели и алгоритмы обучения, что позволит повысить эффективность решения сложных практических задач.

Ключевые слова: интеллектуальные информационные технологии, системы искусственного интеллекта, гибридные системы, нейронные сети, системы нечеткой логики, экспертные системы, гибридные системы

Введение

В последние годы наблюдается рост числа успешных примеров использования гибридных интеллектуальных систем в различных прикладных областях, таких, как, медицинская диагностика, распознавание речи и естественных языков, создание мобильных роботов, мониторинг и контроль производственных процессов, финансовые приложения. Гибридные интеллектуальные системы (ГС), работающие на основе принципов объединения нескольких методов представления и обработки знаний, позволяют получать значительно лучшие результаты решения по сравнению с интеллектуальными системами (ИС), использующими единственный метод для тех же проблем [Kasabov, 2003], [Jang et al., 1997], [Medsker, 1998]. Однако проблемы создания и использования гибридов для конкретных приложений все еще не решены. Остается много вопросов, связанных с тем, на каком уровне проводить объединение разных интеллектуальных технологий, какие гибриды являются наиболее перспективными, как учесть динамику изменения среды функционирования ИС и т.д. Для решения этих и других проблем в статье рассматриваются особенности наиболее известных интеллектуальных технологий и возможности их интеграции в ГС.

В зависимости от поставленной научной или практической задачи, необходимо использовать различные технологии. Поэтому необходимо исследовать возможности различных ИТ для объединения технологий с целью компенсации

недостатков и усиления достоинств каждой из технологий при решении сложных задач.

1. Оценка возможностей информационных технологий для решения задач

Проведем оценку возможностей различных информационных технологий для решения определенного типа задач:

Представление знаний:

ЭС – правила;
НЛ – правила;
НС – модель (персептрон).

Вывод:

ЭС – точный;
НЛ – приближенный;
НС – приближенный.

Степень обучения:

ЭС – средняя;
НЛ – нет;
НС – хорошая.

Степень обобщения:

ЭС – слабая;
НЛ – очень хорошая;
НС – очень хорошая.

Взаимодействие с человеком:

ЭС – хорошее;
НЛ – хорошее;

НС – хорошее.

Объяснение:

ЭС – очень хорошее;
НЛ - очень хорошее;
НС – слабое.

Тестирование:

ЭС – очень хорошее;
НЛ - очень хорошее;
НС – среднее.

Адаптация:

ЭС – слабая;
НЛ - слабая;
НС – хорошая.

Анализируя представленные сравнительные данные, можно заметить, что, например, нейронные сети (НС) являются наилучшей технологией для решения задач обучения, адаптации и обобщения, экспертные системы (ЭС) и нечеткие системы (построенные на основе нечеткой логики – НЛ) обладают хорошими объяснительными возможностями и т.д. В соответствии с этим, в зависимости от решаемой задачи, нужно выбирать такую технологию, которая будет давать наилучший результат. В соответствии с этим гибридные технологии, объединяющие несколько информационных технологий при решении конкретных задач, представляются наиболее перспективными.

Под гибридной интеллектуальной системой (ГС) будем понимать систему, в которой для решения задачи используется более одного метода имитации интеллектуальной деятельности человека. Интеграция методов, с одной стороны, дает возможность использовать индивидуальную силу каждого из методов для решения специфических частей задачи, что позволит создать более эффективные модели представления и обработки знаний. С другой стороны, гибридный подход основывается на том, что только синергетическая комбинация интеллектуальных технологий может достичь полного спектра когнитивных и вычислительных возможностей, реализуемых в компьютерных моделях интеллектуальных систем.

Существуют, по крайней мере, две главные причины, по которым необходимо использовать именно гибридные системы [Kasabov, 2003]:

– некоторые требования для решения проблем искусственного интеллекта (ИИ) не могут быть принципиально выполнены на основе единственного подхода;

– для решения сложных проблем ИИ создаваемые модели также не могут быть реализованы с помощью одного какого-то метода.

По мере социального и технического развития общества появляются новые требования, которые приходится учитывать при решении сложных интеллектуальных задач. Естественно, что одного

метода для удовлетворения всем требованиям не существует. Выбор методов для обработки исходной информации зависит от особенностей решаемых задач, от числа количественных и качественных параметров, описывающих проблему, от уровня проработанности задачи. Поэтому необходимо определять условия применимости каждого из методов, а также алгоритмы, позволяющие адаптировать их к решению конкретных задач проблемной области.

Обзор информационных технологий, использующихся в современных ИС, показывает, что в статических и динамических проблемных средах эффективными являются определенные методы. Например, в некоторых случаях наилучшее решение может быть получено путем использования эволюционных методов, в частности на основе генетических алгоритмов, на всех этапах поиска рационального решения, что позволяет перейти от моделей представления и использования знаний с жесткими связями к моделям с динамически меняющейся структурой в зависимости от решаемой задачи.

2. Классификация гибридных систем

Цели интеграции отдельных технологий в ГС и основная терминология впервые были введены Bezdek J. [Bezdek, 1994]. Предложенная им модель интеграции отдельных технологий базируется на различных уровнях интеллектуальной активности компонентов, объединяемых в иерархическую систему. Главное отличие введенной модели от других ГС состоит в увеличении сложности решаемых задач при переходе от низшего уровня, соответствующего вычислительному интеллекту, к более высокому уровню ИИ, а затем к биологическому, моделирующему человеческий интеллект. Рассматриваемая модель показывает взаимосвязь уровней при обработке информации, при этом, в зависимости от имеющейся информации и решаемых задач, активизируется определенный уровень. Вычислительный интеллект определяет уровень обработки числовых данных, ИИ основан на процедурах обработки символов и данных, используя правила и нечисловые данные. Биологический интеллект обрабатывает сенсорные входы, и на основе ассоциативной памяти выполняет процедуру распознавания образов.

В модели Besdek J. вычислительный и искусственный интеллект являются строительными блоками для построения более сложного биологического интеллекта. Вычислительный интеллект имеет дело только с числовыми данными, обладает способностью распознавания и не использует знания в смысле ИИ, но при этом вычислительные нейроподобные системы являются моделями, которые появились из биологии. Эти компоненты могут быть нейронными сетями (НС) с прямым распространением сигналов,

самоорганизующимися сетями Кохонена и т.д., представляться с помощью генетических алгоритмов или эволюционных моделей. Системы, основанные на явных знаниях, или на прецедентах являются промежуточным уровнем; сюда же можно отнести когнитивные модели, имитирующие работу отдельных элементов мозга. Модели на основе нечеткой логики аккумулируют числовую и семантическую информацию и также являются компонентами символического уровня.

Создание гибридных моделей ГС в смысле Besdek связано с расширением возможностей низкоуровневого вычислительного интеллекта за счет высокоуровневого ИИ с целью реализации биологического интеллекта.

IRIS-модель (Integration of Reasoning, Informing and Serving). Эта гибридная модель была предложена Soucek В. [Soucek, 1991] и предназначалась для создания более эффективной интеллектуальной технологии для бизнеса с целью достижения основных показателей производства: темпы, качество, минимальные затраты. Модель базируется на системном подходе к созданию ИС и включает как инженерные методики, так и ряд интеллектуальных компонент, а именно, методы, используемые в научных дисциплинах, таких, как, биология, когнитивная психология, лингвистика, эпистемология, компьютерные науки. Важность создания таких гибридных моделей заключается в возможности обработки разнообразных данных и обобщении знаний. В настоящее время наиболее высоким уровнем, реализованным в этой гибридной модели, является уровень понимания (распознавания): интеграция знаний из различных источников; способность к объяснению; категоризация объектов в классы; реализация функций понимания; ассоциативность; обучение; обобщение и т.д.

Для реализации более высоких уровней требуется решение проблем организации диалога с системой на естественном языке, возможности обмена информацией любого типа (текста, графики) между уровнями, возможности представления и обработки нечеткой и неопределенной информации. Это наиболее перспективные направления исследования принципов построения ИС.

Классификация по L. Medsker. L. Medsker [Medsker, 1998] предложил пять типов гибридных моделей, классифицируемых по степени связанности отдельных модулей: автономные, трансформационные, слабо связанные, сильно связанные, полностью интегрированные.

Автономные модели состоят из независимых программных компонент, которые не взаимодействуют между собой при решении проблем ИИ. Использование таких моделей преследует несколько целей:

1) возможность сравнения способов решения задач определенного типа для выбора наилучшего;

2) использование моделей в параллельном режиме для сокращения времени получения результата;

3) использование одной технологии после окончания работы другой может подтвердить или опровергнуть полученный результат;

4) использование одной технологии для быстрого создания прототипа ИС, который затем будет развиваться на основе другой технологии; например, ИС может быть быстро обучена на конкретных данных, а более полное исследование проблемы будет впоследствии проведено с помощью ЭС.

Трансформационные модели имеют возможность трансформации друг в друга, в отличие от автономных, при решении одной задачи с целью получения наилучшего результата. Наиболее часто используются гибридные модели ЭС→ИС и ИС→ЭС.

Слабо связанные модели - это по существу первая реальная форма интеграции ИС. Приложение распределяется между различными интеллектуальными компонентами, которые взаимодействуют через файлы данных. Достоинства слабо связанных систем заключаются в простоте разработки и использовании в качестве программного обеспечения коммерчески доступных программ, снижающих время на программирование.

Категории слабо и сильно связанных моделей имеют значительное перекрытие. Сильно связанные системы могут быть представлены теми же компонентами, что и слабо связанные системы: пре- и постпроцессорами, сопроцессорами, но которые работают быстрее.

Полностью интегрированные модели имеют общие структуры данных и знания. Взаимодействие между различными компонентами осуществляется на уровне методов. Наиболее известным гибридом этого класса являются нейро-нечеткие системы [Jang et al., 1997], в которых, например, многослойная нейронная сеть моделирует нечеткий вывод или нечеткая нейронная сеть выполняет нечеткую кластеризацию. Преимущества полной интеграции заключаются в том, что ИС, построенные на такой основе, могут одновременно обладать такими возможностями, как, адаптивность, обобщение, обучаемость, устранение шума или ослабление его влияния на конечный результат, использование логической дедукции. Эти возможности не достижимы в ИС с единственной интеллектуальной технологией.

Классификация по W. Wermter. Предложенные схемы классификации и терминология описания гибридных систем L. Medsker и другими авторами связаны с анализом степени взаимодействия между интеллектуальными модулями и иногда перекрываются. S.Wermter [Wermter, 1997] поставил задачу создания схемы классификации, которая бы включила все или большую часть основных особенностей предложенных ранее

классификаций, а также учитывала бы тенденцию развития новых интеллектуальных технологий, которые еще не были включены в классификационные схемы. В этом смысле S. Wermter рассматривает ГС как непрерывно развивающуюся (эволюционную) систему. Предложенная им классификационная схема включает три основные группы.

Первая группа – это унифицированные гибридные системы, состоящие из специализированных нейросетевых компонент, выполняющих функции символьных систем. Унифицированные системы строятся по двухуровневому способу: нижний уровень состоит из многих НС, каждая из которых реализует одно из правил базы правил, а верхний уровень – это НС, которая из многих правил в реальном масштабе времени выбирает одно активное правило. Несмотря на эффективность работы таких систем, они имеют ограниченное применение вследствие сложности разработки.

Вторая группа - трансформационные модели, использующие два формата представления: в виде ЭС и НС, которые могут преобразовываться в процессе функционирования ГС. Наиболее интересные особенности таких систем – возможность вставки, извлечения и обновления знаний.

Третья группа – модульные системы, функционирующие подобно биологическим системам, встречающимся в природе. Как правило, такие системы включают подсистемы, ответственные за реализацию определенных функций. Модульные ГС содержат несколько НС и несколько баз правил, которые могут иметь различную степень интеграции, при этом в процессе функционирования структура модулей не меняется. Поэтому для работы такой системы целесообразно создавать нейросетевую базу знаний, которая будет содержать набор модулей (обученных НС) для решения определенных проблем. Главная особенность модульных ГС – возможность параллельной обработки информации для повышения скорости или надежности вычислений.

Классификация по Kasabov N. Дальнейшее направление исследований в области построения гибридных систем связывается с созданием эволюционных, постоянно развивающихся систем, работающих в режиме on-line и подстраивающихся под конкретную решаемую задачу. Для этого должны быть предложены новые модели машинного интеллекта, включающие различные типы НС, созданные по модульному и иерархическому принципу, новые нейро-нечеткие и адаптивные системы. Kasabov N. в [Kasabov, 2003] предложил проект многоуровневой постоянно развивающейся эволюционной ГС, на каждом уровне которой возможна подстройка к решаемой задаче.

Генетический уровень (определяет вид и числовые параметры активационной функции НС, величину порога (bias), способ агрегации входов, количество входов и т. д.)

Нейронный уровень (реализует определенную функцию в системе и определяет радиус рецептивной области взаимодействия с другими нейронами).

Ансамбль нейронов (характеризуется структурой, способом распространения сигналов по НС, количеством слоев и нейронов в каждом слое, алгоритмом обучения, функцией оценки и интерпретации результатов работы сети).

Уровень многомодульной иерархической структуры (определяет количество и тип нейросетевых модулей системы, механизмы взаимодействия между отдельными модулями и принципы обмена информацией с НЛ для объяснения результатов решений, способ организации блока принятия решений, связь с внешней средой).

Популяция эволюционных систем (обеспечивает их развитие и взаимодействие на основе использования генетических алгоритмов).

Заключение

Основная цель исследования принципов построения эволюционных ГС заключается в модификации известных методов и технологий ИИ для обработки разнотипной информации в реальном масштабе времени. Другая, не менее важная цель – исследование принципов функционирования мозга и генетики для создания новых вычислительных моделей ГС в практических приложениях.

Библиографический список

- [Kasabov, 2003] Kasabov N. Evolving connectionist systems. Springer-Verlag London Limited.-2003.
- [Jang et al., 1997] Jang J.-S.R., Sun C.-T., Mizutani E. Neuro-Fuzzy and Soft computing. –Prentice-Hall, Inc.-1997.
- [Medsker, 1998] Medsker L. Hybrid intelligent systems.-Kluwer Academic Publishers. -1998.
- [Wermter, 1997] Wermter S. Hybrid approaches to neural network-based language processing. Technical Report TR-97-030. International Computer Science Institute, Berkely, California, 1997.
- [Bezdek, 1994] Bezdek J. What is computational intelligence?, in [Zurada J, Marks I, Robinson C. Computational Intelligence: Imitating Life, IEEE Press, New York, 1994.
- [Soucek, 1991] Soucek B., and the IRIS Group (eds), Neural and Intelligent Systems Integration, John Wiley and Sons, New York. - 1991.

INTEGRATED INTELLIGENT TECHNOLOGY IN HYBRID SYSTEMS

Komartsova L.G.

*The Bauman Moscow State Technical University,
Kaluga, Russia*

lkomartsova@yandex.ru

Problems of creating a hybrid intelligent decision support system are reviewed in this report.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8

A NEURAL NETWORK-LIKE COMBINATORIAL DATA STRUCTURE FOR SYMBOLIC MACHINE LEARNING ALGORITHMS

Xenia Naidenova* and Vladimir Parkhomenko**

* *Military medical academy,
Saint-Petersburg, Russian Federation*

ksennaidd@gmail.com

** *Saint-Petersburg state polytechnic university,
Saint-Petersburg, Russian Federation*

parhomenko.v@gmail.com

A new neural network-like combinatorial data-knowledge structure supporting symbolic machine learning algorithms is advanced. This structure can drastically increase the efficiency of inferring functional and implicative dependencies as like as association rules from a given dataset.

Keywords: Level-wise algorithm; inferring logical rules from a dataset; neural network-like data structure; knowledge representation.

INTRODUCTION

Mining logical rules (dependencies) from datasets in the form of association rules, implicative and functional dependencies, and key pattern attracts a great interest because of its potential very useful application. It has been proven that all the problems of logical rule inferring are algorithmically equivalent [Naidenova, 1992]. These problems are viewed as ones of supervised symbolic machine learning.

One of the algorithms for inferring logical dependencies is the algorithm using an effective inductive method of constructing sets of cardinality $(q+1)$ ($(q+1)$ -sets) from their subsets of cardinality q (q -sets). A $(q+1)$ -set can be constructed if and only if there exist all its proper q -subsets. For example, the algorithms Apriori, AprioriTid, and AprioriHybrid have been presented in [Agravalet et al., 1996; Stumme, 2002] for association rule mining. The same principle underlies the algorithm Titanic for generating key patterns [Stumme, 2002] and the algorithm TANE for discovering functional dependencies [Huhtala et al., 1999]. The level-wise method of $(q+1)$ -sets' construction has also been proposed for inferring good diagnostic tests for a given classification or class of objects [Megretskaya, 1988; Naidenova, 1992, 2005, 2012]. These tests serve as a basis for extracting functional dependences, implications, and association rules from a given dataset.

In all enumerated problems, the same algorithm deals with different sets of elements (items (values of attributes), itemsets, attributes, transactions, indices of itemsets or transactions) and checks the different properties of generated subsets. These properties can be, for example: “to be a frequent (large) itemset”, “to be a key pattern”, “to be a test for a given class of examples”, “to be an irredundant set of attribute values”, “to be a good test for a given class of examples”, and some others. If a constructed subset does not possess a required property, then it is deleted from consideration. This deletion reduces drastically the number of subsets to be built at all greater levels. In section 2, we introduce a Background algorithm solving the task of inferring all maximal subsets of set S (i.e., such subsets that cannot be extended) possessing a given PROPERTY. The set S can be interpreted depending on the context of a considered problem. This algorithm implements the level-wise inductive method of $(q+1)$ -sets' construction. In section 3, we consider some possible ways of increasing the efficiency of Background Algorithm. In Section 4, we propose a neural network-like combinatorial data structure for constructing $(q+1)$ -sets from their q -subsets.

1. BACKGROUND ALGORITHM

By $s_q = (i_1, i_2, \dots, i_q)$, we denote a subset of S , containing q elements of S . Let $S(\text{test-}q)$ be the set of subsets $s = \{i_1, i_2, \dots, i_q\}$, $q = 1, 2, \dots, nt$, satisfying the PROPERTY. Here nt denotes the cardinality of S . We

use an inductive rule for constructing $\{i_1, i_2, \dots, i_{q+1}\}$ from $\{i_1, i_2, \dots, i_q\}$, $q = 1, 2, \dots, nt-1$. This rule relies on the following consideration: if the set $\{i_1, i_2, \dots, i_{q+1}\}$ possesses the PROPERTY, then all its proper subsets must possess this PROPERTY too. Thus the set $\{i_1, i_2, \dots, i_{q+1}\}$ can be constructed if and only if $S(\text{test-}q)$ contains all its proper subsets.

Having constructed the set $s_{q+1} = \{i_1, i_2, \dots, i_{q+1}\}$, we have to determine whether it possesses the PROPERTY or not. If not, s_{q+1} is deleted, otherwise s_{q+1} is inserted in $S(\text{test-}(q+1))$. The algorithm is over when it is impossible to construct any element for $S(\text{test-}(q+1))$.

Background algorithm:

Inferring all maximal (not extended) subsets of S possessing a given PROPERTY.

1. Input: $q = 1$, $S = \{1, 2, \dots, nt\}$, $S(\text{test-}q) = \{\{1\}, \{2\}, \dots, \{nt\}\}$.

Output: the set S_{MAX} of all maximal subsets of S possessing the PROPERTY.

2. $S_q := S(\text{test-}q)$;

3. While $||S_q|| \geq q + 1$ do

3.1 Generating $S(q + 1) = \{s = \{i_1, \dots, i_{(q+1)}\} : (\forall j) (1 \leq j \leq q + 1) (i_1, \dots, i_{(j-1)}, i_{(j+1)}, \dots, i_{(q+1)}) \in S_q\}$;

3.2 Generating $S(\text{test-}(q + 1)) = \{s = \{i_1, \dots, i_{(q+1)}\} : (s \in S(q + 1)) \& (\text{PROPERTY}(s)) = \text{true}\}$;

3.3 Reducing $S(\text{test-}q)$: $S(\text{test-}q) = \{s = \{i_1, \dots, i_q\} : (s \in S(\text{test-}q)) \& ((\forall s')(s' \in S(\text{test-}(q + 1)) s \subset s') \& s \not\subset s')\}$;

3.4. $q := q + 1$;

3.5. $max := q$;

end while

4. $S_{MAX} := \emptyset$;

5. While $q \leq max$ do $S_{MAX} := S_{MAX} \cup \{s : s = \{i_1, \dots, i_s\} \in S(\text{test-}q)\}$;

5.1 $q := q + 1$;

end while

end

For inferring maximally redundant good tests for a given class of examples, we have that S is a set of indices of positive $S(+)$ and negative $S(-)$ examples, t is a set of values of some set of attributes, $s(t) \subseteq S$ is the set of indices of examples in which t appears, and $\text{PROPERTY}(s) = \mathbf{if} s \subset S_+ \mathbf{then true else false}$.

For key pattern, we have that S is a set of values of some attributes describing considered objects, $s \subset S$ and $\text{PROPERTY}(s) = \mathbf{if} \text{for } (\forall A_i) (A_i \in s) P(s) \neq P(s/A_i) \mathbf{then true else false}$, where $P(s)$ and $P(s/A_i)$ are the set of indices of object descriptions in which s and s/A_i appears, respectively. For irredundant implications, $\text{PROPERTY}(s)$ repeats the previous case, but it is necessary to check an additional property of s . This

property is “to be a test for a given set of positive examples”.

The most important factor of Background Algorithm’s computational complexity is the method of generating of q -sets in the level-wise manner. Generally, we use the following inductive rules, where SN is the family of sets S_q of cardinality q , $q = 1, \dots, nt$, $S_q \subseteq S = \{1, \dots, nt\}$ and $C_S(q)$ denotes the number of combinations of S on q .

(1) $q = 1, q + 1 = 2$;

$s_q = \{i\}$, $s_{(q+1)} = \{i, j\}$, $(\forall j) (i \neq j, \{j\} \in SN)$;

(2) $q = 2, q + 1 = 3$;

$s_q = \{i, j\}$, $s_{(q+1)} = \{i, j, l\}$, where l different from i, j and such that there are in SN

a) two sets $s_1 = \{i, l\}$, $s_2 = \{j, l\}$ or

b) $s = \{l\}$;

(3) $q = 3, q + 1 = 4$;

$s_q = \{i, j, m\}$, $s_{(q+1)} = \{i, j, m, l\}$, where l different from i, j, m and such that there are in SN

a) three sets $s_1 = \{i, j, l\}$, $s_2 = \{i, m, l\}$, $s_3 = \{j, m, l\}$

b) three sets $s_1 = \{i, l\}$, $s_2 = \{j, l\}$, $s_3 = \{m, l\}$ or

c) $s = \{l\}$;

(q) $q, q + 1$;

$s_q = \{i_1, i_2, \dots, i_q\}$, $s_{(q+1)} = \{i_1, i_2, \dots, i_q, l\}$, where l different from i_1, i_2, \dots, i_q and such there are in SN

a) sets the number of which is equal to $C_S(q) = C_S(nt - q)$ and the cardinality of which is equal to q , such that $\{i_1, i_2, \dots, i_{p-1}, i_{p+1}, \dots, i_q, l\} \setminus \{i_p\}$ for all $p = 1, \dots, q$ or

b) sets the number of which is equal to $C_S(q - 1) = C_S(nt - (q - 1))$, the cardinality of which is equal to $q - 1$, such that $\{i_1, i_2, \dots, i_q, l\} \setminus \{i_{p_i}, i_{p_j}\}$ for all $\{p_i, p_j\} \subseteq \{1, \dots, q\}$ or

c) sets the number of which is equal to $C_S(q - 2) = C_S(nt - (q - 2))$, the cardinality of which is equal to $q - 2$, such that $\{i_1, i_2, \dots, i_q, l\} \setminus \{i_{p_i}, i_{p_j}, i_{p_k}\}$ for all $\{p_i, p_j, p_k\} \subseteq \{1, 2, \dots, q\}$ or

d) sets the number of which is equal to $C_S(1) = C_S(nt - 1)$, the cardinality of which is equal to 1, such that $\{l\}$, $l \notin \{i_1, i_2, \dots, i_q\}$.

The Background Algorithm has an essential disadvantage consisting in the necessity to generate all subsets of s in S_q , $q = 1, 2, \dots, q_{max}$. But it is possible constructing directly an element $s \in S_q$, $s = \{i_1, i_2, \dots, i_q\}$ without generating all of its subsets.

2. A STRUCTURE OF INTERCONNECTED LISTS FOR BACKGROUND ALGORITHM

The inductive rules can be used not only for extending sets, but also for cutting off both the elements of S and the sets themselves containing these deleted elements. If element j enters in s_{q+1} , then it must enter in

q proper subsets of s_{q+1} . If we observe that j enters in only one doublet (pair), then it cannot enter in any triplet. If j enters in only one triplet, then it cannot enter in any quadruplet and so on. If an element enters in two and only two doublets, it means that it can enter only in one triplet. If an element enters in three and only three doublets, it can enter in only one quadruplet.

This reasoning is applicable to constructing triplets from doublets, quadruplets from triplets and so on. For instance, if a doublet enters in two and only two triplets, then it can enter in one quadruplet. If a triplet enters in two and only two quadruplets, then it can enter in only one set of five elements.

The removal of a certain element (or set of elements) draws the removal of doublets, triplets, quadruplets, ... into which it enters.

Let us name the procedure for removal of elements and sets containing these elements the procedure of "winnowing". We realize this procedure with the use of a Matrix of Correspondences the columns of which are associated with elements of S , and the rows are associated with subsets of S . An entrance $\{i, j\}$ in this matrix equals 1, if index associated with column j enters in s associated with row i . Let the set S be $\{\{1\}, \{2\}, \dots, \{14\}\}$. Consider the Matrix of Correspondences (Table 1) between the 2-component subsets of S possessing a given PROPERTY ($S(\text{test-2})$) and elements of S appearing in these subsets. In this matrix, the columns are ordered by increasing the number of subsets associated with the columns.

Table 1 - The Matrix of Correspondences for the S (test-2)

Subset	9	5	6	14	1	1	1	8	12	3	7	4	2
(9,11)	1				0	1	1						
(1,5)		1				1							
(5,12)		1						1					
(4,6)			1									1	
(6,8)			1				1						
(6,11)			1			1							
(1,14)				1		1							
(2,14)				1									1
(12,14)				1				1					
(2,10)					1								1
(3,10)					1					1			
(8,10)					1		1						
(1,2)						1							1
(1,4)						1							1
(1,7)						1					1		
(1,12)						1		1					
(3,11)							1			1			

Subset	9	5	6	14	1	1	1	8	12	3	7	4	2
(4,11)							1						1
(7,11)							1				1		
(8,11)							1	1					
(2,8)								1					1
(3,8)								1		1			
(4,8)								1					1
(7,8)								1			1		
(2,12)									1				1
(3,12)									1	1			
(4,12)									1				1
(7,12)									1		1		
(2,3)											1		1
(3,4)											1	1	
(3,7)										1	1		
(2,7)											1	1	
(4,7)											1	1	
(2,4)													1

Element 9 enters in one and only one doublet, hence (9,11) cannot be included in any triplet. We can delete the corresponding column and row. We conclude also that set (9,11) cannot enter in any triplet.

Element 5 enters in two and only two doublets, hence it is included in only one triplet (1,5,12). Element 5 cannot be included in any quadruplet. We can delete the corresponding column and rows 2, 3.

Element 6 enters in three and only three doublets, hence it is included in only one quadruplet (4,6,8,11). Element 6 cannot be included in a subset of five indices. We can delete the corresponding column and rows 4, 5, 6.

By analogous reason, we conclude that collection (1,2,12,14) cannot be extended and we can delete the corresponding column and rows 7, 8, 9. Note that all subsets (9,11), (1,5,12), (4,6,8,11), and (1,2,12,14) possess the PROPERTY.

Element 10 enters in three and only three doublets, hence it is included in only one quadruplet (2,3,8,10). This set does not possess the PROPERTY. In this case, we have to construct all the triplets with element 10. These triplets (2,8,10), (2,3,10), (3,8,10) do not possess the PROPERTY, it means that subsets (2,10), (3,10), (8,10) are maximal ones possessing the PROPERTY. Element 10 can be deleted together with rows 10, 11, 12. Table 2 shows the reduced Matrix of Correspondences.

Table 2 - The reduced Matrix of Correspondences (Reduction 1)

Subset	1	11	8	12	3	7	4	2
(1,2)	1							1
(1,4)	1						1	
(1,7)	1					1		
(1,12)	1			1				
(3,11)		1			1			
(4,11)		1					1	
(7,11)		1				1		
(8,11)		1	1					
(2,8)			1					1
(3,8)			1		1			
(4,8)			1				1	
(7,8)			1			1		
(2,12)				1				1
(3,12)				1	1			
(4,12)				1			1	
(7,12)				1		1		
(2,3)					1			1
(3,4)					1		1	
(3,7)					1	1		
(2,7)						1		1
(4,7)						1	1	
(2,4)							1	1

Element 1 enters in 4 doublets. In this case, we construct the following triplets including element 1: (1,2,4), (1,2,7), (1,2,12), (1,4,7), (1,4,12), (1,7,12). Only triplets (1,4,7) and (1,2,12) possess the PROPERTY. We conclude that element 1 cannot be included in any quadruplet possessing the PROPERTY; hence it can be deleted from consideration with rows 13, 14, 15, 16. Since $(1,2,12) \subseteq (1,2,12,14)$, we conclude that subset (1,2,12) is not maximal with respect to the PROPERTY.

Analogously, the consideration of element 11 leads to constructing the following subsets: (3,4,11), (3,7,11), (3,8,11), (4,7,11), (4,8,11), (7,8,11) from which only (7,8,11) and (4,8,11) possess the PROPERTY. We conclude that element 11 cannot be included in any quadruplet possessing the PROPERTY; hence it can be deleted from consideration with rows 17, 18, 19, 20. We also conclude that subset (7,8,11) is maximal with respect to the PROPERTY, but (4,8,11) does not. Table 3 shows the reduced Matrix of Correspondences.

Table 3 - The reduced Matrix of Correspondences (Reduction 2)

Subset	8	12	3	7	4	2
(2,8)	1					1

Subset	8	12	3	7	4	2
(3,8)	1		1			
(4,8)	1				1	
(7,8)	1			1		
(2,12)		1				1
(3,12)		1	1			
(4,12)		1			1	
(7,12)		1		1		
(2,3)			1			1
(3,4)			1		1	
(3,7)			1	1		
(2,7)				1		1
(4,7)				1	1	
(2,4)					1	1

With element 8, the following subsets can be constructed: (2,3,8), (2,4,8), (2,7,8), (3,4,8), (3,7,8), (4,7,8). But only (2,7,8) possesses the PROPERTY. We conclude that it is maximal with respect to the PROPERTY. Element 8 can be deleted together with rows 21, 22, 23, 24. For element 12, the following triplets can be constructed: (2,3,12), (2,4,12), (2,7,12), (3,4,12), (3,7,12), (4,7,12). Only (3,7,12), (4,7,12) possess the PROPERTY. Since element 12 cannot be included in any quadruplet possessing the PROPERTY, we conclude that (3,7,12), (4,7,12) are maximal with respect to the PROPERTY. Element 12 can be deleted together with rows 25, 26, 27, 28.

Table 4 shows the reduced Correspondent Matrix. In this table, (2,3,4,7), the union of all remaining subsets, possesses the PROPERTY, hence the process of generating subsets is over.

Table 4 - The reduced Matrix of Correspondences (Reduction 3)

Subset	3	7	4	2
(2,3)	1			1
(3,4)	1		1	
(3,7)	1	1		
(2,7)		1		1
(4,7)		1	1	
(2,4)			1	1

Currently, we have constructed $31 + 1 = 32$ subsets. Without the procedure of winnowing, it is necessary in Background Algorithm to form $91 + 38 + 3 = 91 + 41 = 132$ subsets, where 91 doublets, 38 triplets, and 3 quadruplets. The application of winnowing reduced the quantity of considered subsets to 123: $91 + 32 = 123$.

3. A SPECIAL COMBINATORIAL NETWORK FOR BACKGROUND ALGORITHM

The idea of the following algorithm is based on the functioning of a combinatory network structure, whose elements correspond to subsets of a finite set S generated in the algorithm. These elements are located in the network along the layers, so that each q -layer consists of the elements corresponding to subsets the cardinality of which is equal to q . All the elements of q -layer have the same number q of inputs or connections with the elements of previous $(q - 1)$ -level. Each element "is excited" only if all the elements of previous layer connected with it are active. The weight of connection going from the excited element is taken as equal to 1; the weight of connection going from the unexcited element is taken as equal to 0. An element of q -layer is activated if and only if the sum of weights of its inputs is equal to q . The possible number N_q of elements (nodes) at each layer is known in advance as the number of combinations of S on q . In the process of the functioning of the network the number of its nodes can only diminish.

An advantage of this network consists in the fact that its functioning does not require the complex techniques for changing the weights of connections and it is not necessary to organize the process of constructing q -sets from their $(q - 1)$ -subsets. The nodes of network can be interpreted depending on a problem to be solved. The assigned properties can be checked via different attached procedures.

If an activated node does not possess the assigned property, then it is excluded from consideration by setting to 0 all connections going from it to the nodes of above layer. Non-activated node does not require checking whether it possesses the PROPERTY or not. The work of this combinational network consists of the following steps:

Step 1. The setting of the first layer nodes of network to active state, the weights of connections leading to the second layer nodes are set equal to 1;

For each level beginning with the second one:

Step 2. The excitation of nodes, if they were not active and all their incoming traffic (links) have the weight equal to 1; checking the assigned property for the activated nodes of this layer;

Step 3. If the assigned property of node is not satisfied, then all the outgoing connections of this node are established to 0. If the assigned property of node is satisfied, then its outgoing connections are set to be equal to 1;

Step 4. The propagation of "excitation" to the nodes of the following higher layer (with respect to the current one) and the passage to analyzing the following layer;

Step 5. "The readout" of the active nodes not connected with above lying active nodes. Such nodes correspond to maximal (not extended) subsets possessing a given property.

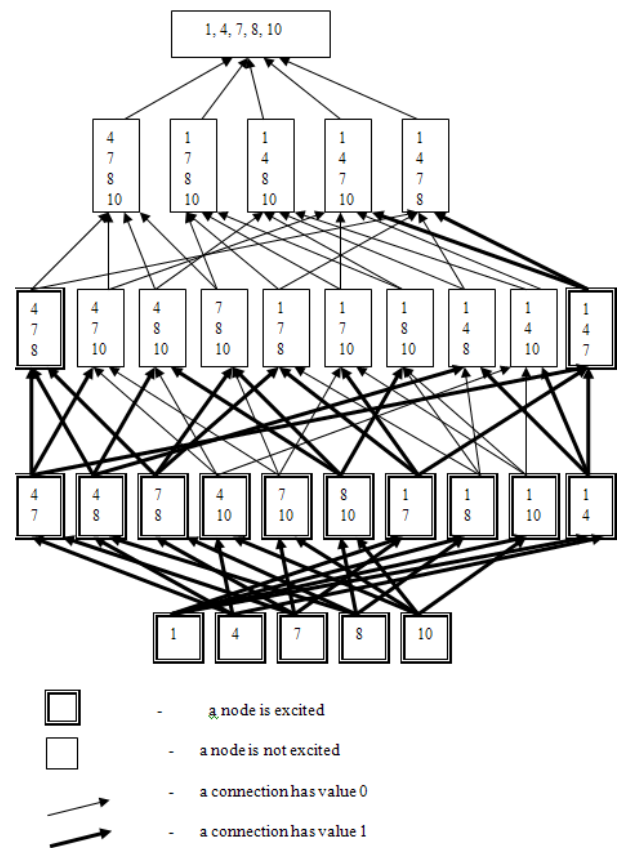


Figure 1 - An example of special combinatorial network

In Figure 1, all the nodes of two first levels are activated but nodes $\{4,10\}$, $\{7,10\}$, $\{1,8\}$, and $\{1,10\}$ do not possess the given property and they have no active outgoing links. At the third level, only two nodes are activated among which node $\{4,7,8\}$ does not possess the given property. As a result, we have two nodes corresponding to maximal subsets possessing the given property: $\{8,10\}$, $\{1,4,7\}$. In the process of network activating, only 12 nodes have been checked and 14 ones did not require to be checked.

Apparently, we can see that the size of network may be a problem if the data is large. But the decomposition of the main problem into sub-problems drastically diminishes the memory size of Background Algorithm. A subproblem is determined by a subnetwork generated by a node of the network.

Generally, the main advantages of combinational network are the following ones: 1) the size of network is computed in advance; 2) it is possible to decompose network into autonomic fragments; 3) different fragments of network can be joined via common nodes; 4) the states of nodes can be established by the use of attached procedures; 5) this can be used for problems of pattern recognition based on using logical rules [Naidenova, 2012].

This combinational network can be used for solving many problems of data mining such that finding frequent patterns, association rule mining, discovering functional dependencies and some others. The application of neural network models for these

problems is a new field for investigating. We can refer the readers only to one work in this direction related to optical neural network model used for mining frequent itemsets in large databases [Bhatnager, 2001]. The optical neural network model proposes the most optimized approach with only one database scan and parallel computation of frequent patterns.

CONCLUSION

In this paper, we describe an algorithm, called Background Algorithm, based on the method of mathematical induction. This algorithm is applicable to inferring many kinds of logical dependencies from a given dataset: implicative and functional dependencies, association rules, key patterns and some others. For the implementation of this algorithm, we proposed a neural network-like combinatorial structure of data and knowledge the advantage of which consists in the fact that the functioning of it does not require the complex techniques for changing the weights of connections. The nodes of network can be interpreted depending on a problem to be solved. The assigned properties of nodes can be checked via different attached procedures.

Furthermore, the advantages of combinatorial network are the following ones:

- the size of network is computed in advance;
- it is possible to decompose network into autonomic fragments which can operate in a parallel way;
- different fragments of network can be joined via common nodes;
- this network can be used not only for inferring logical rules from datasets but also for problems of pattern recognition based on these induced rules.

A model of pattern recognition in which the deductive and inductive reasoning rules interact has been given in [Naidenova, 2012].

BIBLIOGRAPHICAL REFERENCES

- [Agraval et al., 1996] Agraval, R. Fast discovery of association rules / R. Agraval, H. Mannila, R. Srikant, H. Toivonen, A. Verkamo // Advances in Knowledge Discovery and Data Mining. – California.: AAAI Press, 1996. – P. 307-328.
- [Bhatnager et al., 2001] Bhatnagar, D. Distributed approach for mining frequent itemsets using optical neural network model / D. Bhatnagar, N. Adlakha, A. A. Swaroopsaxena // International Journal of Engineering Science and Technology. – 2011. – Vol. 3, №5. – P. 3979-3981.
- [Huhtala et al., 1999] Huhtala, Y. Tane: an efficient algorithm for discovering functional and approximate dependencies / Y. Huhtala, J. Karkkainen, P. Porkka, H. Toivonen // The computer Journal. – 1999. – Vol. 42, №2. – P. 100-111.
- [Megretskaya, 1988] Megretskaya, I. A. Construction of natural classification tests for knowledge base generation / I. A. Megretskaya // The Problem of the expert system application in the national economy: Reports of the Republican Workshop. – Kishinev.: Mathematical Institute with Computer Centre of Moldova Academy of Sciences, 1988. – P. 89-93.
- [Naidenova, 1992] Naidenova, X.A. Machine learning as a diagnostic task / X. A. Naidenova // Knowledge-Dialog-Solution: Materials of the Short-Term Scientific Seminar – Saint-Petersburg.: State North-West Technical University, 1992. – P. 26-36.

[Naidenova, 2005] Naidenova, X.A. DIAGARA: An incremental algorithm for inferring implicative rules from examples / X. A. Naidenova // International Journal Information Theories & Applications. – 2005. – Vol. 12, №2. – P. 171-186.

[Naidenova, 2012] Naidenova, X.A. Constructing Galois Lattices as a commonsense reasoning process / X. A. Naidenova // Diagnostic test approaches to machine learning and commonsense reasoning systems. – USA.: IGI Global, 2012. – P. 34-70.

[Stumme, 2002] Stumme, G. Efficient Data Mining Based on Formal Concept Analysis / G. Stumme // DEXA 2002, LNCS 2453. – Berlin, Heidelberg.: Springer-Verlag, 2002 – P. 534-546.

НЕЙРОПОДОБНАЯ КОМБИНАТОРНАЯ СТРУКТУРА ДАННЫХ ДЛЯ АЛГОРИТМОВ СИМВОЛЬНОГО МАШИННОГО ОБУЧЕНИЯ

Найдёнова К.А.* , Пархоменко В.А.**

**Военно-медицинская академия,
Санкт-Петербург, Российская Федерация*
ksennaidd@gmail.com

*** Санкт-Петербургский государственный
политехнический университет,
Санкт-Петербург, Российская Федерация*
parhomenko.v@gmail.com

Предложена новая нейроподобная комбинаторная структура данных и знаний, увеличивающая эффективность алгоритмов символьного машинного обучения для вывода различного рода логических правил из данных, таких как имплицативные и функциональные зависимости, ассоциативные правила, паттерны, описывающие классы объектов. Все перечисленные зависимости генерируются с помощью одного и того же алгоритма и одной и той же предложенной структуры данных. Данная структура также интегрирует задачи вывода правил и их использования при распознавании образов.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.02 : 004.9

ПРИМЕНЕНИЕ КЛАСТЕРНЫХ СТРУКТУР ДЛЯ ПРЕДСТАВЛЕНИЯ КЛАССОВ ПРИ ПОСТРОЕНИИ СИСТЕМ РАСПОЗНАВАНИЯ

Анищенко И.С., Жукевич А.И., Родченко В.Г.

*Гродненский государственный университет имени Янки Купалы,
г. Гродно, Республика Беларусь*

`ivan.ani@mail.ru`

`san@grsu.by`

`rovar@mail.ru`

Реализация системы распознавания начинается с определения алфавита классов, формирования априорного словаря признаков и построения обучающей выборки. Внутри этой выборки каждый объект представляется в виде многомерного вектора в пространстве, сформированном на основе априорного словаря. Применение кластерных структур позволяет формализовать процедуру представления образов классов и производить оценку взаимного размещения образов и эталонов классов при построении систем распознавания.

Ключевые слова: интеллектуальная система, распознавание образов, алфавит классов, словарь признаков, классифицированная обучающая выборка, кластерная структура.

ВВЕДЕНИЕ

Процесс построения системы распознавания на основе анализа наблюдаемых данных предполагает выполнение трех основных этапов. Первый из них является *подготовительным* и связан с определением алфавита классов, формированием априорного словаря признаков и построением исходной классифицированной обучающей выборки. Второй этап является *центральный* и направлен на реализацию процедуры *обучения* [Васильев, 1989]. В результате выполнения из априорного словаря признаков (АСП) исключаются все малоинформативные признаки, которые не обеспечивают разделение формальных образов эталонов классов в признаковом пространстве принятия решений. Заключительный третий этап связан с выполнением процедуры *принятия решения*, когда исследуемый образ либо классифицируется к одному из предусмотренных классов, либо выделяется в отдельный джокер-класс [Родченко, 2006].

Формально весь процесс распознавания может быть реализован в результате выполнения следующей последовательности преобразований:

$$S \xrightarrow{F_1} C \xrightarrow{F_2} A \xrightarrow{F_3} T \xrightarrow{F_4} A^* \xrightarrow{F_5} E \xrightarrow{F_6} R$$

где S – алфавит классов; C – словарь наблюдаемых (измеряемых) характеристик; A – априорный

словарь признаков; T – классифицированная обучающая выборка; A^* – уточненный словарь признаков для построения пространства решений; E – множество эталонов классов; R – множество решений; F_1 – алгоритм получения наблюдаемых характеристик; F_2 – алгоритм построения априорного словаря признаков; F_3 – алгоритм формирования классифицированной обучающей выборки; F_4 – алгоритм сепарирования признаков из априорного словаря по степени их информативности для построения пространства решений; F_5 – алгоритм построения образов эталонов классов в пространстве решений; F_6 – алгоритм процедуры принятия решения.

Для формального представления образов классов или эталонов классов в многомерном признаковом пространстве предлагается на основе использования всех экземпляров классов строить соответствующие кластерные структуры.

Построение кластерных структур

Предположим, что определены алфавит классов $S = \{S_1, S_2, \dots, S_k\}$ и априорный словарь признаков $A = \{A_1, A_2, \dots, A_n\}$.

Пусть каждый отдельный объект класса описывается n признаками из АСП в виде вектор-столбца $x^T = (x_1, x_2, \dots, x_n)$, где x_i – значение i -го признака, и однозначно ассоциируется с одним из классов.

Набор объектов отдельного класса образует исходное описание этого класса в априорном признаковом пространстве. Объединение всех объектов из всех классов образует классифицированную обучающую выборку, которая описывается в виде таблицы типа “*объект-свойство*” и формально представляется в виде матрицы $X_n \times m$, где $m=m_1+m_2+\dots+m_k$, а m_i - количество объектов i -го класса.

Для формального представления образов классов предлагается на основе всех экземпляров класса строить соответствующую кластерную структуру. Поскольку каждый отдельный экземпляр класса представляет собой вектор в пространстве R^n с координатами вершины (x_1, x_2, \dots, x_n) , где x_i - значение i -го признака, то объединение всех векторов одного класса в кластерную структуру и будет представлять собой формальное описание класса.

Построение кластерных структур можно осуществить на основании универсального алгоритма, предусматривающего выполнение 4 шагов:

1. Начальная инициализация центров кластеров (например, можно задать начальные значения случайно выбранными точками из пространства, в котором определены данные; можно выбрать случайные точки из входного массива данных и т.д.).
2. E-шаг (expectation): происходит ассоциация между элементами данных и кластерами, которые представлены в виде своих центров (центроидами).
3. M-шаг (maximization): пересчитываются центры кластеров, как средние значения от данных, которые были включены с соответствующий кластер (другими словами, происходит модификация параметров модели таким образом, чтобы максимизировать вероятность попадания элемента в выбранный кластер). В случае, если кластер после шага 2 оказался пустым, то происходит инициализация каким-либо другим способом.
4. Шаги 2-3 повторяются до сходимости, либо пока не выполнится другой критерий остановки алгоритма (например, превышение некоторого числа итераций).

Однако использование данного подхода имеет и ряд недостатков, в частности:

- Алгоритм не гарантирует определения лучшего из возможных расположений центров гиперсфер (достижение глобального минимума суммарного квадратичного отклонения), но гарантирует сходимость к какому-либо решению, т.е. итерации не застревают.
- Результат построения кластерной структуры зависит от выбора исходных центров гиперсфер.

Об одной реализации общего подхода

Процесс построения кластерной структуры начинается с поиска наиболее удаленного от других

вектора. Затем определяется ближайший к найденному вектору представитель класса, и он включается в состав кластера. Затем вычисляются и запоминаются значения координат вспомогательного вектора, который указывает на середину отрезка, соединяющего два очередных экземпляра класса. В результате формируется “скелет” кластерной структуры для построения формального образа класса, который содержит $2 * m_i - 1$ векторов, из которых m_i векторов-экземпляров i -го класса (где m_i - количество объектов i -го класса) и $m_i - 1$ вспомогательных векторов. В дальнейшем каждый экземпляр “скелета” выступает в качестве центра гиперсферы при построении кластерной структуры, представляющей собой объединение областей, образованных пересекающимися гиперсферами.

1. Возьмем наиболее удаленный от всех экземпляров класса $X^{(1)}$ и найдем для него ближайший экземпляр $X^{(2)}$, и расстояние между ними обозначим $l^{(1)}$.

2. Построим гиперсферы (далее сферы) радиуса $r^{(1)} = \frac{l^{(1)}}{2}$ с центрами в $X^{(1)}, X^{(2)}$. Обозначим точку касания двух сфер - $O^{(1)}$ с координатами $(o_1^{(1)}, \dots, o_n^{(1)})$ и построим сферу радиуса $r^{(1)}$ с центром в $O^{(1)}$.

3. Для экземпляра $X^{(2)}$ найдем ближайший экземпляр $X^{(3)}$, и расстояние между ними обозначим $l^{(2)}$, причем из поиска исключаем $X^{(1)}$. Построим сферы радиуса $r^{(2)} = \frac{l^{(2)}}{2}$ с центрами в $X^{(2)}, X^{(3)}$ и получим точку касания сфер $O^{(2)} = (o_1^{(2)}, \dots, o_n^{(2)})$. Построим сферу радиуса $r^{(2)}$ с центром $O^{(2)}$. Поскольку $X^{(2)}$ является центром двух сфер радиуса $r^{(1)}$ и $r^{(2)}$, то для данного экземпляра выбираем сферу с максимальным радиусом.

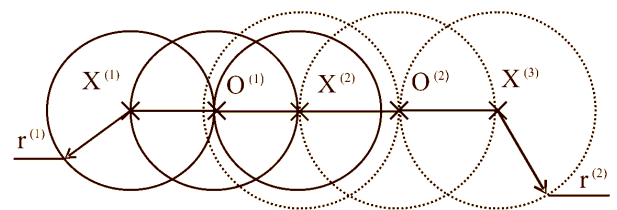


Рисунок 1 - Сферы с центрами $X^{(1)}, O^{(1)}, X^{(2)}, O^{(2)}, X^{(3)}$

Аналогично продолжим построение сфер для остальных экземпляров класса. Опишем результат такого построения с помощью таблицы, в которой первый столбец содержит порядковые номера сфер, второй столбец содержит координаты центра сферы с радиусом, значение которого в третьем столбце. Четвертый столбец принимает значение 1 - если центром сферы является x_i и 0 - если центром сферы оказывается точка пересечения o_i .

№	Центр	Радиус	Признак
1	$X^{(1)}$	$r^{(1)}$	1

2	$O^{(1)}$	$r^{(1)}$	0
3	$X^{(2)}$	$r^{(2)}$	1
....
2m-2	$O^{(m-1)}$	$r^{(m-1)}$	0
2m-1	$X^{(m)}$	$r^{(m-1)}$	1

Экземпляры $X^{(1)}$ и $X^{(m)}$ являются “крайними”, а потому для них в таблицу записываем радиусы $r^{(1)}$ и $r^{(m-1)}$. Для каждого экземпляра класса $X^{(2)}, \dots, X^{(m-1)}$ в таблицу записываем максимальное значение радиуса, связанных с построением соответствующих сфер. В результате получаем, что область, объединяющая все сферы, и будет представлять собой кластер класса.

При построении кластеров используются гиперсферы, для которых объем V радиуса r в пространстве R^n равен для n четных $V = \frac{2^2 \pi^2}{n!} r^n$ и

нечетных $V = \frac{2^2 \pi^2}{n!} r^n$ соответственно [Розенфельд, 1966].

Объем построенного кластера можно вычислить по формуле $V = \sum_{j=1}^{2m-1} V^{(j)} - U$, где U представляет собой объем пересечения сфер, образующих кластер. Для вычисления значения U можно воспользоваться методом Монте-Карло.

Рассмотрим область пересечения G двух сфер C_1 с центром в точке M_1 и радиусом r_1 , и C_2 с центром в точке M_2 и радиусом r_2 в пространстве R^n .

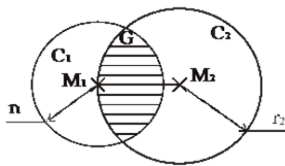


Рис.2. Пересечение сфер C_1 и C_2

Для вычисления объема V_G . возьмем сферу C_1 , объем которой V_1 . Выберем N случайных точек, равномерно распределенных в C_1 , и обозначим через N' количество точек, попавших в G . Точка $M_G \in G$ тогда и только тогда, когда $|M_1 - M_G| \leq r_1$ и $|M_2 - M_G| \leq r_2$.

Очевидно, что при большом значении N , $\frac{N'}{N} \approx \frac{V_G}{V_1}$, а значит $V_G \approx V_1 \frac{N'}{N}$. Тогда объем области объединения двух сфер C_1 и C_2 рассчитывается по формуле $V = V_1 + V_2 - V_G$ или $V \approx V_2 + V_1 \left(1 - \frac{N'}{N}\right)$.

Теперь перейдем к вычислению объема U пересечения сфер кластерной структуры. Найдем для j -ой сферы, где $j=1, 2m-1$, все сферы с которыми она пересекается, причем к рассмотрению будем брать сферы, у которых порядковый номер больше j . После чего оценим объем по формуле

$$U^{(j)} \approx V^{(j)} \frac{N'_j}{N_j}$$

Для последней сферы, порядковый номер которой равен $2m-1$, имеем $U^{(2m-1)}=0$. Тогда суммарный объем пересечения сфер кластера

$$U = \sum_{j=1}^{2m-1} U^{(j)}$$

Подставляем полученное значение U в формулу для V и получаем искомое значение объема

$$V \approx \sum_{j=1}^{2m-1} V^{(j)} - \sum_{j=1}^{2m-1} U^{(j)}$$

Плотность кластера ρ можно вычислить по формуле $\rho = V/k$, где k количество экземпляров класса.

ЗАКЛЮЧЕНИЕ

При построении систем распознавания первоначально каждый отдельный экземпляр класса формально может быть представлен в виде вектора в многомерном признаковом пространстве. Формальное же описание образа класса предложено реализовать в виде кластерных структур, получаемых путем объединения соответствующих гиперсфер.

В статье описан метод, который позволяет представить образ класса в виде кластерной структуры, получаемой путем объединения гиперсфер в многомерном признаковом пространстве.

Реализован оригинальный алгоритм построения кластерных структур, которым предусматривается возможность вычисления объема и плотности кластерной структуры, что позволяет производить оценку ее компактности.

Предложенный подход базируется на применении аппарата кластерного анализа, а для вычисления объема пересечения гиперсфер в многомерном признаковом пространстве используется метод Монте-Карло.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Васильев, 1989] Васильев, В.И. Проблема обучения распознаванию образов / В.И. Васильев – К: Выща шк. Головное изд-во, 1989.
- [Родченко, 2006] Родченко, В.Г. Об одном методе реализации процедуры обучения при построении системы распознавания образов / В.Г. Родченко. // Известия Гомельского государственного университета имени Ф.Скорины. - 2006. - №4. - С.73-76
- [Розенфельд, 1966] Розенфельд, Б.А. Многомерные пространства / Б.А. Розенфельд – М: Наука, 1966.

THE USE OF CLUSTER STRUCTURES TO REPRESENT OF CLASSES FOR CONSTRUCTION OF RECOGNITION SYSTEMS

Anishchenko I.S., Zhukevich A.I.,
Rodchenko V.G.

*Yanka Kupala Grodno State University, Grodno,
Republic of Belarus*

ivan.ani@mail.ru

san@grsu.by

rovar@mail.ru

The construction of the recognition system begins with the definition of the alphabet classes, the formation of a priori dictionary features and builds a learning sample. Within this sample of each object is represented in the form of multidimensional vector in the space, formed on the basis of a priori dictionary. The use of cluster structures allows to formalize the procedure of submission of patterns classes and to assess the mutual placement of patterns and etalons classes when building of the recognition systems.

INTRODUCTION

At building of real systems on the basis of use of pattern recognition theory apparatus there is a necessity of formal representation of classes in corresponding multidimensional feature space. Such space is built on the basis of a priori feature vocabulary, or on the basis of the specified feature vocabulary.

For formal representation of pattern class in multidimensional feature space building of corresponding cluster structure on the basis of all class standards is suggested. Each class standard represents a vector in space R^n with top coordinates (x_1, x_2, \dots, x_n) , where x_i – i feature value. Association of all vectors of one class in cluster structure also will represent the formal description of a class in multidimensional feature space.

MAIN PART

Cluster structure building process begins with search of such example class which is the most removed from others. Then the nearest to a found example representative of a class is defined, and it joins the cluster structure as a following element. Besides, are calculated and if necessary are remembered coordinates values of an auxiliary vector which indicates the midpoint, connecting two next class examples. As a result cluster "skeleton" for building of pattern class is formed.

Let to a priori feature vocabulary be formed $P=\{P_1, P_2, \dots, P_n\}$, and every class example is described on the basis of n features from this vocabulary. Formally each such example is represented in the form of vector-column $X^T = (x_1, x_2, \dots, x_n)$, where x_i – i feature value.

We will begin cluster structure building with search

of the most remote from all class examples. We will designate it $X^{(1)}$. The given example will be the first element of formed cluster structure.

Then we will find for $X^{(1)}$ the nearest example $X^{(2)}$, the distance between them will be designated as $l^{(1)}$. We will build hyperspheres (further spheres) of radius $r^{(1)}$, where $r^{(1)} = l^{(1)}/2$ with centers in $X^{(1)}, X^{(2)}$. We will designate the contact point of two spheres – $O^{(1)}$ with coordinates $(o_1^{(1)}, \dots, o_n^{(1)})$ and we will build radius sphere $r^{(1)}$ with center in $O^{(1)}$. Further for the example $X^{(2)}$ we will find the nearest example $X^{(3)}$ and distance between them will be designated as $l^{(2)}$, and is excluded $X^{(1)}$. We will build radius spheres $r^{(2)}$, where $r^{(2)} = l^{(2)}/2$ with centers in $X^{(2)}, X^{(3)}$ and we will get the sphere contact point $O^{(2)}=(o_1^{(2)}, \dots, o_n^{(2)})$. We will build radius sphere $r^{(2)}$ with center $O^{(2)}$. The example $X^{(2)}$ is a center of two radius spheres $r^{(1)}$ and $r^{(2)}$, the maximum radius is chosen for the cluster structure description.

As a result we received that the area uniting all spheres will represent cluster structure which is subject of a formal pattern class in multidimensional feature space.

For cluster structure characteristics estimation the possibility of cluster structure volume and density calculation is provided.

CONCLUSION

In the article the method which allows presenting a pattern class in cluster structure, received by association of hyperspheres in multidimensional feature space, is described.

This method can be used at designing and building of recognition systems for procedure performance of the analysis of mutual placing of class patterns and standards in multidimensional feature space.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 330.115

ИТЕРАЦИОННЫЕ МЕТОДЫ СОГЛАСОВАННОЙ ОПТИМИЗАЦИИ С ИДЕНТИФИКАЦИЕЙ ЭЛЕМЕНТОВ МОДЕЛИ ВЫБОРА

Виноградов Г.П.* , Кузнецов В.Н.*

**Тверской государственной технической университет, г. Тверь, Россия*

WGP272NG@mail.ru

Исследуется сходимость интерактивной процедуры согласованной оптимизации в задаче управления эволюцией организационно-технологической системы при различных вариантах информированности центра о возможностях агентов. Приведены условия, при которых возможно построение процедуры интерактивного обмена информацией между центром и агентами, гарантирующими сходимость решения задачи согласованной оптимизации.

Ключевые слова: активная система; интеллектуальный агент; согласованная оптимизация; принятие решений.

ВВЕДЕНИЕ

Статья посвящена итерационным методам решения задачи оптимального согласованного планирования, когда центр, не обладая полной информацией о возможностях активных, целеустремленных агентов, вынужден осуществлять одновременно с определением плановых заданий идентификацию модели своих представлений по отдельным ее составляющим: исходным данным, ограничениям, взаимосвязи между входами и выходами, о технологических множествах агентов, о целевых функциях агентов и т.п. Такие методы, с одной стороны, расширяют вычислительные возможности теории согласованного планирования и управления, а с другой – могут использоваться как средство моделирования и управления эволюционирующими системами. Это соответствует ситуации, когда объект управления для центра плохо поддается адекватному моделированию из-за его сложности, проблемы получения точной информации об его параметрах, неудовлетворительной структурированности, плохой формализуемости, распределения знания о предельных возможностях подсистем среди агентов. Причем в последнем случае информация центром может быть получена только в результате коммуникативного процесса с агентами.

Поэтому расчет согласованного плана в этих условиях превращается в сложный вычислительный процесс итерационного типа с интерактивным взаимодействием его участников, цель которого формирование у всех участников согласованных представлений о ситуации выбора. Центр для

формирования представлений о возможностях агентов использует результаты анализа возникающих в системе ситуаций от управляющих воздействий, возмущений.

Представления центра в этом случае можно формализовать в виде нестационарной модели, свойства и структура которой определяются информированностью центра. Тогда используемый центром итерационный метод $\varphi \in \{R^n \rightarrow R^n\}$ должен порождать сходящуюся последовательность представлений, которая характеризуется помехоустойчивостью, независимостью логической структуры оператора φ от конкретных исходных данных. Условием, обеспечивающим реализацию перечисленных свойств, является необходимость введения в итерационный оператор φ параметров, характеризующих объект управления: среду интересы участников процесса, – например, такого типа $\{x_{h+1} = \varphi[\Omega_h]x_h\}$, $h = 1, 2, \dots$, где $\Omega_h = \{y_h, \omega_h, \dots\}$ – перечисленная система параметров (числовых, символьных, лингвистических, предикатных) со значениями на момент h .

Особенность этой схемы состоит в необходимости использования алгоритмов, позволяющих получить информацию, доопределяющую и уточняющую параметры итерационного оператора. Основным источником такой информации являются взаимодействующие агенты, а средством ее получения – интерактивные алгоритмы.

1. Постановка задачи

Рассмотрим задачу планирования в двухуровневой организационно-технологической системе (ОТС), состоящей из центра и m агентов, обладающих свойствами активности, автономности, креативности. Каждый агент управляет каким-либо одним технологическим узлом. План для k -го агента описывается n_k -мерным вектором

$$x_k = \{x_{kj}, k = \overline{1, n_k}\},$$

положительные компоненты которого указывают на выпускаемые продукты, а отрицательные – на затраты для выпуска заданной продукции. Возможности k -го агента по выпуску продукции описываются технологическим множеством – областью X_k в пространстве из n_k измерений: любой допустимый план x_k k -го агента должен принадлежать области X_k :

$$x_k \in X_k, k = \overline{1, m}.$$

План ОТС будет описываться вектором $x = \{x_k, k = \overline{1, m}\}$ с размерностью $\sum_{k=1}^m n_k = N$.

Очевидно, что $x \in X = \prod_{k=1}^m X_k$. Допустимый план

x должен удовлетворять также ряду глобальных ограничений вида $G(x) \geq b$, где $G(x) = \{g_i(x), i = \overline{1, n}\}$, $b = \{b_i, i = \overline{1, n}\}$, n – количество ограничений. Будем считать, что функции $g_i(x), i = \overline{1, n}$ – вогнутые, дифференцируемые функции, а множество X – выпуклое множество. Тогда задача, решаемая центром, может рассматриваться как задача вогнутого программирования

$$f(x) \rightarrow \max$$

$$G(x) \geq b, x \in X = \prod_{k=1}^m X_k \quad (1)$$

Задача (1) является задачей оптимального планирования (ОП) [Полтерович, 1969]. Ее особенность заключается в том, что центр не знает все потенциальные возможности агентов. Поэтому ее решение зависит от информации о своих возможностях, которую агенты сообщают центру.

Пусть интересы агента описываются вектором $o_k \in O_k^o$, где O_k^o – множество состояний, имеющих различную ценность для агента. Значения вектора o_k определяются значениями вектора $y_k = \{y_{k1}, y_{k2}, \dots, y_{km_k}\}$ фактического выпуска продукции, полученными при реализации плана x_k агентом. При управлении технологическим узлом

свои предпочтения на множестве O_k^o агент реализует путем выбора значений режимных параметров z_k , которые в свою очередь определяют значения вектора y_k , а значит и o_k . Справедливо условие $z_k \in Z_k$, где Z_k – множество допустимых значений режимных параметров, определяемое технологическим регламентом. На технологический узел воздействуют различного рода возмущающие воздействия $\xi_k \in \Omega_k$, из которых часть $\zeta_k \in \Xi_k$ агент принимает во внимание. Поэтому $\Xi_k \subseteq \Omega_k$ и $\Xi_k \cap \Omega_k \neq \emptyset$.

Поведение человека как интеллектуального агента, зависящее от его субъективных представлений о ситуации выбора рассматривалось в работах [Виноградов, 2011, Vinogradov, 2009, Чхартишвили, 2004]. В них было показано, что принимаемое решение агентом о способе действия определяется его оценками компонент ситуации целеустремленного состояния, зависящими от структуры его информированности I_t , которая определяется его знаниями, убеждениями, ценностями, нормами, опытом. В этих же работах была предложена модель принятия решений агентом, позволяющая учитывать его индивидуальные оценки компонент ситуации целеустремленного состояния, которая имеет следующий вид

$$P_k(z_k) = \text{Arg max } E\varphi_k(o_k, u_k)$$

$$\zeta_k \in \Xi_k, z_k \in Z_k(I_t^i), u_k \in U_k,$$

$$o_k = o_k(y_k) \in O_k^o, y_k = f_{I_t}(z_k, \zeta_k) \in X_k, \quad (2)$$

$$O_k^o = \{o_{kj}^o, j = \overline{1, N_k}\}, I_t \subseteq M,$$

$$\chi_1^k(E\varphi_i) \geq \chi_1^{ko}, \chi_2^k(EE_i) \geq \chi_2^{ko},$$

где $E\varphi_k$ и EE_k – оценки агента удельной ценности ситуации целеустремленного состояния по результату и эффективности; $\chi_l^{ko}, l = \overline{1, 2}$ – оценки, отражающие эмоциональное отношение агента к ситуации выбора, u_k – управляющее воздействие центра.

Оптимальное решение задачи (1), учитывающей субъективное поведение агента в форме (2), практически невозможно. Проблема состоит в построении моделей выбора и интерактивного процесса обмена информацией между центром и агентами так, чтобы получить оптимальное решение

задачи (1, 2) и создать для каждого агента условия для наиболее полного использования своих возможностей в интересах системы, не используя детальную математическую модель их производственных возможностей. На возможность построения таких процедур указано в работе [Полтерович, 1969].

2. Задача согласованной оптимизации с доопределением модели выбора

Математическая постановка задачи оптимального согласованного планирования имеет вид: $f(x) \rightarrow \max; G(x) \leq b; x \in X; x \in S$. Здесь

$S = \prod_{k=1}^m s_k$ множество согласованных планов, таких

что $s_k = \{x_k \mid \max E\varphi_k(x_k^*, x_k) \leq E\varphi_k(x_k, x_k^*)\}$, где

$$E\varphi_k(x_k^*, x_k) = E\varphi_k(x_k^*) + \begin{cases} c_k & \text{при } x_k^* = x_k \\ 0 & \text{при } x_k^* \neq x_k \end{cases}$$

целевая функция k -го агента с учетом фонда материального поощрения c_k , планируемого центром за выполнение предлагаемого им плана x_k ; x_k^* – план производственной программы, выбираемый k -м агентом с учетом своих интересов и технологических возможностей.

Допустим, что суммарный фонд материального поощрения зависит от результатов работы системы в целом, и центр может назначать c_k исходя из очевидного ограничения $\sum_{k=1}^m c_k \leq c(x)$.

Предположим, что выполняется условие благожелательности агента по отношению к центру. Оно заключается в том, что если максимальный выигрыш k -го агента достигается как при плане, назначаемом центром, так и при некоторых других возможных значениях x_k^* , то агент предпочтет план центра x_k .

Необходимым и достаточным условием принадлежности плана x_k к согласованным ($x \in S$) является соотношение

$$\forall k : E\varphi_k(x_k) \geq \max_{x_k^* \in X_k} E\varphi_k(x_k^*).$$

Следовательно, условия согласованного планирования будут иметь вид: $c(x) - \sum_{k=1}^m c_k \geq 0$,

$$E\varphi_k(x_k) + c_k \geq \max_{x_k^* \in X_k} E\varphi_k(x_k^*).$$

Если функции $E\varphi_k(x)$, $g(x)$, $c(x)$ являются вогнутыми функциями, а множество X – выпуклым множеством, то задача оптимального согласованного планирования (ОСП) является задачей вогнутого программирования.

Пусть в задаче согласованного планирования такие элементы, как целевые функции агентов

$E\varphi_k(x_k) \in \{R^{m_k} \rightarrow R\}$, вектор правых частей

системы ограничений $g \in \{R^{m_k} \rightarrow R^n\}$, условия

включения $x \in X$ и $x \in S$ – плохо формализуемы. Это означает, что способа конструктивного задания элементов $E\varphi_k$, g , X и S нет. Центр их определяет

по результатам наблюдений, экспериментов, экспертных оценок и т.п. Это означает, что ему на момент h известны пары множеств X' и X'' ,

S' и S'' , G' и G'' таких, что $g(x) \leq b \forall x \in G'$ и

$$g(x) > b \forall x \in G'', x \in X', X'' \subset R^{m_k} \setminus X',$$

$$x \in S', S'' \subset R^{m_k} \setminus S'$$

Относительно целевых функций агентов $E\varphi_k$

представления центра состоят в том, что имеется

множество $O^0 \subset R^n$ (как правило, оно является конечным) такое, что для любого его элемента

$o \in O^0$ известно либо значение $E\varphi_k$, либо

некоторые характеристики этого значения, например, $\text{sgn}(E\varphi_k)$, либо подмножество

согласованных планов для k -го агента, когда

$x_k \in S'_k$, либо принадлежность $E\varphi_k$ некоторому

интервалу и т. п. Это означает, что центр обладает

правилом $x(o) \rightarrow \Delta(x(o)) \subset R$ для всех $o \in O^0$, то

есть центр изучает информацию о потребностях, мотивирующих поведение агентов. Поскольку O

конечно, то и множество $\Delta(x(o))$ для всех $o \in O^0$

конечно. Центр предполагает, что существует

дополнение $X \setminus X'$, которое он рассматривает как

резервы агентов, которые он должен выявить. Тогда

доопределение модели согласованного планирования

будет состоять в выявлении $X' \subseteq X, S' \subseteq S$ и $S' \cap X'$ таких, что

$E\varphi_k(x_k \in X') \leq E\varphi_k(x_k \in X)$. Тогда задача

оптимального согласованного планирования

становится конструктивным описанием

символической модели в виде

$$f(x) \rightarrow \max; G(x) \leq b; x \in X'; x \in S' \quad (3)$$

и допускает аналитические исследования и численные расчеты.

Построение модели (3) предполагает адаптацию представлений центра о технологических множествах агентов путем активного и пассивного

экспериментирования для организации процесса последовательного пополнения данных о множествах

$X', X'', S', S'', G', G''$ и $\{E\varphi_k(x(o)), o \in O^o\}$.

Это дает возможность построить последовательность представлений об этих множествах, например, такую, что

$\lim_{h \rightarrow \infty} (X_k \setminus X_k^h) = \emptyset$. В основе такого процесса

лежит предположение о том, что агенты обладают свойствами активности и креативности. При выполнении гипотезы рационального выбора агентами центр может формировать управляющее воздействие $u^{(h)} = \{u_k^{(h)}, k = \overline{1, m}\}$ такое, что агенты будут мотивированы выбирать напряженные планы.

Обозначим через $\Sigma^{(h)}$ материал результатов наблюдений на момент h , тогда итерационный оператор для модели (3) примет вид $\varphi[\Sigma^{(h)}](x)$, а через $\{E\varphi_k(x(o)), o \in O^o\}$, $g^{(h)}(x), X'_h, S'_h$ – представления центра о модели выбора (3). На каждом шаге h центр решает задачу согласованной оптимизации

$$\sup\{f(x): g^{(h)}(x) \leq b^{(h)}, x \in X^{(h)}, x \in S'\} \quad (4)$$

и организует процесс по итерационной схеме

$$x^{(h+1)} = \varphi[\Sigma^{(h)}](x^{(h)})_{\forall h}, \text{ где } x^{(h+1)} \text{ является}$$

одним из компонентов $u^{(h+1)}$.

В качестве конструктива можно использовать любой алгоритм адаптации, например, стохастической аппроксимации или методы деформируемого многогранника. Пополнение материала наблюдений может осуществляться следующим образом. Пусть центр построил вариант плана $X^{(h)}$ путем решения задачи (4). В пространстве плановых показателей определяется точка $x^{-(h)}$ более предпочтительного плана, при котором $f(x^{-(h)}) > f(x^{(h)})$, но не выполняется некоторые ограничения задачи (4). В этом направлении центр определяет величину шага и рассчитывает $x^{(h+1)} = x^{(h)} + \Delta^{(h)}$ и остальные компоненты вектора $u^{(h+1)}$. Результаты расчета сообщаются агентам. Агенты рассчитывают свои варианты планов и формируют предложения по модернизации и интенсификации технологического процесса. Эти данные диагностируются центром и используются им для пополнения множеств

$X', X'', S', S'', G', G''$, а множество O_h^o переходит в $O_{h+1}^o \cup o(x^{(h+1)})$.

Скорректированное множество O_{h+1}^o служит основанием для организации институционального, стимулирующего и информационного управления центром.

3. Поведение центра

Итеративный процесс планирования может быть представлен следующей схемой:

1. центр по исходной и дополнительной информации, сообщенной на каждом h -м шаге k -м агентом, рассчитывает вариант производственной программы $x^{(h)} = \{x_k^{(h)}\}$ и планируемый размер

фонда материального поощрения $c^{(h)} = \{c_k^{(h)}\}$ за принятие k -м агентом данного варианта в качестве плана. Центр сообщает агентам значения $\{x_k^{(h)}\}$ и $\{c_k^{(h)}\}$;

2. агент выбирает и сообщает центру наилучший для себя план $x_k^{(h)*}$. Примем, что на каждом h -м шаге выполняется гипотеза о локально-оптимальном поведении агента;

3. если $x_k^{(h)*} \neq x_k^{(h)}$, то агент по запросу центра сообщает дополнительную информацию, т.е. осуществляется переход к первому шагу данной схемы. В случае совпадения $x_k^{(h)*} = x_k^{(h)}$, процедура планирования заканчивается.

Алгоритм определения плана будет зависеть от степени информированности центра. Рассмотрим три его варианта.

Вариант 1. Полная информированность центра о локальных ограничениях агентов и их целевых функциях. Таким образом, центр имеет исходную информацию:

$$f(x), G(x), b, X, X_k, \max_{x_k^* \in X_k} E\varphi_k(x_k^*).$$

Этому варианту можно сопоставить следующий алгоритм определения плана.

1. На основании исходной информации центр может рассчитать план либо путем решения задачи ОП ($x^{(h)} = x^o$), либо задачи ОСП ($x^{(h)} = x^{oc}$). Т.к. рассматриваемая процедура является итеративной, то центр на h -м шаге процедуры планирования имеет вектор значений планов, выбранных агентами за предыдущие $(h-1)$ шагов: $\{x^{*(1)}, x^{*(2)}, \dots, x^{*(h-1)}\}$, где $x^{*(i)} = \{x_k^{*(i)}\}$.

2. На каждом шаге центр по дополнительной информации корректирует

предлагаемое агентом значение плана производственной программы и размера поощрения:

$$x_k^{(h)} = x_k^{(h-1)} + \beta \left(x_k^{(h-1)*} - x_k^{(h-1)} \right),$$

$$c_k^{(h)} = c_k^{(h-1)} + \alpha \left[\max_{x_k^* \in X_k} E\varphi_k(x_k^{(h)*}) - E\varphi_k(x_k^{(h-1)}) \right],$$

где α и β – коэффициенты акселерации, выбираемые из соображений, имеющих, как правило, неформальный характер, хотя можно определить правило их изменения.

3. На каждом шаге агент выбирает наилучшую стратегию путем

$$\max \left\{ \max_{x_k^* \in X_k} E\varphi_k(x_k^{(h)*}), E\varphi_k(x_k^{(h)}) \right\} \quad \text{и}$$

принимает один из вариантов плана: $x_k^{(h)*}$ или $x_k^{(h)}$.

4. Если $x_k^{(h)*} \neq x_k^{(h)}$, то процедура повторяется с п. 2, в случае $x_k^{(h)*} = x_k^{(h)}$ процедура заканчивается утверждением плана.

Сходимость предложенной схемы зависит от вида решаемой задачи.

1. Решается задача ОП. Пусть $x_k^{(h)} \notin S_k$, а также $X \cap S \neq \emptyset$, тогда

$$E\varphi_k(x_k^{(h)}) < \max_{x_k^* \in X_k} E\varphi_k(x_k^{(h)*}).$$

Так как согласно гипотезе благожелательности агента к центру выполняется условие

$$\max \left\{ \max_{x_k^* \in X_k} E\varphi_k(x_k^{(h)*}), E\varphi_k(x_k^{(h)}) \right\} =$$

$$= \max_{x_k^* \in X_k} E\varphi_k(x_k^{(h)*})$$

то $x_k^{(h)*} \neq x_k^{(h)}$ и требуемое число шагов процедуры планирования $h > 1$.

Сходимость процедуры будет обеспечена путем корректировки величины $c^{(h)}$ по правилу:

$$c^{(h)}(x) = c^{(h-1)}(x) + \gamma(c^{(h-1)}(x^{(h)*}) - c^{(h-1)}(x^{(h)})), \quad \text{где}$$

$\gamma > 0$, что означает уступку выигрыша центра в пользу агента для обеспечения $x \in S$.

2. Пусть $x_k^{(h)} \in S_k$ и $X \cap S = \emptyset$ (центр не учитывает интересы агентов). Рассматриваемая процедура планирования не сходится к $x_k^{(h)}$. Доказательство очевидно.

3. Если $x_k^{(h)} \in S_i$ и $X \cap S \neq \emptyset$, то $E\varphi_k(x_k^{(h)}) \geq \max_{x_k^* \in X_i} E\varphi_k(x_k^{(h)*})$, тогда из условия благожелательности агента к центру следует, что

$$\max \left\{ \max_{x_k^* \in X_k} E\varphi_k(x_k^{(h)*}), E\varphi_k(x_k^{(h)}) \right\} =$$

$$= \max_{x_k^* \in X_k} E\varphi_k(x_k^{(h)*})$$

Следовательно, $x_k^{(h)*} = x_k^{(h)}$ и процедура планирования сходится за один шаг, т.е. $h = 1$.

Вариант 2. Полная информированность центра о целевых функциях агента и неполной информированности о их локальных ограничениях, т.е. о множествах X_k . Т.о. центр имеет исходную информацию: $f(x), G(x), b, \max_{x_k^* \in X_k} E\varphi_k(x_k^*)$.

В этом случае можно рассмотреть следующую итеративную процедуру, использующую первую схему блочного программирования Полтеровича.

1. На h -м шаге центр получает дополнительную информацию от агентов о предлагаемых ими напряженных планах $x^{(h)*} = \{x^{(h)*1}, x^{(h)*2}, \dots, x^{(h)*n}\}$. Центр, используя набор $\{x^{(1)*}, x^{(2)*}, \dots, x^{(h)*}\}$ аппроксимирует множество X , используя гипотезу о его выпуклости:

$$x^{(h)} = \left\{ x/x = \sum_{j=1}^h \alpha_j x^{(j)*}, \alpha_j \geq 0, \sum_{j=1}^h \alpha_j = 1 \right\}$$

2. Центр решает на каждом шаге либо задачу ОП: $f(x) \rightarrow \max, G(x) \leq b, x \in X^{(h)}$, либо задачу ОСП, т.е. задачу ОП при дополнительных ограничениях, причем переменными являются как x_k , так и $c_k, k = 1, m$

$$E\varphi_k(x_k^{(h)}) + c_k^{(h)} \geq \max_{x_k^* \in X_k} E\varphi_k(x_k^{(h)*}), \quad (5)$$

$$c(x) - \sum_{k=1}^m c_i - \sum_{j=2}^h \sum_{k=1}^n \delta_k^{(j)} \geq 0$$

где $\delta_k^{(j)}$ – планируемый размер дополнительного фонда материального поощрения, назначаемый центром агентам на h -м шаге вместе с планом $x_k^{(j)}$. Его величина определяется как $\delta_k^{(j)} = \langle p_k^{(j-1)}, x_k^{(j-1)} \rangle$, где

$$p_k^{(j-1)} = \alpha_k^{(j-1)} \text{grad}_x F(x_k^{(j-1)}, y_k^{(j-1)}), k = \overline{1, m}$$

, $\text{grad}_x F(\bullet)$ – градиент функции Лагранжа, которая для задачи ОСП имеет вид:

$$F(x, y) = f(x) - \sum_{l=1}^n y_l [g_l(x) - b_l] + \sum_{k=1}^m y_{n+k} \left[c_i - \max_{x_k^* \in X_k} E\varphi_k(x_k^*) \right] + y_{n+m+1} \left[c(x) - \sum_{k=1}^m c_i - \sum_{j=2}^h \sum_{k=1}^m \delta_k^{(j)} \right]$$

Для задачи ОП функция Лагранжа имеет тот же вид, но $y_{m+n+1} = 0$. Путем решения этой задачи центр получает значения $x^{(h)}, c^{(h)}, \delta^{(h)}$ и

$$p^{(h)} = \text{grad}_x F(x^{(h)}, y^{(h)}), \text{ где } (x^{(h)}, y^{(h)}) -$$

седловая точка функции Лагранжа на множестве $X^{(h)}$. $\alpha^{(h)} > 0$ подбирается из условия обеспечения расходов центра на стимулирование агентов за счет роста величины $c(x)$. Причем предполагается, что $\langle \text{grad}_x c(x^{(h)}), x^{(h)} - x^{(h-1)} \rangle - (c^{(h)} - c^{(h-1)}) > 0$, т.е. на каждом шаге фонда поощрения должно хватать для стимулирования агентов за выбор напряженного плана. Так как

$$\langle \alpha^{(h)} \text{grad}_x F(x^{(h)}, y^{(h)}), x^{(h)} - x^{(h-1)} \rangle + (c^{(h)} - c^{(h-1)}) \leq \langle \text{grad}_x c(x^{(h)}), x^{(h)} - x^{(h-1)} \rangle$$

то

$$\alpha^{(h)} \leq \frac{\langle \text{grad}_x F(x^{(h)}, y^{(h)}), \Delta x^{(h)} \rangle}{\langle \text{grad}_x c(x^{(h)}), \Delta x^{(h)} \rangle - (c^{(h)} - c^{(h-1)})},$$

где $\Delta x^{(h)} = x^{(h)} - x^{(h-1)}$.

Значения $x^{(h)}, c^{(h)}, \delta^{(h)}, p^{(h)}$ центр сообщает агентам.

3. По своей целевой функции

$$E\varphi_i(x_k^{(h)*}) + \begin{cases} c_k^{(h)}(x_k^{(h)}) + \sum_{j=2}^{h-1} \delta_k^{(j)} + \delta_k^{(h)} > 0 \\ 0 + \delta_k^{(h)} < 0 \end{cases}$$

где для первого слагаемого в фигурных скобках должно выполняться условие

$x_k^{(h)*} = x_k^{(h)}$ и $x_k^{(h)*} \in L^k$, а для второго $x_k^{(h)*} \neq x_k^{(h)}$ и $x_k^{(h)*} \in X_k^{(h)}$; агенты в соответствии с гипотезой о локально-оптимальном поведении выбирают наилучшую для себя ситуацию:

$$\max \{ \max_{x_k \in X_k} E\varphi_k(x_k^{(h)*}); E\varphi_i(x_k^{(h)*}) + c_k^{(h)} + \sum_{j=2}^h \delta_k^{(j)}; E\varphi_k(x_k^{(h)*}) + c_k^{(h)} + \sum_{j=1}^h \delta_k^{(j)} + \langle p_k^{(h)}, x_k^{(h)*} - x_k^{(h)} \rangle \}$$

Для задачи ОП

$$L^{(h)} = \{x / G(x^{(h)}) \geq b, f(x^{(h)}) > f(x^{(h)*})\},$$

а для задачи ОСП

$$L^{(h)} = \{x / G(x^{(h)}) \geq b, E\varphi_k(x_k) + c_k \geq$$

$$\geq \max_{x_k \in X_k} E\varphi_k(x_k^{(h)*}), (c(x^{(h)}) - \sum_{i=1}^n c_i^{(h)}) -$$

$$- \sum_{j=2}^h \sum_{k=1}^m \delta_k^{(j)} \geq 0, f(x^{(h)}) > f(x^{(h)*})\}$$

здесь $x^{(h)*}$ – план, выбираемый агентом на h -м шаге, $x^{(h)*} \in L^{(h)}$ – напряженный план, предлагаемый агентом центру на h шаге;

$$\begin{cases} \delta_k^{(h)} > 0 \text{ при } x_k^{(h)*} \in L^{(h)} \\ \delta_k^{(h)} < 0 \text{ при } x_k^{(h)*} \in X^{(h)} \end{cases}$$

4. Агенты сообщают центру свои наилучшие стратегии. Если

$x_k^{(h)*} \neq x_k^{(h)}, x_k^{(h)} \in X_k^{(h)}$, то центр корректирует свое решение по схеме, описанной в п.2. Если

$x_k^{(h)} \in L_k^{(h)}$, то центр обновляет информацию,

расширяет множество $X^{(h)} \subset X^{(h+1)}$ и вычисляет $x^{(h+1)}, c^{(h+1)}, \delta^{(h+1)}, p^{(h+1)}$. Если $x_k^{(h)*} = x_k^{(h)}$,

то это означает, что

$$E\varphi_k(x_k^{(h)}) + c_k^{(h)} + \sum_{j=2}^h \delta_k^{(j)} \geq E\varphi_k(x_k^{(h)*}) + c_i^{(h+1)} + \sum_{j=1}^h \delta_k^{(j)} + \langle p_k^{(h)}, x_k^{(h)*} - x_k^{(h)} \rangle$$

Так как

$$E\varphi_k(x_k^{(h)*}) + c_k^{(h+1)} \geq E\varphi_k(x_k^{(h)*}) + c_k^{(h)} \geq \max_{x_k^* \in X_k} E\varphi_k(x_k^{(h)*})$$

то $\langle p_k^{(h)}, x_k^{(h)*} - x_k^{(h)} \rangle \leq 0$ для всех k -х агентов,

$$x_k^{(h)} \neq x_k^{(h+1)}, \text{grad } x^c(x_k^{(h)}) \neq 0,$$

т.е. . Поэтому

$$\text{grad } x^c(x_k^{(h)}, y_k^{(h)}) \neq 0$$

$$\alpha_k^{(h)} \neq 0 \quad \text{и}$$

$$\langle \text{grad } x^c(x_k^{(h)}, y_k^{(h)}), x_k^{(h)*} - x_k^{(h)} \rangle \leq 0.$$

Следовательно, точка $(x_k^{(h)}, y_k^{(h)})$ оказывается седловой точкой Лагранжа в задаче ОП или ОСП, и значение $x_k^{(h)*} = x_k^{(h)}$ является соответственно оптимальным планом или оптимальным согласованным планом. Интерактивная процедура заканчивается.

Предположим, что условия образования фонда материального стимулирования таковы, что для любого $x \in X$ выполняется неравенство $|\text{grad } x^c(x)| > 0$, где $|\bullet|$ – длина вектора. Тогда рост фондов материального поощрения больше расходов на расширение множества $X^{(h)}$, поэтому в этом случае допустимое множество

$$\begin{aligned} & \{x/G(x^{(h)}) \geq b; c(x^{(h)}) - \sum_{k=1}^m c_k^{(h)} - \\ & - \sum_{j=2}^h \sum_{k=1}^m \delta_k^{(j)} \geq 0; E\varphi_k(x_k^{(h)*}) + c_k^{(h)} \geq \} \quad (6) \\ & \geq \max_{x_k^{(h)*} \in X_k^{(h)}} E\varphi_k(x_k^{(h)*}) \cap X^{(h)} \end{aligned}$$

с ростом множества $X^{(h)}$ монотонно не сужается.

Достаточным условием сходимости описанного итеративного процесса является выполнение условия (3). Поскольку $x_k^{(h)} \in S_k$, то

$$E\varphi_k(x_k^{(h)}) + c_k^{(h)} \geq \max_{x_k^* \in X_k} E\varphi_k(x_k^{(h)*}). \quad \text{Из}$$

соотношения

$$c_k^{(h+1)} = c_k^{(h)} + \langle \text{grad } x^c(x_k^{(h+1)}), x_k^{(h+1)} - x_k^{(h)} \rangle$$

следует неравенство

$$E\varphi_k(x_k^{(h)*}) + c_k^{(h+1)} \geq E\varphi_k(x_k^{(h)}) + c_k^{(h)} \geq \max_{x_k^* \in X_k} f_i(x_k^{(h)*})$$

для

$x_k^{(h)*} \in L^{(h)}$. Согласно условию алгоритма блочного программирования

$$\langle p_k^{(h)}, x_k^{(h)*} - x_k^{(h)} \rangle > 0 \quad \text{для всех } x_k^{(h)*} \in L^{(h)}.$$

Отсюда следует, что

$$\begin{aligned} & E\varphi_k(x_k^{(h)*}) + c_k^{(h+1)} + \sum_{j=2}^{h+1} \delta_k^{(j)} + \\ & + \langle p_k^{(h)}, x_k^{(h)*} - x_k^{(h)} \rangle > E\varphi_k(x_k^{(h)}) + c_k^{(h)} + \\ & + \sum_{j=2}^h \delta_k^{(j)} \end{aligned}$$

Поэтому при решении задачи ОСП агент всегда будет выбирать $x_k^{(h)*} \in L^{(h)}$

Следовательно, на каждом h -м шаге агенты формируют предложения для $(h+1)$ -го шага в виде значения напряженного плана $x_k^{(h)*} \in L^{(h)}$, который они могли бы реализовать. В силу условий (5) и (6) новое множество $X^{(h+1)}$ будет расширяться по сравнению с множеством $X^{(h)}$. При этом будут выполняться ограничения по фонду материального поощрения и условия совершенного согласования.

Поскольку $X^{(1)} \subset \dots \subset X^{(h)} \subset \dots \subset X$, то допустимые множества в задаче ОСП не убывают. За счет расширения допустимого множества значений $f(x^{(h)})$ для каждой задачи ОСП будет только возрастать. Поэтому последовательность $f(x^{(1)}), f(x^{(2)}), \dots, f(x^{(h)}), \dots$ монотонно не убывает и в силу компактности множества X имеет предел f^* , т.е. $\lim_{h \rightarrow \infty} f(x^{(h)}) = f^*$ и $f^* = f(x^{OC})$.

Можно показать, что если решается задача ОП и нарушено условие совершенного согласования, то $x_k^{(h)*} \neq x_k^{(h)}$ и $x_k^{(h)*} \notin L^{(h)}$. Следовательно,

$x_k^{(h)*} = x_k^{(h+1)}$ и множество $X^{(h)}$ не расширяется в силу отсутствия у агентов стимулов отыскивать резервы для принятия напряженных планов и процедура итеративного планирования закончится не достигнув x^0 .

Вариант 3. Полная информированность центра об ограничениях и неполной информированности о целевых функциях $f(x), E\varphi_k(x_k)$. Центр имеет исходную информацию: $G(x), b, X, X_k$. То есть

центр имеет полную математическую модель системы и ему не надо затрачивать усилия на идентификацию технологического множества системы. Но центру не известны мотивы и

поведение агентов при выборе ими плановых заданий.

В этом случае можно рассмотреть любой итерационный алгоритм, например, использующий релаксационную схему условного градиента.

1. Центр на каждом шаге получает дополнительную информацию:

$$\text{grad}_{x_k} E\varphi_k(x_k^{(h-1)*}), \text{grad}_{x_k} f(x_k^{(h-1)*}), x_k^{(h-1)*} \\ \text{и } c_k^{(h-1)*}$$

2. Центр на каждом h -м шаге в точке $x_k^{(h+1)*}$ выбирает направление поиска на основе решения задачи ОП

$$\langle \text{grad}_{x_k} f(x_k^{(h-1)*}), x_k - x_k^{(h-1)*} \rangle \rightarrow \max$$

или

$$G(x) \geq b, x \in X^{(h-1)}$$

задачи ОСП с учетом условий согласования и ограничения по фонду материального поощрения

$$\sum_{k=1}^m c_k \leq c(x).$$

В последнем случае в задачу оптимизации вводится новая переменная $c_k > 0$. После этого центр сообщает агентам значение плана $x_k^{(h)}$ и размер планируемого поощрения $c_k^{(h)}$.

3. Агент по направлению поиска на каждом шаге выбирает наилучший для себя шаг $\beta_k^{(h)}$, решая в соответствии с гипотезой о локально-оптимальном поведении задачу:

$$E\varphi_k \left[x_k^{(h)*} + \beta_k \left(x_k^{(h)*} - x_k^{(h-1)*} \right) \right] + \\ + \left[c_k^{(h)*} + \beta_k \left(c_k^{(h)*} - c_k^{(h-1)*} \right) \right] \rightarrow \max_{\beta_k \in [0,1]}$$

4. По величине шага агент определяет план $x_k^{(h)*}$, который он обязан выполнить за поощрение $c_k^{(h)*}$, т.е.

$$x_k^{(h)*} = x_k^{(h-1)*} + \beta_k^{(h)} \left(x_k^{(h)} - x_k^{(h-1)*} \right), c_i^{*k} = \\ = c_i^{*(k-1)} + \beta_i^k \left(c_k^{(h)} - c_k^{(h-1)*} \right)$$

и сообщает его величину центру.

5. Если $x_k^{(h)*} \approx x_k^{(h)}$, то центр заканчивает процесс планирования, в противном случае центр запрашивает у агентов дополнительную информацию, т.е. осуществляет переход к п.1 данной схемы.

4. Поведение агента

Как было показано выше, при разработке производственной программы центр использует приближенное описание множества X производственных возможностей агентов. Агент, обладая более детализированной информацией, может при создании центром более привлекательных условий u_k путем решения задачи

(2) определять для себя такой вектор $x_k^{(h)}$, который

позволяет расширить множество $X_k^{(h)}$ представлений центра о его возможностях, h – шаг итерационного процесса. В данном случае вектор $x_k^{(h)}$ можно рассматривать как вектор способов действия.

Обозначим через $\omega_k = \{\omega_k^{(h)}, h = \overline{1, H}\} \in A_k$ – вектор параметров состояния, определяющий значения вектора действий

$$x_k^{(h)} = \left\{ x_{kj}^{(h)}, j \in \overline{[1, m_k]} \right\} \in X_k^{(h)}.$$

Можно считать, что этот вектор описывает знание агента возможностей контролируемого им объекта управления. Здесь A_k – множество возможных значений вектора состояния. Будем считать, что агент обладает способностями, знаниями, которые гарантируют существование $\Psi_k : A_k \rightarrow X_k^{(h)}$.

Конструктивные возможности технологических узлов и доступный агенту уровень знаний делают справедливым предположение о существовании для k -го агента предельного множества параметров состояния. Обозначим через

$O_k^* = \{o_k^* / o_k(x_k), x_k \in X_k^{(h)}(\omega_k^{(h)}), \omega_k^{(h)} \in A_k\}$ – множество достижимости или множество предельных возможностей.

Будем предполагать, что агент за счет своих креативных способностей, способности к самообучению и поиску новой информации при соответствующем стимулирующем воздействии центра способен определять такие состояния

$\omega_k^{(1)} \in A_k$ и $\omega_k^{(2)} \in A_k$, что возможно $\omega_k^{(2)} \succ \omega_k^{(1)}$, где символ \succ означает «более

значимо» и при этом $X_k^{(1)}(\omega_k^{(1)}) \subseteq X_k^{(2)}(\omega_k^{(2)})$.

Следовательно, существует такая последовательность $\omega_k^{(1)}, \omega_k^{(2)}, \omega_k^{(3)}, \dots$, что

$$\lim_{h \rightarrow \infty} o_k^{(h)}(x_k^{(h)}(\omega_k^{(h)})) = O_k^*.$$

То есть агент путем изучения объекта управления способен определить его предельные возможности для

достижения желаемого для себя состояния. Последовательность $\omega_k^{(1)}, \omega_k^{(2)}, \omega_k^{(3)}, \dots$ будем называть последовательностью вскрытия резервов технологического узла за счет лучшего обслуживания и управления им агентом.

Такая способность агента формировать расширяющееся множество способов действия позволяет определить следующие свойства целевой функции агента и областей достижимости:

$$\forall \omega_k^{(1)}, \omega_k^{(2)} \in A_k, \omega_k^{(2)} \succ \omega_k^{(1)}, X_k^{(1)}(\omega_k^{(1)}) \subseteq X_k^{(2)}(\omega_k^{(2)}) \mapsto E\varphi_k(x_k^{(2)}) > E\varphi_k(x_k^{(1)}) \quad (7)$$

Условие (7) означает, что поведение агента при выполнении принципа рациональности соответствует закону повышающихся потребностей, который в литературе по психологии поведения определяет мотивированность и целеустремленность агента.

В реальных условиях агент при превышении некоторого порога значимости $\Delta = E\varphi_k(x_k^{(2)}) - E\varphi_k(x_k^{(1)})$ изменения ценности ситуации целеустремленного состояния по результату способен идентифицировать предпочтительные способы действия и видит открывающиеся возможности при изменении структуры своей информированности (знания). Выработка решения при таком подходе заключается в реализации совокупности последовательных процедур, предназначенных для поиска промежуточных решений, на основании которых агент уточняет свои возможности и формирует окончательное решение. Полный цикл его формирования k -м агентом состоит в выполнении следующих шагов на этапе h :

1. Формирование множеств A_k и $X_k^{(h)}$ на основе знаний, опыта, интуиции и располагаемой информацией о параметрах состояния $\omega_k^{(h)}$.

Просмотр множества A_k и формирование точки

$$O_k^{(h)*} = \left\{ \begin{array}{l} o_k^{(h)*} / o_k^{(h)*}(x_k^{(h)}), \\ x_k^{(h)} \in X_k^{(h)}(\omega_k^{(h)}), \omega_k^{(h)} \in A_k \end{array} \right\}.$$

Проверка, существует ли $x_k^{(h)*}$ такое, что $o_k^{(h)}(x_k^{(h)*}) = o_k^{(h)*}$. Если – да, то $x_k^{(h)*}$ – это компромиссное решение, а $o_k^{(h)*}$ – прогнозируемая ситуация, в противном случае переход к п.2.

2. Решение задачи поиска потенциально-предпочтительного набора действий $x_k^{(h)*} \in X_k^{(h)}(\omega_k^{(h)})$, позволяющего

сформировать вектор $o_k^{-(h)}$ предельных значений

критериев при использовании имеющегося на данный момент знания о правиле $\Psi_k^{(h)}$ и структуре

множества A_k . Так как компоненты $o_{ki}^{-(h)}, i = \overline{1, N}$

порожны достижимы, а вместе – нет, то делается попытка найти компромиссное решение. Если агент не согласен попытаться найти компромиссное решение за счет компенсаторных уступок по каждому критерию, которые несколько хуже решения $o_k^{-(h)}$, то переход к п. 3, иначе к п. 5.

3. Исследование направлений возможного расширения множества A , организация процедур

поиска новой информации (знания) о $\omega_k^{(h)} \in A_k$ и

правиле $\Psi_k^{(h)} : A_k \rightarrow X_k^{(h)}$.

4. Если расширение множества A_k возможно, то переход к п. 1, иначе фиксация ситуации, что компромиссное решение не может быть найдено при выбранном векторе $o_k^{(h)*}$.

5. Получение сведений от агента достаточных для определения вектора $o_k^{(h)} \prec o_k^{(h)*}$, где $o_k^{(h)}$ минимальные требования агента к принимаемым им во внимание результатам.

6. Выполнение процедуры поиска минимально-предпочтительной точки в пространстве критериев по направлению предпочтения $o_k^{(h)}, o_k^{(h)*}$,

определение вектора $o_k^{(h)*} \in A_k$ и

$x_k^{(h)*} \in X_k^{(h)}(\omega_k^{(h)*})$ – минимального значения плановых показателей, соответствующих значениям компонент вектора $o_k^{(h)}$.

7. Если полученные значения для $x_k^{(h)*}, o_k^{(h)*}$ принимаются как компромиссное решение, то процедура останавливается, в противном случае переход к п. 8.

8. Для ограничений на $o_k^{(h)}$ определяется приоритетная координата $i \in [\overline{1, N}_k]$, по которой

делается расширение множеств A_k и $X_k^{(h)}$, так чтобы $o_k^{(h)}(x_k^{(h)}) = o_k^{(h)} + \Delta_k^{(h)}$, где $\Delta_k^{(h)}$ минимально возможное улучшение, которое является значимым для агента и определяется по его высказываниям о “гибкости” ограничения на основе выполнения процедур поиска дополнительной информации. Переход к п. 1.

Легко видеть, что в данном случае возможно информационное управление со стороны центра. Оно состоит в предоставлении управляемому субъекту определенной информации (информационная картина), ориентируясь на которую агент, имея возможность доступа к собственным источникам информации, выбирает линию своего поведения при формировании встречной информации.

Заключение

1. В процессе определения оптимального плана возможны две ситуации. Первая связана с тем, что оптимальное решение $x^* \in X_k^{(h)}$, тогда интерактивное взаимодействие центра и агента должно быть направлено на локализацию решения. Если же x^* находится на границе или вне множества $X_k^{(h)}$, то это говорит о том, что центр не обладает представлениями о предельных возможностях агентов и должен предпринимать усилия для получения информации по расширению множества $X_k^{(h)}$, для решения задачи вида

$$f^{(h+1)}(x) \rightarrow \max / x \in X_k^{(h+1)},$$

где множество допустимых решений шире, чем на $X_k^{(h)}$ и выполняется условие

$f^{(h)}(x) < f^{(h+1)}(x)$, что определяет мотивированность центра в расширении представлений о множестве $X_k^{(h)}$.

Такой прием будем называть расширением представлений центра за счет получения информации от агентов об их возможностях.

2. Для случая полной информированности центра показано, что решение задачи ОСП будет реализовано агентами, если найдено стимулирующее воздействие $c_k^{(h)}$, при котором

$$E\varphi_k \left(x_k^{(h)} \right) + c_k^{(h)} \geq \max_{x_k^* \in X_k} E\varphi_k \left(x_k^{(h)*} \right).$$

3. Для случая неполной информированности центра о множестве X_k сходимость к оптимальному решению будет обеспечена путем дополнительного стимулирования агентов за более полное раскрытие ими своих возможностей. Приведена итеративная процедура, использующая первую схему алгоритма блочного программирования Полтеровича. Показано, что достаточным условием ее сходимости будет условие (3).

Библиографический список

- [Полтерович, 1969] Полтерович, В.М. Блочные методы выпуклого программирования и их экономическая интерпретация / В.М. Полтерович // Экономика и математические методы. 1969, т. V, № 6
- [Виноградов, 2011] Виноградов, Г.П. Моделирование поведения агента с учетом субъективных представлений о ситуации выбора / Г.П. Виноградов, В.Н. Кузнецов // Искусственный интеллект и принятие решений. № 3. 2011. с. 58-72.
- [Vinogradov, 2009] Vinogradov, G.P. Model of Individual Decosion-Making: Behavior of Intellectual Agent / G.P. Vinogradov, B. V. Paluch // Interactive Systems and Technologies: the Problems of Human-Computer Interaction. Volume III. – Collection of scientific papers. – Ulyanovsk: UISTU, 2009. P. 127-134.
- [Чхартишвили, 2004] Чхартишвили, А.Г. Теоретико-игровые модели информационного управления / А.Г. Чхартишвили М.: ЗАО «ПМСОФТ», 2004.

ITERATIVE METHODS OF OPTIMIZATION WITH COORDINATED IDENTIFICATION ELEMENOTOV CHOICE MODELS

Vinogradov, GP Kuznetsov, VN

Tver State Technical University, Tver, Russia

WGP272NG@mail.ru

The convergence of interactive procedures agreed in the optimization problem of controlling the evolution of organizational and technological systems in different variants of the center of awareness about the possibilities of agents. The conditions under which the procedure is possible to build an interactive exchange of information between the center and its agents, ensuring the convergence of the solution of the consistent optimization.

Keywords: active system, an intelligent agent, consistent optimization, decision making



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИЧЕСКОЕ ОПРЕДЕЛЕНИЕ ТИПА АГЕНТА В МНОГОАГЕНТНОЙ СИСТЕМЕ. ПРОБЛЕМА МЕЖАГЕНТНОГО ВЗАИМОДЕЙСТВИЯ

Мутовкина Н.Ю., Ключин А.Ю., Кузнецов В.Н.

Тверской государственный технический университет,

г. Тверь, Россия

Mutovkina_N@mail.ru

Klalex@inbox.ru

Bumhouse@mail.ru

В работе рассматриваются подходы к определению типа интеллектуальных агентов и состояния многоагентной системы, основанные на теории нечетких множеств и нечеткой логике. Описываемые в работе понятия и методы иллюстрируются на примерах определения типа агентов и приведения многоагентной системы в состояние компромисса. Также в статье сформулированы производные от названных задачи, актуальные при разработке моделей межагентного взаимодействия.

Ключевые слова: система многоагентная; логика нечеткая; взаимодействие агентов; состояние агентов.

ВВЕДЕНИЕ

Необходимым условием для разработки моделей и методов согласованного управления в многоагентных системах (МАС) является определение ключевых характеристик элементов МАС – интеллектуальных агентов и возможных вариантов их взаимодействия. Взаимодействие необходимо для распределения задач между агентами и их совместного решения. При этом одним из базисных понятий является антропоморфное определение агента, в котором ведущее место занимают ментальные свойства. [Кузнецов и др., 2012] Установлено, что любой интеллектуальный агент (ИНАГ), имеющий определенные цели и предпочтения, нуждается в ресурсах для их достижения и демонстрирует при этом некоторое поведение, т.е. обладает встроенными механизмами мотивации или механизмами формирования предпочтений. Это обстоятельство следует учитывать при выборе метода управления в МАС с тем, чтобы воздействие на ИНАГов в итоге обеспечило максимум запланированных результатов функционирования МАС. Тогда управление можно считать эффективным.

Вопросы коммуникации и кооперации агентов рассматривались в работах Варшавского В.И., Гермейера Ю.Б., Городецкого В.И., Новикова Д.А., Поспелова Д.А., Рыбиной Г.В., Тарасова В.Б.,

Чхартишвили А.Г., Акоффа Р., Брукса Р., Майерсона Р.Б., Хьюитта К., Шоэма И., Эмери Ф. и других.

Коллективное поведение ИНАГов образуется на основе индивидуальных интеллектуальных поведений. При этом предполагается согласование целей, интересов и действий различных агентов, разрешение конфликтов путем переговоров и переадресации задач. Успешность коммуникации и кооперации агентов зависит от того, насколько дружелюбным является климат в МАС и в какой степени ИНАГи готовы пойти на уступки. Поэтому особенно важным для выполнения условий коммуникации и кооперации ИНАГов представляется определение и, при необходимости, изменение поведенческого типа агента. В определенной ситуации каждый ИНАГ может быть отнесен к одному из трех поведенческих типов: уклоняющийся, принуждающий или компромиссный (сотрудничающий).

Целью данного исследования является разработка методики определения поведенческого типа ИНАГа, отражающего его предпочтения, и изменения установленного типа (при необходимости) посредством управляющих воздействий, с использованием теории нечетких множеств и нечеткой логики. [Беллман, 1976] Влияние на состояние ИНАГов, входящих в состав МАС, определяет состояние самой системы, поэтому важнейшим аспектом исследования является изучение и построение моделей взаимодействия ИНАГов друг с другом,

окружающей средой и управляющим элементом. [Городецкий и др., 2009], [Новиков и др., 2011], [Рыбина, 2008] В роли управляющего элемента может выступать ИНАГ-Центр или пользователь МАС.

1. Постановка и метод решения задачи

Поведение ИНАГа в МАС выражается его переходами из состояния в состояние: $s_1 \rightarrow \dots \rightarrow s_{t-1} \rightarrow s_t \rightarrow s_{t+1} \rightarrow \dots \rightarrow s_T$, где s_{t-1} – состояние агента, предшествующее моменту времени t , s_T – состояние достижения цели, T – число итераций, за которое цель будет достигнута. В общем виде состояние ИНАГа можно представить как совокупность следующих понятий: X_1 – знания, X_2 – убеждения, X_3 – желания, X_4 – намерения, X_5 – цели, X_6 – обязательства, X_7 – мобильность, X_8 – благожелательность, X_9 – правдивость, X_{10} – рациональность.

ИНАГ владеет определенными знаниями о себе, других агентах и окружающей среде. Знания не меняются с течением времени, а могут лишь дополняться. На основе этих знаний формируются некоторые убеждения относительно той или иной ситуации, в которую попадает агент, т.е. определенный «взгляд» на проблему. Убеждения (веру) можно интерпретировать как субъективное восприятие знаний, которое может со временем меняться. Причинами этого может быть результат, полученный от применения агентом объективных знаний (информации) на практике или из опыта других агентов. Желания представляют собой состояния, ситуации, достижение которых по разным причинам является для ИНАГа самоцелью, однако они могут быть противоречивыми и потому агент не может ожидать, что все они будут достигнуты. Намерения – это то, что агент или обязан сделать в силу своих обязательств по отношению к другим агентам (ему поручено решение задачи и он взял на себя обязательство по ее решению), или то, что вытекает из его желаний (т.е. непротиворечивое подмножество желаний, выбранное по тем или иным причинам, и которое совместимо с принятыми на себя обязательствами). Целями называется конкретное множество конечных и промежуточных состояний, достижение которых ИНАГ принял в качестве текущей стратегии поведения. Обязательствами по отношению к другим агентам являются задачи, которые агент берет на себя по просьбе (поручению) других агентов в рамках кооперативных целей или целей отдельных агентов в рамках сотрудничества. Мобильность – это способность агента перемещаться в МАС и вне ее (по сети Internet) в поисках необходимой информации для решения своих задач, при кооперативном решении задач совместно или с помощью других агентов и т.д. Важным условием кооперации и сотрудничества

ИНАГов является свойство благожелательности, т.е. готовности агентов помочь друг другу и решать именно те задачи, которые поручает пользователь конкретному ИНАGu, что предполагает отсутствие у агента конфликтующих целей. Не менее важными свойствами ИНАГов являются правдивость и рациональность. Правдивость агента заключается в том, что он не манипулирует информацией, про которую ему заведомо известно, что она ложная. Рациональность – свойство агента действовать так, чтобы достигнуть своих целей, а не уклоняться от их достижения, по крайней мере, в рамках своих знаний и убеждений. [Городецкий, 1998]

Все перечисленные понятия можно интерпретировать как нечеткие множества. [Беллман, 1976] Например, в момент времени $t=1$ ИНАГ имеет знания (x_{1t}) в объеме 50% от необходимых для решения задачи 100%, в момент времени $t=2$ объем знаний увеличивается уже до 65% и т.д. Значение 50% может интерпретироваться как «недостаточное» для решения задачи и иметь оценки в интервале (0; 0,5), 65% – как «недостаточное» или «среднее» и иметь оценки в интервалах (0; 0,5) и (0,5; 0,8) соответственно. Значения из интервалов являются степенями принадлежности элемента x_{1t} к нечеткому множеству X_1 . Еще пример: в момент времени $t=1$ агент готов взять на себя обязательства по решению какой-либо задачи с готовностью 37%, а в следующий момент эта готовность может увеличиться до 60%. При $t=3$ в силу влияния экзогенных и эндогенных факторов, например, под воздействием мнения других агентов или из-за изменения постановки задачи пользователем готовность ИНАГа взять на себя обязательства по решению задачи может снизиться до 52% и т.д. Таким образом, значения 37%, 60%, 52% также могут быть оценены в лингвистической шкале «не готов», «скорее не готов, чем готов», «скорее готов, чем не готов», «готов» и выражены числами в интервале (0;1), которые являются степенями принадлежности элемента x_{6t} к нечеткому множеству X_6 . Следовательно, можно записать:

$$\begin{aligned} X_1 &= \left\{ \left\langle x_{1t}, \mu_{X_1}(x_{1t}) \right\rangle \mid x_{1t} \in U \right\}, \\ X_2 &= \left\{ \left\langle x_{2t}, \mu_{X_2}(x_{2t}) \right\rangle \mid x_{2t} \in U \right\}, \\ &\dots \\ X_{10} &= \left\{ \left\langle x_{10t}, \mu_{X_{10}}(x_{10t}) \right\rangle \mid x_{10t} \in U \right\}, \end{aligned} \quad (1)$$

где $\mu_{X_1}, \mu_{X_2}, \dots, \mu_{X_{10}}$ – функции принадлежности, т.е. $\mu_{X_1}, \mu_{X_2}, \dots, \mu_{X_{10}} : U \rightarrow (0;1)$ – характеристические функции множеств $X_1, X_2, \dots, X_{10} \subseteq U$, значения которых указывают, являются ли $x_{1t}, x_{2t}, \dots, x_{10t} \in U$ элементами соответствующих множеств X_1, X_2, \dots, X_{10} ; U – так называемое универсальное множество, из

элементов которого образованы все остальные множества, рассматриваемые в данном классе задач. Значения $\mu_{x_1}(x_{1t}), \mu_{x_2}(x_{2t}), \dots, \mu_{x_{10}}(x_{10t})$ называются степенями принадлежности элементов $x_{1t}, x_{2t}, \dots, x_{10t}$ к нечетким множествам X_1, X_2, \dots, X_{10} соответственно. [Беллман, 1976]

Модель состояния ИНАГа в любой момент времени можно представить в следующем виде:

$$\begin{aligned} S &= P_1 \cap P_2; \\ P_1 &= X_1 \cup X_2; \\ P_2 &= X_3 \cup \dots \cup X_{10}, \end{aligned} \quad (2)$$

где множество P_1 – это позиция агента, его точка зрения; P_2 – множество, задающее поведение агента.

Тип ИНАГа $(r, r \in R)$, отражающий его предпочтения, является оценкой его состояний, в которых он пребывает при выполнении конкретной задачи; R – множество возможных предпочтений агента. Тип агента может быть рассчитан по формуле:

$$r = \frac{\sum_{t=1}^T s_t}{T}. \quad (3)$$

Введем следующие семантические правила: агент является уклоняющимся, если $r \in (0; 0,5)$; агент называется компромиссным (сотрудничающим), если $r \in [0,5; 0,8]$ и агент будет принуждающим в случае $r \in (0,8; 1)$.

Тип ИНАГа в процессе его функционирования может меняться и рассматриваться отдельно на каждой итерации. В этом случае выполняется равенство:

$$r_t = s_t. \quad (4)$$

В [Кузнецов и др., 2012] установлено, что МАС находится в равновесном состоянии, если все агенты могут быть отнесены к компромиссному типу. Изменение типа агента осуществляется путем применения моделей и методов согласованного управления как совокупности правил принятия решений ИНАГа в виде зависимостей, ставящих каждому состоянию (типу) агента (4) конкретное значение управляющего воздействия $v_i, i \in \overline{1, N}$, $v_i \in V_t$, где N – число агентов в МАС, V_t – множество управляющих воздействий на ИНАГа в момент времени t со стороны агента-Центра или пользователя МАС. [Новиков и др., 2011] Наличие механизмов управления на каждой итерации функционирования МАС позволяет получить лучший результат и решать задачи согласования и оптимизации при взаимодействии агентов друг с другом. [Чхартишвили, 2011]

2. Пример определения типа агентов в МАС

Пусть МАС состоит из 10 ИНАГов, каждый из которых в процессе решения определенной задачи может проходить несколько итераций. Необходимо определить тип агентов на каждой итерации и, при необходимости, предпринять управляющие воздействия для его изменения, чтобы в условиях коммуникации и кооперации агентов задача была решена за 6 итераций.

Исходя из приведенной выше постановки задачи, для определения типа ИНАГа следует сформировать таблицы вида (табл. 1) на каждой итерации t .

Таблица 1 – Шаблон поведенческой таблицы

t	A_1	A_2	...	A_t	...	A_N
$\mu_{x_1}(x_{1t})$						
$\mu_{x_2}(x_{2t})$						
...						
$\mu_{x_{10}}(x_{10t})$						

Данные таблицы заполняются значениями функций принадлежности $\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_{10}}$ соответственно. Далее определяются максимумные значения следующих функций принадлежности:

$$\begin{aligned} P_{1_t} &= \mu_{x_{1_t} \cup x_{2_t}}(x_t) = \max \{ \mu_{x_{1_t}}(x_t), \mu_{x_{2_t}}(x_t) \}, \\ P_{2_t} &= \mu_{x_{3_t} \cup \dots \cup x_{10_t}}(x_t) = \max \{ \mu_{x_{3_t}}(x_t), \dots, \mu_{x_{10_t}}(x_t) \}, \\ S_t &= \mu_{P_{1_t} \cap P_{2_t}}(x_t) = \min \{ \mu_{P_{1_t}}(x_t), \mu_{P_{2_t}}(x_t) \} \end{aligned} \quad (5)$$

Если положиться лишь на возможности самоорганизации МАС, т.е. рассмотреть процесс взаимодействия ИНАГов без реализации управляющих воздействий со стороны ИНАГа-Центра (или пользователя МАС), то решение задачи будет получено, как показал эксперимент, за 11 итераций. За первые шесть итераций система перешла в состояние (табл. 2), которое выражается через типы агентов, рассчитанные по формуле (3):

Таблица 2 – Типы агентов за 6 итераций работы МАС

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
s_1	0,75	0,88	0,84	0,79	0,79	0,87	0,22	0,78	0,72	0,31
s_2	0,47	0,63	0,89	0,73	0,93	0,86	0,62	0,13	0,73	0,49
s_3	0,64	0,39	0,74	0,80	0,84	0,26	0,90	0,43	0,57	0,79
s_4	0,99	0,78	0,79	0,85	0,50	0,70	0,75	0,76	0,79	0,26
s_5	0,77	0,78	0,71	0,62	0,61	0,45	0,96	0,83	0,39	0,18
s_6	0,70	0,70	0,93	0,31	0,91	0,74	0,18	0,52	0,99	0,62
r	0,72	0,70	0,82	0,68	0,76	0,65	0,61	0,58	0,70	0,44

Согласно принятым в первом разделе настоящей статьи семантическим правилам определения типа агента, за 6 итераций без управляющих воздействий было получено, что один из агентов относится к принуждающему типу ($r_{A_3} = 0,82$), один – к уклоняющемуся ($r_{A_{10}} = 0,44$), остальных агентов можно определить как сотрудничающих (компромиссных). Задача будет решена только тогда, когда все ИНАГи будут компромиссными, поэтому для решения задачи за 6 итераций необходимо определять тип агента в соответствии с формулой (4) и осуществлять некоторые управляющие воздействия. Они могут быть [Губанов и др., 2010]:

- институциональными, подразумевающими изменение ограничений, взаимосвязей и норм деятельности;
- мотивационными, т.е. заключающимися в изменении предпочтений;
- информационными, т.е. состоящими во влиянии на информированность агента.

Т.е., задача ИНАГа-Центра (или пользователя МАС), зная тип агента, определить, какие управляющие воздействия следует выбрать, чтобы действия агента оказались более выгодными для Центра. В данном случае предлагается применять механизмы управления, при которых ИНАГам, как рациональным сущностям, имеющим свои предпочтения, т.е. стремящимся максимизировать собственную функцию полезности, выгодно быть честными. Это так называемые механизмы «честной игры» или неманипулируемые механизмы. Агентам выгодно быть честными лишь в одном случае: когда сообщаемая ими информация не используется им же во вред. Это и есть основной принцип механизмов «честной игры». [Новиков, 2011] В дальнейшем применяется механизм управления, состоящий из следующих этапов:

1. Центр сообщает агенту механизм функционирования («правила игры») в общем виде, например, зависимость размера вознаграждения агента от достигнутых им результатов. Здесь под результатами понимается состояние агента в момент времени t .

2. Агент сообщает Центру информацию о параметрах, неизвестных Центру. Например, сообщает информацию о своих предпочтениях.

3. Центр доводит до сведения агента параметры механизма функционирования с их обязательной детализацией. Например, представляет план, являющийся ожидаемым от агента результатом деятельности.

4. Агент выбирает стратегию поведения, что сопровождается изменением значений $\mu_{x_1}(x_{1t}), \mu_{x_2}(x_{2t}), \dots, \mu_{x_{10}}(x_{10t})$ и, осуществляя действия согласно выбранной стратегии, получает определенный результат. При этом он переходит в следующее состояние, например, $s_1 \rightarrow s_2$.

5. Центр получает информацию о новом состоянии агента, а, следовательно, и о полученном

им результате деятельности.

6. Центр в соответствии с механизмом функционирования сообщает агенту свое действие, например, размер поощрения, количество выделенного ресурса и т.п.

Процесс управления продолжается до тех пор, пока не будет найдено решение задачи за оптимальное число итераций.

Возвращаясь к нашему примеру, очевидно, что на первой итерации состояния агентов A_2, A_3, A_6, A_7 и A_{10} нарушают равновесие и устойчивость МАС. С помощью управляющих воздействий $v_{2_1}, v_{3_1}, v_{6_1}, v_{7_1}$ и v_{10_1} на второй итерации были получены следующие состояния ИНАГов (табл. 3):

Таблица 3 – Состояния агентов на второй итерации работы МАС

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
s_1	0,75	<u>0,88</u>	<u>0,84</u>	0,79	0,79	<u>0,87</u>	<u>0,22</u>	0,78	0,72	<u>0,31</u>
s_2	<u>0,08</u>	0,78	<u>0,99</u>	<u>0,86</u>	0,71	0,73	0,79	<u>0,89</u>	<u>0,84</u>	0,69

Как видно из табл. 3, агентов A_2, A_6, A_7 и A_{10} удалось привести в состояние компромисса (значения функций принадлежности выделены полужирным шрифтом), однако усилилась склонность к конфликту агента A_3 , агентов A_4, A_8 и A_9 . Агент A_1 стал демонстрировать отстраненность от решения поставленной задачи (значения подчеркнуты). При выборе следующих управляющих воздействий $v_{1_2}, v_{3_2}, v_{4_2}, v_{8_2}$ и v_{9_2} Центр должен руководствоваться принципами примирения. В табл. 4 показаны состояния агентов на третьей итерации.

Таблица 4 – Состояния агентов на третьей итерации работы МАС

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
s_2	<u>0,08</u>	0,78	<u>0,99</u>	<u>0,86</u>	0,71	0,73	0,79	<u>0,89</u>	<u>0,84</u>	0,69
s_3	<u>0,32</u>	0,71	<u>0,85</u>	0,79	0,64	<u>0,82</u>	0,66	0,75	<u>0,81</u>	0,54

Далее снова применяются управляющие воздействия и так – до тех пор, пока в МАС не установится состояние согласованной кооперации [Myerson, 2001] и, как следствие, не будет достигнута поставленная цель. Серия испытаний показала, что если в момент t уделять внимание (оказывать управляющие воздействия) только лишь на агентов, поведение которых отклоняется от нормы (норма – компромисс, сотрудничество), то те агенты, которые в этот момент времени находились в компромиссном состоянии, в момент $t+1$ могут перейти состояние, нежелательное для МАС. Поэтому на каждой итерации следует контролировать поведение всех ИНАГов без исключения, стараться сохранить состояние тех агентов, которые показывают хорошие результаты

работы.

Применение этого принципа позволило получить равновесное состояние МАС уже на пятой итерации (табл. 5). Здесь все агенты готовы на компромисс.

Таблица 5 – Состояния агентов на пятой итерации работы МАС

	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀
s ₄	0,52	0,67	<u>0,81</u>	0,77	0,71	0,68	0,66	0,75	<u>0,83</u>	0,65
s ₅	0,58	0,63	0,78	0,75	0,79	0,64	0,67	0,71	0,77	0,78

Такое состояние МАС гарантирует решение поставленной задачи с высокой степенью уверенности.

3. ПРОИЗВОДНЫЕ ЗАДАЧИ ИССЛЕДОВАНИЯ

С помощью нечеткой логики [Беллман, 1976] можно решать различные задачи, относящиеся к теории МАС, например, задачу прогнозирования временных затрат для достижения МАС поставленной цели. При этом учитывается состояние МАС и сложность цели (число итераций, за которое можно ее достичь). Для этого необходимо построить нечеткую базу знаний, основанную на нечеткой импликации. Нечеткий вывод представим в виде:

$$A \rightarrow B = \overline{A \cup B},$$

$$\mu_{A \rightarrow B}(x) = \max \{1 - \mu_A(x), \mu_B(x)\}. \quad (6)$$

Основой для проведения операции нечеткого логического вывода является база правил, содержащая нечеткие высказывания в форме ЕСЛИ→ТО и функции принадлежности для соответствующих лингвистических термов. При этом должны соблюдаться следующие условия:

- существует хотя бы одно правило для каждого лингвистического термина выходной переменной;
- для любого термина входной переменной имеется хотя бы одно правило, в котором этот терм используется в качестве предпосылки.

Иначе имеет место неполная база нечетких правил. Лингвистической переменной называется набор:

$$\langle \gamma, T, U, G, M \rangle, \quad (7)$$

где γ – наименование лингвистической переменной; T – множество ее значений (терм-множество), областью определения каждого из которых является множество U ; G – синтаксическая процедура, позволяющая оперировать элементами терм-множества T , в частности, генерировать новые термы; M – семантическая процедура, позволяющая превратить каждое новое значение лингвистической переменной, образуемое процедурой G , в нечеткую переменную, т.е. сформировать соответствующее нечеткое множество. Нечеткая переменная

характеризуется набором $\langle \alpha, U, A \rangle$, где α – наименование переменной; U – универсальное множество (область определения α); A – нечеткое множество на U , описывающее ограничения (т.е. $\mu_A(x)$) на значения нечеткой переменной α .

Для реализации логического вывода необходимо выполнить следующие действия:

- 1) сформулировать на естественном языке в виде предложений «Если..., то...» закономерности предметной области;
- 2) выделить из этих предложений лингвистические переменные, их значения (построить их функции принадлежности), высказывания различных видов, формализовать нечеткие правила;
- 3) проверить полученную базу знаний на полноту;
- 4) провести фаззификацию, т.е. подготовить задачу для решения методами нечеткой логики;
- 5) провести аккумуляцию;
- 6) выполнить дефаззификацию. Это действие предполагает применение какого-либо математического метода, например, вычисления «центра тяжести» нечеткого множества, т.е. срединного численного значения полученного нечеткого множества.

Пример. Рассматривается МАС, состоящая из 10 ИНАГов. Требуется построить нечеткую базу знаний для задачи оценки временных затрат, необходимых МАС для достижения ею поставленной цели. При этом учитывается состояние МАС (характеризуется состояниями ИНАГов) и сложность достижения цели (для простоты описывается числом итераций).

1) Предложения, описывающие задачу, следующие:

- если состояние МАС компромиссное или почти компромиссное и сложность достижения цели незначительна, то временные затраты небольшие;
- если состояние МАС компромиссное или почти компромиссное и сложность достижения цели высокая, то временные затраты достаточно большие;
- если состояние МАС не компромиссное и сложность достижения цели высокая, то временные затраты большие;
- если состояние МАС почти компромиссное и сложность достижения цели значительная, то временные затраты достаточно большие.

2) Из этих предложений выделяются лингвистические переменные, определяемые через формальную запись (7):

$$\checkmark \gamma = \text{Состояние_МАС},$$

$T = (\text{«компромиссное»}, \text{«почти компромиссное»}, \text{«не компромиссное»})$, $U = [0, 10]$ (число агентов $N = 10$), $G = (\text{«совсем не компромиссное»}, \text{«далеко от компромиссного»}, \text{«компромиссное или почти компромиссное»})$, M – уменьшение на единицу

степени принадлежности нечеткой переменной «компромиссное», операция объединения нечетких множеств;

✓ γ = Сложность_достижения_цели,

T = («незначительная», «значительная», «высокая»), $U = [1,30]$ (число итераций), G = («очень высокая», «значительная или высокая»), M – увеличение на единицу степени принадлежности нечеткой переменной «высокая», операция объединения нечетких множеств;

✓ γ = Количество_времени, T = («мало», «достаточно», «много»), $U = [1,20]$ (количество минут), G = («очень много», «достаточно или мало»), M – увеличение на единицу степени принадлежности нечеткой переменной «много», операция объединения нечетких множеств.

Для полного задания лингвистической переменной необходимо определить нечеткие переменные, входящие в T (рисунок 1).

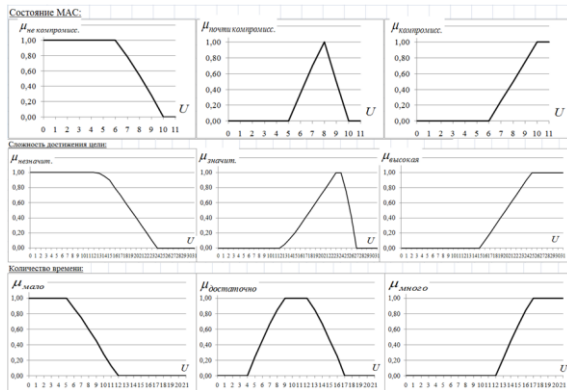


Рисунок 1 – Нечеткие переменные

С учетом выделенных лингвистических переменных нечеткие правила следующие:

Правило 1. Если Состояние_МАС= «компромиссное» или Состояние_МАС= «почти компромиссное» и Сложность_достижения_цели= «незначительная», то Количество_времени= «мало».

Правило 2. Если Состояние_МАС= «компромиссное» или Состояние_МАС= «почти компромиссное» и Сложность_достижения_цели= «высокая», то Количество_времени= «достаточно».

Правило 3. Если Состояние_МАС= «некомпромиссное» и Сложность_достижения_цели= «высокая», то Количество_времени= «много».

Правило 4. Если Состояние_МАС= «почти компромиссное» и Сложность_достижения_цели= «значительная», то Количество_времени= «достаточно».

3) Проверка полученной базы на полноту. Согласно приведенным выше условиям: выходная переменная Количество_времени имеет три терма: «мало» используется в первом правиле, «достаточно» – во втором и четвертом, «много» – в третьем правиле. Используются две входные переменные: Состояние_МАС и

Сложность_достижения_цели. У каждой из них – по три терма, применяемые в сформулированных правилах хотя бы один раз. Следовательно, полученная база нечетких правил полная.

4) Пусть МАС находится в состоянии s_4 (табл. 5) и ей поручено выполнение задачи, состоящей из 18 итераций. Для оценки временных затрат определим степени уверенности простейших утверждений (табл. 6).

Таблица 6 – Степени уверенности базовых утверждений

Переменная	Значения	Степени уверенности
Состояние МАС	«компромиссное»	0,54
	«почти компромиссное»	1,00
	«не компромиссное»	0,50
Сложность достижения цели	«незначительная»	0,60
	«значительная»	0,50
	«высокая»	0,30

Определяются степени уверенности посылок правил (табл. 7).

Таблица 7 – Степени уверенности посылок правил

Правило 1	$\mu_1 = \min(\max(0,54;1,00);0,60) = 0,60$
Правило 2	$\mu_2 = \min(\max(0,54;1,00);0,30) = 0,30$
Правило 3	$\mu_3 = \min(0,50;0,30) = 0,30$
Правило 4	$\mu_4 = \min(1,00;0,50) = 0,50$

Графическое изображение результатов, полученных в табл. 7, представлено на рисунке 2.

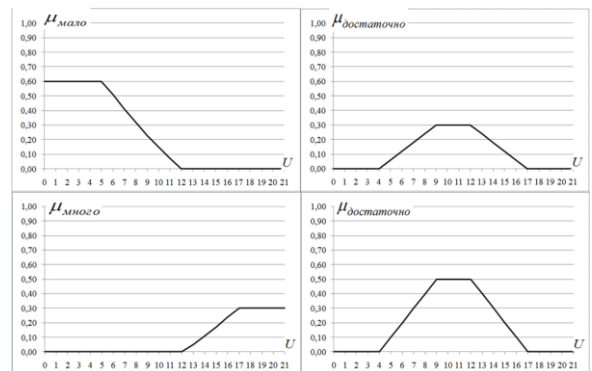


Рисунок 2 – Фаззификация

5) Построение новой выходной нечеткой переменной с использованием полученных степеней уверенности (рисунок 3).

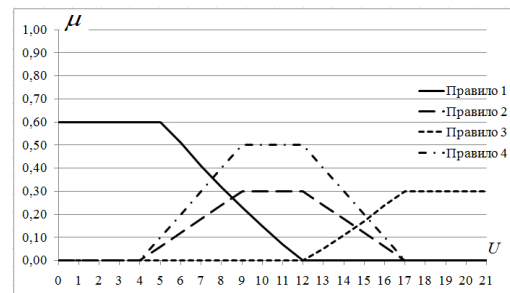


Рисунок 3 – Аккумуляция

б) Новый терм выходной переменной Количество_часов представлен на рисунке 4.

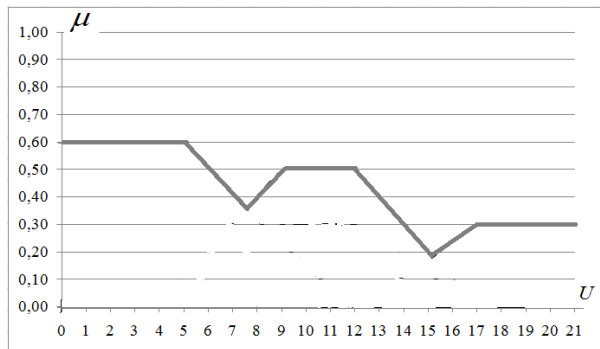


Рисунок 4 – Дефаззификация

Исходя из полученного графика степени принадлежности выходного термина, можно сказать, что МАС, имеющей состояние s_4 , на достижение цели сложностью 18 итераций понадобится, скорее всего, от одной до пяти минут. Степень уверенности данного утверждения составляет 0,6.

На основе представленного семантического базиса и правил нечеткой логики можно решать и обратные задачи, например, определять оптимальное число ИНАГов в МАС и оценивать сложность достижения поставленной в МАС цели. В первом случае входными переменными будут Состояние_МАС, Сложность_достижения_цели и Количество_времени, выходной переменной – Число_агентов. Во втором случае входными переменными являются Состояние_МАС, Число_агентов и Количество_времени. Однако с увеличением числа переменных сложность решения возрастает, поэтому предлагается следующий метод оценки сложности достижения цели:

1) задается множество лингвистических переменных, их нечеткие значения описываются функциями принадлежности. Сложность достижения цели может быть охарактеризована двумя переменными: Числом итераций и Количеством времени и представлена с помощью интегрального коэффициента сложности (ξ_{goal});

2) мнения ИНАГов, высказанные относительно возможного значения оцениваемого показателя, описываются на языке лингвистических переменных. Нечеткие значения могут задаваться вербально или определяться на основе точных числовых данных. Мнения каждого ИНАГа оцениваются в зависимости от его типа. Принимаются весовые коэффициенты (табл. 8), которые могут быть интерпретированы как степени уверенности в отсутствии искажающей действительности составляющей сообщений агента.

Таблица 8 – Степени уверенности в правдивости агентов

Тип ИНАГа	$r \in (0;0,5)$	$r \in [0,5;0,8]$	$r \in (0,8;1)$
«Вес»	0,2	0,5	0,3

3) к лингвистическим переменным,

выражающим полученные от агентов мнения, применяются соответствующие логические операции для расчета значения итогового показателя. Например, для нахождения интегральной оценки значения коэффициента ξ_{goal} в виде нечеткого множества, может быть использовано максиминное правило композиции нечетких множеств:

$$\xi_{goal} \circ \dots \circ \xi_{goal_k} = \left\{ \begin{array}{l} p, \mu(p) = \max \{ \min [\mu_1(p_1), \dots, \mu_N(p_N)] \} \\ p \in P, \forall p_i \end{array} \right\}, \quad (8)$$

где $\xi_{goal} \circ \dots \circ \xi_{goal_k}$ – композиция нечетких множеств, определенных на скорректированной шкале предпочтений ИНАГов P (с учетом их типа);

P_1, \dots, P_N – дискретные значения на шкале предпочтений P , которым соответствуют значения функций принадлежности множеств $\xi_{goal_1}, \dots, \xi_{goal_k}$;

$\mu_1(p_1), \dots, \mu_N(p_N)$ – значения функций принадлежности множеств $\xi_{goal_1}, \dots, \xi_{goal_k}$, соответствующие значениям p_1, \dots, p_N .

С учетом принятого дискретного типа шкалы предпочтений ИНАГов P значения p_i , получаемые по формуле (8), округляются до ближайшего шага t . Таким образом, интегральная оценка значения коэффициента сложности достижения цели представляет собой нечеткое множество, описанное с помощью функции принадлежности $\mu \in [0;1]$, дискретно определенной на шкале предпочтений P : $p_i \in [\xi_{goal}^{\min}; \xi_{goal}^{\max}]$ с выбранным шагом t ;

4) искомое значение коэффициента сложности достижения цели (ξ_{goal}) представляется в виде среднего значения (9) полученного нечеткого множества, на основе интерпретации которого и принимается решение.

$$\xi_{goal} = \frac{\sum_{i=1}^N p_i \mu_i}{\sum_{i=1}^N \mu_i}. \quad (9)$$

Интерпретация ξ_{goal} выполняется ИНАГом-Центром или пользователем МАС.

ЗАКЛЮЧЕНИЕ

Нечеткая логика является удобным средством обучения и мониторинга МАС. Управление на основе нечеткой логики позволяет сформировать базу нечетких правил для осуществления оптимальных управляющих воздействий на ИНАГов с целью разрешения конфликтов и

противоречий, возникающих в МАС. При этом оптимальным решением является внедрение формулы управления в антропоморфную сущность ИНАГа с тем, чтобы ее действие было неявным: «Мне никто не дает команд, но мое поведение управляется моей собственной концепцией Я, содержащей принципы, согласно которым я должен вести себя именно так, а не иначе».

Правила нечеткой логики позволяют обеспечить:

- 1) применение существующего опыта управления;
- 2) использование гибких правил взаимодействия агентов в МАС;
- 3) улучшение качества управления посредством саморегулирования управляющей системы и упреждающего изменения выходного воздействия (значения функции упреждения). Это позволяет свести к минимуму возможность выхода МАС из состояния равновесия.

Для согласованного принятия решения необходимо доверие, иначе нельзя достичь в МАС синергии успеха. Доверие может быть измерено быстротой согласования групповых совместных решений. На скорость принятия согласованного группового решения может влиять наличие лидера, общей проблемы и цели. Агенты-лидеры в спонтанном сетевом общении формируются в процессе взаимодействия. Принятие групповых решений характеризуется итерационной последовательностью действий участников. Чтобы процесс формирования и согласования решений оказался сходящимся, т.е. с каждой итерацией настойчиво продвигающимся ближе к цели, нужны специальные механизмы. Такими механизмами могут быть механизмы «честной игры». Они позволяют достичь равновесного состояния МАС, т.е. состояния, одностороннее отклонение от которого не выгодно ни одному из ИНАГов.

Работа выполнена при поддержке гранта РФФИ 12-01-01005-а «Разработка моделей и методов согласованного управления в многоагентных системах».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Беллман, 1976] Беллман, Р. Принятие решений в расплывчатых условиях/Р. Беллман, Л. Заде// В кн.: Вопросы анализа и процедуры принятия решений [под ред. И.Ф. Шахнова] М.: Мир, 1976. 230 с., С. 172 – 215
- [Городецкий и др., 2009] Городецкий, В.И. Прикладные многоагентные системы группового управления/В.И. Городецкий, О.В. Карсаев, В.В. Самойлов, С.В. Серебряков// Искусственный интеллект и принятие решений. - 2009. - № 2. - С.3-24
- [Городецкий и др., 1998] Городецкий, В.И. Многоагентные системы (обзор)/В.И. Городецкий, М.С. Грушинский, А.В. Хабалов// Новости искусственного интеллекта. - 1998. - № 2. - С.64-116
- [Губанов и др., 2010] Губанов, Д.А. Социальные сети: модели информационного влияния, управления и противоборства / Д.А. Губанов, Д.А. Новиков, А.Г. Чхартишвили. – М.: Физматлит, 2010. 228 с.
- [Кузнецов и др., 2012] Кузнецов, В.Н. Модели и методы согласованного управления в многоагентных системах / В.Н.

Кузнецов, А.Ю. Клошин, Н.Ю. Мутовкина, Г.В. Кузнецов // Программные продукты и системы, 2012, № 4, С. 231-235

[Новиков и др., 2011] Механизмы управления / Под ред. Д.А. Новикова. – М.: ЛЕНАНД, 2011. 192 с.

[Рыбина, 2008] Рыбина, Г.В. Моделирование процессов взаимодействия интеллектуальных агентов в многоагентных системах / Г.В. Рыбина, С.С. Паронджанов // Искусственный интеллект и принятие решений, 2008, № 3, С. 3-15

[Чхартишвили, 2011] Чхартишвили, А.Г. Согласованное информационное управление / А.Г. Чхартишвили // Проблемы управления. - 2011. - № 3. - С.43-48

[Myerson, 2001] Myerson, R.B. Game theory: analysis of conflict / R.B. Myerson // London: Harvard Univ. Press, 2001. 568 p.

SEMANTIC DEFINITION OF THE TYPE OF AGENT IN MULTI-AGENT SYSTEM. THE PROBLEM OF INTERACTION BETWEEN AGENTS

Mutovkina N. Yu, Klyushin A. Yu., Kuznetsov V.N.

Tver State Technical University, Tver, Russia

Mutovkina_N@mail.ru

Klalex@inbox.ru

Bumhouse@mail.ru

The paper describes approaches to the definition of the type of intelligent agents and the status of the multi-agent system, based on the theory of fuzzy sets and fuzzy logic. Described in the concepts and methods are illustrated by examples of determining the type of agents and bringing multi-agent system in a state of compromise. Also the article contains derivatives of these tasks relevant for the development of models of interaction between agents.

INTRODUCTION

The necessary condition for the development of models and methods of coherent control in multi-agent systems (MAS) is to define the key characteristics of the elements of the MAS – intelligent agents and the possible ways of their interaction. In the basis of the research is anthropomorphic definition of an agent. The purpose of this research is the development of methodology of determination of behaviour type of INAG, reflecting his preferences, and change the type (if necessary) by means of the control actions, using the theory of fuzzy sets and fuzzy logic.

MAIN PART

In the main part of the article are:

- 1) setting and method of solving the problem;
- 2) example of the types of agents in MAS;
- 3) derivative objectives of the study.

CONCLUSION

Management on the basis of fuzzy logic allows to form the base of fuzzy rules for the implementation of the optimal control actions on INAG's with a view to the settlement of the conflicts and contradictions emerging in the MAS.

СЕКЦИЯ 5.

**СЕМАНТИЧЕСКИЕ МОДЕЛИ, СРЕДСТВА И МЕТОДЫ
КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ПОЛЬЗОВАТЕЛЬСКИХ
ИНТЕРФЕЙСОВ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ**

SECTION 5.

**SEMANTIC MODELS, TOOLS AND METHODS OF COMPONENT
DESIGN USER INTERFACES OF INTELLIGENT SYSTEMS**



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК [004.522+004.934+004.91]:004.89

КАМПАНЕНТЫ ІДЭНТЫФІКАЦЫІ КОЛЬКАСНЫХ ВЫРАЗАЎ З АДЗІНКАМІ ВЫМЯРЭННЯ Ў ТЭКСТАХ НА БЕЛАРУСКАЙ І РУСКАЙ МОВАХ

Гецэвіч Ю.С.* , Скопінава А.М.*

** Аб'яднаны інстытут праблем інфарматыкі Нацыянальнай акадэміі навук Беларусі,
г. Мінск, Беларусь*

yury.hetsevich@gmail.com

skelena777@gmail.com

Разгледжана актуальнасць і акрэслены складанасці праблемы ідэнтыфікацыі колькасных выказаў з адзінкамі вымярэння на прыкладзе беларускай і рускай моў на матэрыяле навукова-тэхнічных і прававых тэкставых карпусоў. Апісаньня і алгарытмічна прадстаўленьня адасобленьня семантычных кампанентаў для пошуку колькасных выказаў з улікам варыятыўных спосабаў запісу іх складнікаў: лічбавых дэскрыптараў і непасрэдна адзінак вымярэння. Для атрыманых практычных вынікаў азначаны магчымыя вобласці прымянення.

Ключавыя словы: кампанент; ідэнтыфікацыя выказаў; колькасны выраз; адзінка вымярэння; канчатковы аўтамат, NooJ.

Уводзіны

Калі заходзіць гаворка пра распрацоўку інтэлектуальных сістэм, заўсёды ўзнікае пытанне пра спосабы распазнавання (расшыфроўкі) структураванай інфармацыі. Вядома, што тэксты часта ўтрымліваюць складаныя літарна-сімвалыя канструкцыі, якія былі ўжытыя стваральнікамі тэкстаў для апісання розных падзей ці сфер жыцця. Такімі канструкцыямі з'яўляюцца колькасныя выразы ў спалучэнні з мернымі сістэмамі або з сістэмамі адзінак вымярэння, якія з'яўляюцца важнымі для метралогіі, матэматыкі, інфарматыкі, фізікі, тэорыі кадавання, прамысловасці, эканомікі і гандлю і інш. Колькасныя апісанні ўласцівыя агульнай навуковай карціне свету, і, канешне, бытавой сферы жыцця. У якасці прыкладаў можна назваць наступныя спалучэнні колькасцяў з часта выкарыстанымі адзінкамі вымярэння: *сіла току 59 мА, даўжыня 400 м, маса трыццаць пяць кілаграм, цеплыня 200 кДж, 225 ккал, 35 руб., пяць галоўнаў паліва* і г. д.

Увогуле тэксты, якія ўтрымліваюць колькасныя выразы з адзінкамі вымярэння (КВАВ), патрабуюць спецыяльных прыкладных алгарытмаў для рашэння актуальных задач у наступных сферах:

- *інтэлектуальныя пытальна-адказныя сістэмы* (для фармавання разнастайных запытаў кампанента інтэлектуальнага інтэрфейса; для хуткай класіфікацыі выказаў з адзінкамі вымярэння

па спецыяльных класах у базах ведаў; для ўдакладнення параметраў вызначаных аб'ектаў у блоку лагічных аперацый);

- *сістэмы сінтэзу маўлення па тэксце* (для правільнай генерацыі арфаграфічнага тэксту па ўваходным тэксце; для правільнага ўтварэння недзялімых паслядоўнасцяў слоў і сінтагмаў);

- *сістэмы пошуку і апрацоўкі інфармацыі, каталогі і бібліятэкі* (для фармавання пашыраных пошукавых запытаў, якія змогуць знайсці канкрэтныя мерныя выразы у Інтэрнэце ці ў лакальнай базе тэкстаў; для аўтаматычнага рэфэравання і анатавання);

- *выдавецкія установы* (для аўтаматызаванай лакалізацыі канкрэтнага спісу выказаў з адзінкамі вымярэння і для хуткага размеркавання знойдзеных выказаў па класах; для хуткай праверкі правільнасці ўжыванняў разгорнутых формаў назваў адзінак вымярэння ў тэкстах).

Пералічым асноўныя складанасці, якія ўзнікаюць пры распрацоўцы алгарытмаў ідэнтыфікацыі колькасных выказаў з адзінкамі вымярэння ў тэкстах:

1. *Выразы з колькасцю і адзінкамі вымярэння маюць шырокую варыятыўнасць як па напісанні, так і па ўтварэнні.* Менавіта з-за гэтага адразу запісаць правільны лакалізацыі выказаў для ўсіх выпадкаў практычна немагчыма. Для спрашчэння гэтага працэсу патрэбна выкарыстоўваць прыстасаванні, якія дазваляюць зручна карэктаваць ужо распрацаваныя правільны і дадаваць новыя.

2. *Выраз з колькасцю і адзінкай вымярэння складана ідэнтыфікаваць і выдзеліць ў ім колькасць (лік ці лічэбнік) і назву адзінкі (скарочаную ці поўную назвы) без добра падрыхтаваных лінгвістычных слоўнікаў.* Так адбываецца па той прычыне, што гэтыя слоўнікі павінны ўтрымліваць апісанне усіх словаформаў, скарачэнняў і правілаў пабудовы вытворных формаў колькасцяў і назваў адзінак вымярэння. Напрыклад, гэта патрэбна для правільнай лакалізацыі і аналізу наступных выразаў з варыятыўнымі спосабамі запісу адзінак вымярэння даўжыні: *25 метраў, 21 метр, 100 м, тры метры.*

3. *Выразы з колькасцю і адзінкай з'яўляюцца мовазалежнымі.* Напрыклад, у англійскай мове скарачэннем слова *метр* з'яўляецца сімвал "m", а ў беларускай – "м". Назвы адзінак вымярэння адрозніваюцца па напісанні ў беларускай і рускай мовах (напрыклад, «гадзіна», «час»). Таму для кожнай мовы патрэбна рабіць адмысловыя ўдакладненні пры распрацоўцы алгарытмаў ідэнтыфікацыі КВАВ.

Варта адзначыць, што некаторыя крокі для рашэння вышэй абазначаных праблем былі ажыццэўлены ў 2009 г. групай харвацкіх лінгвістаў, якія пабудавалі алгарытмы для вызначэння мерных выразаў даўжыні, плошчы і лікавых дыяпазонаў для ангельскай і харвацкай моў [Bekavac, 2009]. Мэтавая прадметная вобласць атрымала некаторае асвятленне ў працах шэрагу іншых еўрапейскіх лінгвістаў. Аднак іх даследаванні былі зробленыя больш тэарэтычна, чым практычна. Яны маюць апісальны характар і сканцэнтраваны не канкрэтна на адзінках вымярэння як на асобных паняццях, а як на «вызначаных выпадках ужыванняў слоў і выразаў, якія складаюць спецыяльную катэгорыю пайменаваных еднасцяў» [Cunningham, 1999] [пераклад тут і далей наш]. Таму яны апісаныя толькі паверхнева. Супрацоўнікі Балгарскай акадэміі навук і аддзялення Шэфілдскага ўніверсітэта адзначаюць, што іх «назіранні пра лінгвістычную сутнасць славянскіх пайменаваных еднасцяў заснаваны толькі на агульных характарыстыках і высновах згодна з асаблівасцямі іх ужывання ў тэкстах» [Paskaleva, 2002]. На практыцы аказваецца, што за тэрмінам «пайменаваная еднасць» хаваецца велізарны набор іншых складаных катэгорый: геаграфічных найменняў, асабістых прозвішчаў, імёнаў, мянушак людзей, назваў арганізацый, пазначэнняў дат, часу, грашовых і працэнтных адносін [Paskaleva, 2002; Mukowiecka, 2007]. Такім чынам, каб атрымаць эфектыўныя алгарытмы ідэнтыфікацыі, да кожнай з гэтых катэгорый варта падысці асобна. Балгара-сербскай камандай паняцце выразу з мернымі адзінкамі было разгледжана больш вузка і фармальна прадстаўлена ў выглядзе графа як «структура, якая складаецца са спалучэння ліку, запісанага словамі ці лічбамі, і індыкатара мернай адзінкі (*кіламетр, градус ухілу, міля, фут*, і да т.п.)» [Duško, 2007]. Тым не менш, атрыманы вынік прывязаны да вызначаных моўных сістэм

(балгарскай і сербскай), у той час як астатнія на практыцы могуць у большай ці меншай ступені выкарыстоўвацца для харвацкай, ангельскай, польскай і ніводная – для беларускай ці рускай моў.

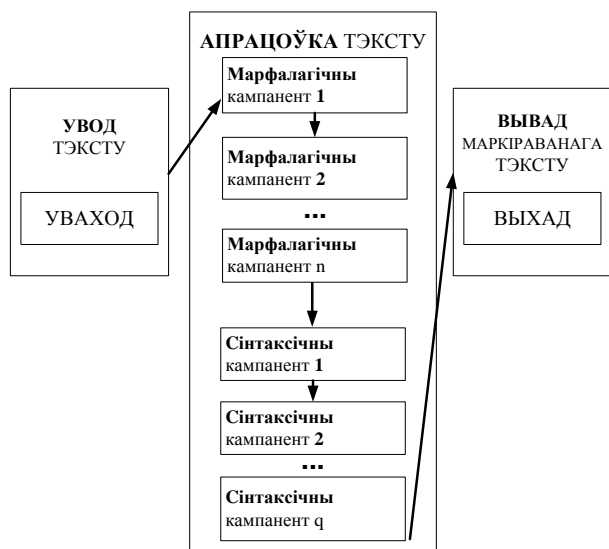
Таксама на сённяшні час ніводная пошукавая сістэма не прадстаўляе інтэрфейс для пабудовы пошукавага запыту з функцыяй лакалізацыі пэўных адзінак вымярэння.

Аўтарамі гэтага дакладу ўжо былі рэалізаваныя некаторыя крокі ў вызначаным напрамку. На XI міжнароднай канферэнцыі РІНТІ'2013 (г. Мінск, 15 лістапада 2012 г.) былі прадстаўлены алгарытмы, якія знаходзяць колькасныя выразы з адзінкамі вымярэння і класіфікуюць іх па трох тыпах (C1, вытворны ад C1, не C1) з дакладнасцю 72% [Гецэвіч, 2012]. На практыцы кожны з гэтых алгарытмаў быў рэалізаваны ў выглядзе завершанага кампанента пры дапамозе камп'ютэрна-лінгвістычнага сродка NooJ [Silberztein, 2003].

У гэтым дакладзе ставіцца задача далей паляпшаць распрацаваныя алгарытмы. А менавіта на базе тэкстаў навукова-тэхнічнай і прававой тэматыкаў распрацаваць комплекс асобных сінтаксічных і марфалагічных кампанентаў, якія будуць здольныя ідэнтыфікаваць колькасныя выразы з *адзінкамі* вымярэння, якія перададзеныя пры дапамозе розных цэлых і скарачаных, дольных і кратных прэфіксаў (*фемтаграмы, кБайт, дм, гПа*), а таксама пры дапамозе цэлых і скарачаных асноў (*нанафарад, мЗв*).

1. Агульны агляд працэсу ідэнтыфікацыі колькасных выразаў з адзінкамі вымярэння

У кантэксце інтэлектуальных сістэм увесь шлях ад увода першапачатковага тэкставага фрагменту на ўваходзе да вывада апрацаванага тэксту з ідэнтыфікаванымі КВАВ можна прадставіць праз паслядоўнасць марфалагічных і сінтаксічных кампанентаў-апрацоўшчыкаў (малюнак 1).



Малюнак 1 – Універсальная схема працэсу ідэнтыфікацыі колькасных выказаў з адзінкамі вымярэння праз марфалагічныя і сінтаксічныя кампаненты

Спачатку ўведзены тэкст на любой мове апрацоўваецца з дапамогай слоўнікаў – кожны токен абазначаецца лексікаграфічнымі пазнакамі (напрыклад, род, лік, склон, клас канчаткаў), а далей незалежнымі марфалагічнымі (M1, M2, ..., Mn) і сінтаксічнымі (S1, S2, ..., Sq) кампанентамі. Кожны з кампанентаў з’яўляецца выканальным канчатковым аўтаматам, які выяўляецца графам з адным уваходам і адным выходам, злучанымі шляхамі-канэктарамі з умовамі пераходу з аднаго вузла-стана ў іншы вузел-стан. Марфалагічныя кампаненты паслядоўна прымяняюцца да словаформаў-токенаў, каб даследаваць іх склад і абазначыць па жаданні распрацоўшчыка іх асаблівасці праз спецыяльныя адвольныя карыстальніцкія маркеры. Сінтаксічныя кампаненты таксама дазваляюць ужываць маркеры, але ўжо для *спалучэнняў* слоў, лічбаў, знакаў пунктуацыі і інш. сімвалаў.

З пункту погляду рэалізацыі і выкарыстання марфалагічных і сінтаксічных кампаненты з’яўляюцца спецыяльнымі файламі, якія могуць быць выкліканы незалежна адзін ад аднаго староннімі праграмамі праз унутраныя сродкі NooJ.

Тэарэтычна рашэнне вышэй пастаўленай задачы таксама можна рэалізаваць з дапамогай рэгулярных выказаў. Але, у адрозненне ад іх, візуальныя графы (ці канчатковыя аўтаматы) дазваляюць хутчэй і зручней мадэлізаваць і правяраць алгарытмы кампанентаў падчас распрацоўкі. Гэта сапраўды важна, бо КВАВ4 уласцівы варыятыўны характар. Фармальна пералічыць і апісаць усе выпадкі іх ужыванняў адразу практычна не магчыма.

Акрамя гэтага, NooJ дазваляе ствараць тэставыя корпусы тэкстаў, неабходныя для адладкі працы алгарытмаў. Аўтарамі былі створаны 4 тэставыя корпусы для 2 розных тэматычных даменаў: навукова-тэхнічнага (астраномія, фізіка, геаграфія, хімія, авіяцыя, касманаўтыка, гісторыя, энергетыка, транспарт і сувязь) і прававога (збор правіл

дарожнага руху для Беларусі) – па 2 корпусы для кожнай з моў [Гецэвіч, 2012]. Далей усе корпусы былі папоўненыя тэкстамі з КВАВ розных тыпаў.

Наступны раздзел апісвае назначэнне і асаблівасці рэалізацыі кожнага кампанента комплексу ідэнтыфікацыі КВАВ.

2. Комплекс кампанентаў для ідэнтыфікацыі колькасных выказаў з адзінкамі вымярэння па словаўтваральных прыкметах

Згодна са словаўтваральнымі асаблівасцямі, у корпусе сустракаюцца наступныя адзінкі вымярэння:

- з цэлай асновай і без прэфікса (*метр, Герц, Ом*);
- з цэлай асновай і з цэлым прэфіксам (*нанафарады, міліампер*);
- з цэлай асновай і са скарачаным прэфіксам (*кБайт*).
- са скарачанай асновай і без прэфікса (*Дж, га, Па*);
- са скарачанай асновай і са скарачаным прэфіксам (*км, дл, гПа*);

Гэтая ўмоўная класіфікацыя была выкарыстана як галоўны прынцып пры распрацоўцы кампанентаў ідэнтыфікацыі КВАВ.

Спачатку былі створаныя лінгвістычныя слоўнікі S для беларускай (БМ) і рускай (РМ) моў (малюнак 2). Гэтыя слоўнікі ўтрымліваюць базавыя асновы адзінак вымярэння – назойнікі і іх абрэвіятуры. Яны адпавядаюць класіфікацыі як поўныя і скарачаныя асновы адзінак вымярэння, прычым кожная аснова пазначана адпаведным атрыбутам «Base» або «Mbase». Да поўных асноў дадаецца пазнака флексійнага класа. Відавочна, што слоўнік S з’яўляецца мовазалежным лінгвістычным рэсурсам, у адрозненне ад кампанентаў-алгарытмаў ідэнтыфікацыі КВАВ, якія пабудаваныя як мованезалежныя модулі.

Далей былі пабудаваныя мовазалежныя лінгвістычныя рэсурсы (Fsubmultiple, Fmultiple, Ssubmultiple, Smultiple) з усімі дазволенымі прэфіксамі для ўтварэння адзінак вымярэння. Крыніцай для гэтага былі дадзеныя Міжнароднага бюро мер і вагаў [BIPM – SI brochure (8th edition), 2006]. Для ўтварэння адзінак вымярэння могуць выкарыстоўвацца кратныя (multiple) ці дольныя (submultiple) прэфіксы, прычым ў скарачанай (прэфікс S) або поўнай (прэфікс F) формах (адпаведна малюнак 3, малюнак 4).

Згодна з вышэй апісанай класіфікацыяй па словаўтваральным прынцыпе, былі распрацаваныя 4 марфалагічныя мованезалежныя самастойныя кампаненты, якія выкарыстоўваюць слоўнік S і лінгвістычныя рэсурсы з прэфіксамі Fsubmultiple, Fmultiple, Ssubmultiple, Smultiple. Прызначэнне

першага з іх – ідэнтыфікацыя базавых адзінак вымярэння ў поўнай або скарачанай слоўнікавай форме без прэфіксаў (малюнак 5). Будзем гаварыць, што граф ці падграф «спрацаваў» тады, калі быў знойдзены любы шлях ад яго ўваходу да выхаду пры ўмове выканання ўсіх праверак паміж яго ўваходам і выходам.

Алгарытм працы марфалагічнага кампанента M1 наступны:

1. Для кожнага слова T зыходнага тэксту TT зрабіць дзеянні 2-3, калі словы скончацца, то крок 4.
2. Калі T супадае з любой асновай A з пазнакай Base са слоўніка S , то перанесці на T у зыходны тэкст TT усе граматычныя характарыстыкі A і дадаць маркер Mub , далей крок 1.
3. Калі T супадае з любой асновай A з пазнакай $Mbase$ са слоўніка S , то перанесці на T у зыходны тэкст TT усе граматычныя характарыстыкі A і дадаць маркер $Mbase$, далей крок 1.
4. Вывад – прамаркаваны TT . Канец алгарытму.

Трэба адзначыць, што на выхадзе марфалагічнага кампанента M1 знойдзена адзінка вымярэння атрымае адпаведны маркер, з дапамогай якога пазней можна будзе яе ідэнтыфікаваць у тэксце. Напрыклад, слова $G\zeta$ атрымае маркер Mub , паводле гэтага маркера па запыце $\langle ABBREVIATION+Mub \rangle$ можна будзе яго пабачыць у выніковым канкардансе (малюнак 6).

Далей пачалася праца над марфалагічным кампанентам M2. Ён ідэнтыфікуе адзінкі вымярэння, якія ўтвораны з дапамогай кратных і/ці дольных цэлых прэфіксаў (малюнак 7). У выніку яго працы адзінкі вымярэння могуць быць абазначаны адным з трох маркераў:

- $Mump$ – значыць, што ідэнтыфікавана адзінка вымярэння з кратным прэфіксам;
- $Musp$ – значыць, што ідэнтыфікавана адзінка вымярэння з дольным прэфіксам;
- $Muhr$ – значыць, што ідэнтыфікавана адзінка вымярэння з некалькімі прэфіксамі (напрыклад, *мікрамегафарад*). Такім спосабам утвараць адзінкі вымярэння не прадугледжана ў CI, таму дадзеныя словы трэба абазначыць у тэксце, каб пазней вывесці спіс памылкова ўтвораных адзінак вымярэння.

Алгарытм працы марфалагічнага кампанента M2 наступны:

1. Для кожнага слова T зыходнага тэксту TT зрабіць дзеянні 2-5, калі словы скончацца, то крок 6.
2. Калі пачатковая частка T супадае з любой колькасцю паўтораў любога прэфікса P з $Fmultiple$, а астатняя рэшта T супадае з асновай A з пазнакай Base са слоўніка S , то

перанесці на T у зыходны тэкст TT усе граматычныя характарыстыкі A і дадаць маркер $Mump$, далей крок 1.

3. Калі пачатковая частка T супадае з любой колькасцю паўтораў любога прэфікса P з $Fsubmultiple$, а астатняя рэшта T супадае з асновай A з пазнакай Base са слоўніка S , то перанесці на T у зыходны тэкст TT усе граматычныя характарыстыкі A і дадаць маркер $Musp$, далей крок 1.
4. Калі пачатковая частка T супадае з любой колькасцю паўтораў любога прэфікса P з $Fmultiple$, наступная частка T супадае з любой колькасцю паўтораў любога прэфікса P з $Fsubmultiple$, а астатняя рэшта T супадае з асновай A з пазнакай Base са слоўніка S , то перанесці на T у зыходны тэкст TT усе граматычныя характарыстыкі A і дадаць маркер $Muhr$, далей крок 1.
5. Калі пачатковая частка T супадае з любой колькасцю паўтораў любога прэфікса P з $Fsubmultiple$, наступная частка T супадае з любой колькасцю паўтораў любога прэфікса P з $Fmultiple$, а астатняя рэшта T супадае з асновай A з пазнакай Base са слоўніка S , то перанесці на T у зыходны тэкст TT усе граматычныя характарыстыкі A і дадаць маркер $Muhr$, далей крок 1.
6. Вывад – прамаркаваны TT . Канец алгарытму.

Прыклады працы марфалагічнага кампанента M2 прадстаўлены на малюнках 8 і 9.

Заўважым, што распрацаваныя марфалагічныя кампанеты з-за прастаўленых флексійных класаў у заходным слоўніку асноў ідэнтыфікуюць асновы адзінак вымярэння з любой варыятыўнасцю канчаткаў, напрыклад, *кіламетров* ці *кілометр*. Таму здымаецца неабходнасць першапачатковай нармалізацыі слова да пачатковай формы, каб яго правільна ідэнтыфікаваць. Таксама на выхадзе марфалагічнага кампанента адзінка вымярэння атрымлівае ўсе ўласцівасці, якія былі замацаваныя за яе базавай асновай ў слоўніку S . Дзякуючы гэтаму, словы, напрыклад, *дэкалітрамі*, *наносекундамі*, застаюцца паўнаwartасным назоўнікамі з усімі прыналежнымі ім канчаткамі і характарыстыкамі, нягледзячы на адсутнасць гэтых слоў-асноў у яўным выглядзе ў слоўніку S (малюнак 8).

Наступным этапам стала распрацоўка марфалагічных кампанентаў M3 і M4 для ідэнтыфікацыі адзінак вымярэння, якія адпаведна ўтвораны пры дапамозе цэлай асновы і скарачанага прэфіксу (малюнак 10) альбо – скарачанага асновы і скарачанага прэфіксу (малюнак 11).

Алгарытм працы марфалагічнага кампанента M3 наступны:

1. Для кожнага слова T зыходнага тэксту TT зрабіць дзеянні 2-3, калі словы скончацца, то крок 4.
2. Калі пачатковая частка T супадае з любым прэфіксам P з $S_{multiple}$, а астатняя рэшта T супадае з асновай A з пазнакай $Base$ са слоўніка S , то перанесці на T у зыходны тэкст TT усе граматычныя характарыстыкі A і дадаць маркер $Mump$, далей крок 1.
3. Калі пачатковая частка T супадае з любым прэфіксам P з $S_{submultiple}$, а астатняя рэшта T супадае з асновай A з пазнакай $Base$ са слоўніка S , то перанесці на T у зыходны тэкст TT усе граматычныя характарыстыкі A і дадаць маркер $Mump$, далей крок 1.
4. Вывад – прамаркаваны TT . Канец алгарытму.

Алгарытм працы марфалагічнага кампанента $M4$ наступны:

1. Для кожнага слова T зыходнага тэксту TT зрабіць дзеянні 2-3, калі словы скончацца, то крок 4.
2. Калі пачатковая частка T супадае з любым прэфіксам P з $S_{multiple}$, а астатняя рэшта T супадае з асновай A з пазнакай $Mbase$ са слоўніка S , то перанесці на T у зыходны тэкст TT усе граматычныя характарыстыкі A і дадаць маркер $Mump$, далей крок 1.
3. Калі пачатковая частка T супадае з любым прэфіксам P з $S_{submultiple}$, а астатняя рэшта T супадае з асновай A з пазнакай $Mbase$ са слоўніка S , то перанесці на T у зыходны тэкст TT усе граматычныя характарыстыкі A і дадаць маркер $Mump$, далей крок 1.
4. Вывад – прамаркаваны TT . Канец алгарытму.

На малюнках 12 і 13 прыведзены адпаведныя прыклады выніковых канкардансаў, якія атрымліваюцца пасля адпаведных пошукавых запытаў да прамаркаваных TT кампанентамі $M3$ і $M4$:

- $\langle NOUN+Mump \rangle$ накіраваны на пошук назоўнікаў з цэлымі асновамі са скарачанымі кратнымі прыстаўкамі;
- $\langle ABBREVIATION+Musp \rangle$ накіраваны на пошук абрэвіатурных адзінак вымярэння са скарачанымі асновамі і скарачанымі дольнымі прэфіксамі.

Пасля рэалізацыі слоўніка SS і кампанентаў $M1-M4$ застаецца вызначыць колькасны дэскрыптар, які, часцей за ўсё, стаіць перад адзінкай вымярэння, і разам з ёй утварае $KBAV$. Колькасны дэскрыптар можа выражацца матэматычна (лічбамі, знакамі ці лічбавымі выразамі) альбо лінгвістычна (пры дапамозе лічэбнікаў, колькасных займеннікаў, прыслоўяў і іх спалучэнняў), напрыклад: *450 нанафарад, 15•10⁴(-25) Тэсла, тры молі, шмат градусаў, некалькі секунд* і да т.п.

На дадзеным этапе ажыццёўлены алгарытм

ідэнтыфікацыі колькасных дэскрыптараў, якія перададзеныя сродкамі матэматыкі. Для гэтага быў распрацаваны самастойны сінтаксічны кампанент $S1$ (малюнак 14), які спрацоўвае не толькі на простыя, дзесятковыя і дробавыя лічбы ў розных варыяцыйных пісьмовага запісу, але і на лічбавыя выразы з экспаненцыяльнымі часткамі (малюнак 15). Значым, што дадзены кампанент з'яўляецца мованезалежным.

Для збору усіх маркераў, якія былі расстаўленыя ў тэксце TT праз слоўнік S і марфалагічныя кампаненты $M1-M4$, быў распрацаваны сінтаксічны кампанент $S2$ (малюнак 16). Ён спрацоўвае толькі на выразы з адзінкамі вымярэння, перад якімі стаіць колькасна-лічбавы дэскрыптар, які ідэнтыфікуецца праз убудаваны асобным падграфам сінтаксічны кампанент $S1$. У выніку кожнаму $KBAV$ прысвойваецца маркер $\langle MUEXP \rangle$, паводле яго будуюцца выніковыя мэтавыя канкардансы колькасных выказаў з адзінкамі вымярэння (малюнак 17).

Такім чынам, каб у любым уваходным тэксце ідэнтыфікаваліся колькасныя выразы з адзінкамі вымярэння, якія ўтвораюцца з дапамогай кратных ці дольных прыставак з цэлымі ці скарачанымі асновамі, патрэбна напоўніць універсальную схему з малюнка 1 канкрэтнымі кампанентамі. У выніку будзе пабудаваны шматкампанентны комплекс ідэнтыфікацыі колькасных выказаў з адзінкамі вымярэння (малюнак 18), які складаецца са слоўніка S , марфалагічных $M1-M4$ і сінтаксічных $S1-S2$ кампанентаў.

Заклучэнне

Такім чынам, была пастаўлена і вырашана задача распрацоўкі кампанентаў для ідэнтыфікацыі колькасных выказаў з адзінкамі вымярэння з улікам варыяцыйных спосабаў запісу іх складнікаў. Акрэслены магчымыя сферы ўжывання рашэнняў гэтай задачы: сістэмы сінтэзу маўлення па тэксце, сістэмы пошуку інфармацыі, бібліятэкі, выдавецкія установы, натуральна-маўленчыя інтэрфейсы і інтэлектуальныя сістэмы.

Распрацаваныя алгарытмы ідэнтыфікацыі з'яўляюцца самастойнымі незалежнымі кампанентамі, якія рэалізаваныя ў форме канчатковых аўтаматаў марфалагічных і сінтаксічных граматык у праграме $NooJ$. Кожны з гэтых кампанентаў паасобку ці ўсе кампаненты разам могуць быць убудаваныя ў іншыя сістэмы для дапамогі рашэння адпаведных задач.

Варта зазначыць, што атрыманыя кампаненты граматык $NooJ$ наглядна перадаюць зыходныя алгарытмы і дазваляюць іх хутка мадэфікаваць, маштабаваць і папаўняць лінгвістычнымі рэсурсамі, што важна для павелічэння дакладнасці і паўнаты ідэнтыфікацыі выказаў з адзінкамі вымярэння ў тэкстах.

У будучым плануецца паляпшаць кампаненты праз вырашэнне наступных задач:

- ідэнтыфікаваць не толькі матэматычныя, але і лінгвістычна запісаныя колькасныя дэскрыптары ў спалучэннях з назвамі адзінак вымярэння;

- падключыць да дадзенага комплексу прадстаўлення на канферэнцыі РІНТГ'2013 кампаненты для ідэнтыфікацыі колькасных выказаў з адзінкамі вымярэння ў міжнароднай сістэме СІ, вытворнымі ад СІ і не з СІ;

- ідэнтыфікаваць знак адмоўнасці/дадатнасці перад лікавым дэскрыптарам;

- зменшыць колькасць памылак пры ідэнтыфікацыі шматзначных выказаў, напрыклад, у тых выпадках, калі алгарытм «блытае» адзінкі вымярэння між сабой (*г* для *год*, *грам*, *гадзіна*) ці адзінкі вымярэння з маркамі транспартных сродкаў (*МАЗ-4А 5*, а не *4 амперы*);

- папоўніць слоўнік базавых асноў адзінак вымярэння менш ужывальнымі асновамі, а да графаў дзесятковых прэфіксаў дадаць двойкавыя прэфіксы Міжнароднай электратэхнічнай камісіі.

```

Б, ABBREVIATION+Mbase
байт, NOUN+FLX=БАЙТ+s2+UNAMB+Base
бекерэль, NOUN+FLX=АБАЛЬ+s6+UNAMB+Base
біт, NOUN+FLX=БАЙТ+s2+UNAMB+Base
В, ABBREVIATION+Mbase
Вт, ABBREVIATION+Mbase
ват, NOUN+FLX=БАЙТ+s2+UNAMB+Base
вольт, NOUN+FLX=БАЙТ+s2+UNAMB+Base
г, ABBREVIATION+Mbase
га, ABBREVIATION+Mbase
гг, ABBREVIATION+Mbase
гектар, NOUN+FLX=ГЕКТАР+s5+UNAMB+Base
герц, NOUN+FLX=АМПЕР+s2+UNAMB+Base
год, NOUN+FLX=ГОД+sN+UNAMB+Base
град, ABBREVIATION+Mbase
грам, NOUN+FLX=ГРАМ+s3+UNAMB+Base
Гц, ABBREVIATION+Mbase
гц, ABBREVIATION+Mbase

```

а)

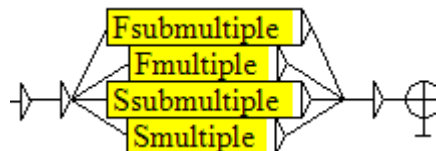
```

ампер, NOUN+FLX=АЛТЫН+s4+UNAMB+Base
А, ABBREVIATION+Mbase
байт, NOUN+FLX=АБАЖУР+s2+UNAMB+Base
бит, NOUN+FLX=АБАЖУР+s2+UNAMB+Base
Б, ABBREVIATION+Mbase
ватт, NOUN+FLX=АЛТЫН+s2+UNAMB+Base
Вт, ABBREVIATION+Mbase
вольт, NOUN+FLX=АЛТЫН+s2+UNAMB+Base
В, ABBREVIATION+Mbase
гектар, NOUN+FLX=АБАЖУР+s5+UNAMB+Base
га, ABBREVIATION+Mbase
герц, NOUN+FLX=ГЕРЦ+s2+UNAMB+Base
Гц, ABBREVIATION+Mbase
год, NOUN+FLX=ГОД+sN+UNAMB+Base
г, ABBREVIATION+Mbase
гг, ABBREVIATION+Mbase
град, ABBREVIATION+Mbase
грамм, NOUN+FLX=АНГСТРЕМ+s3+UNAMB+Base

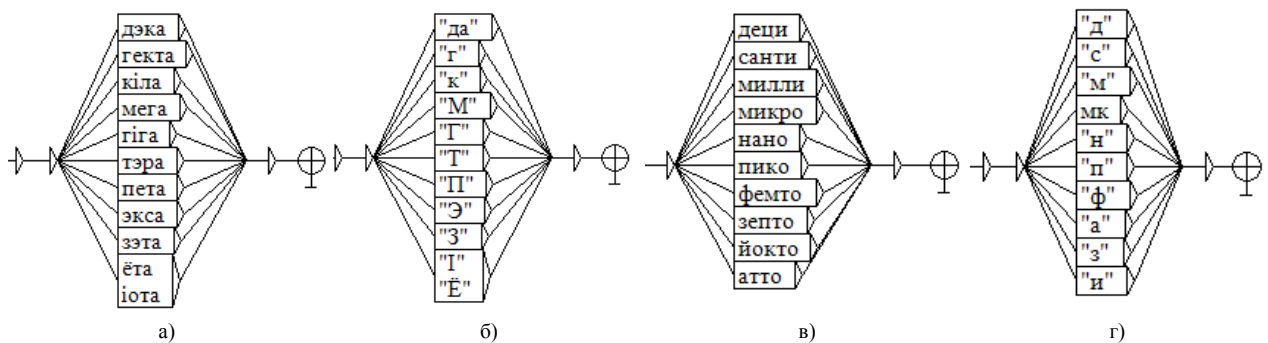
```

б)

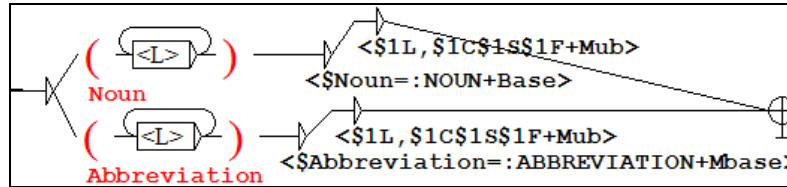
Малюнак 2 – Слоўнікі-рэсурсы базавых асноў адзінак вымярэння для беларускай (а) і рускай (б) моў



Малюнак 3 – Класіфікацыя прэфіксаў у выглядзе канчатковага аўтамата



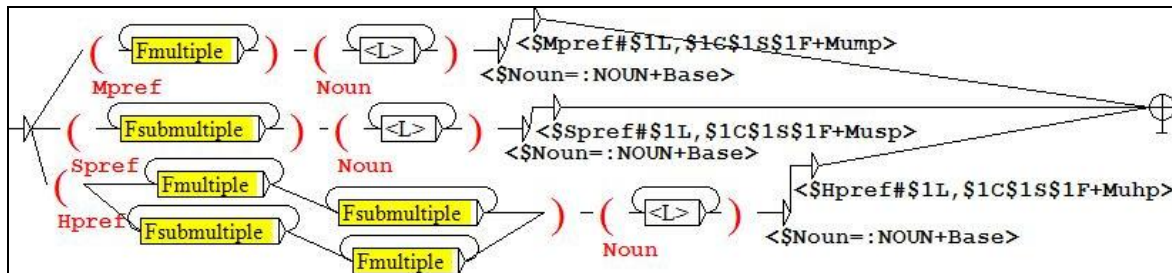
Малюнак 4 – Графы для ідэнтыфікацыі кратных прыставак у поўнай (а) і скарачанай (б) формах для беларускай мовы і дольных прыставак у поўнай (в) і скарачанай (г) формах для рускай мовы



Малюнак 5 – Марфалагічны кампанент M1 для ідэнтыфікацыі базавых адзінак вымярэння з цэлай і скарачанай асновай без прэфіксаў

Before	Seq.	After
ніта на халадзільніку 1-2.4	Тл	- у зазоры магніта тыповага
зблізку 2,4 мЗв у год.	Н	ёсць сіла зямнога прыцягнення
00 км. Кулон (пазначэнне: Кл	, С)	— адзінка вымярэння электрычнага
правадніка пры сіле току 1	А	за час 1 с. ... Комплекс
току 1 А за час 1	с Комплекс ASTER, які складаецца
е атрымліваць кожныя 30	хв	(у штатным рэжыме) здымкі
, 2-200 МЭВ, 2-30 кэВ, 0,1	Гц	-300 кгц, 0-50 кгц; перыядычнасць абна
тэратуры раўняецца 0,025	эВ	. Энергія электрона ў прамянёвай

Малюнак 6 – Выніковы канкарданс базавых адзінак вымярэння па запыце <ABBREVIATION+Mub> для беларускай мовы



Малюнак 7 – Марфалагічны кампанент M2 для ідэнтыфікацыі адзінак вымярэння з цэлымі асновамі і цэлымі кратнымі і/ці дольнымі прэфіксамі

дэкалітрамі.
317
дэкалітр.NOUN+Meaning=Common
+Animation=Inanimate
+Case=Instrumental
+Gender=Masculine+Number=Plural
+s2+Meas=Base+Mump

а)

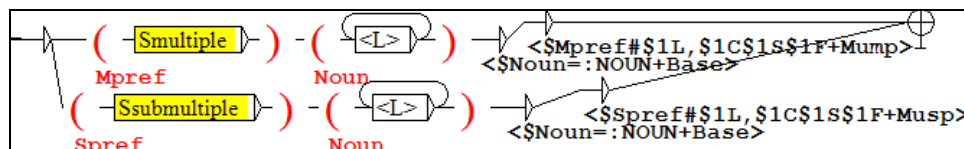
наносекундамі
0
наносекунда.NOUN+ProperCommon=Common
+Gender=Feminine+Animation=Inanimate
+Case=Instrumental+Number=Plural
+s4+Meas=Base+Musp

б)

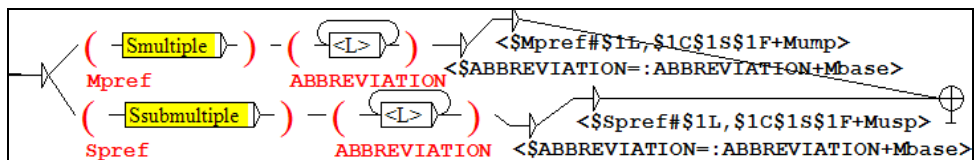
Малюнак 8 – Прыклад анатаванай словаформы для беларускай (а) і рускай (б) моў

Before	Seq.	After
на расстоянии несколько сотен	километров	. Первый вариант ударного
кусков может достигать нескольких	килограммов	. Куски брони поражают
излучения мощностью в сотни	мегаватт	. Проблема в том
составляет уже десятки тысяч	мегагерц	, что соответствует волнам
своих жестких дисках тысячи	гигабайт	информации, третьи подключаются
энергии лазерного излучения порядка	мегаджоуля	(106 Дж) и кпд

Малюнак 9 – Выніковы канкарданс адзінак вымярэння ў поўнай форме па запыце <NOUN+Mump> на прыкладзе рускай мовы



Малюнак 10 – Марфалагічны кампанент M3 для ідэнтыфікацыі адзінак вымярэння з цэлай асновай і скарачаным прэфіксам



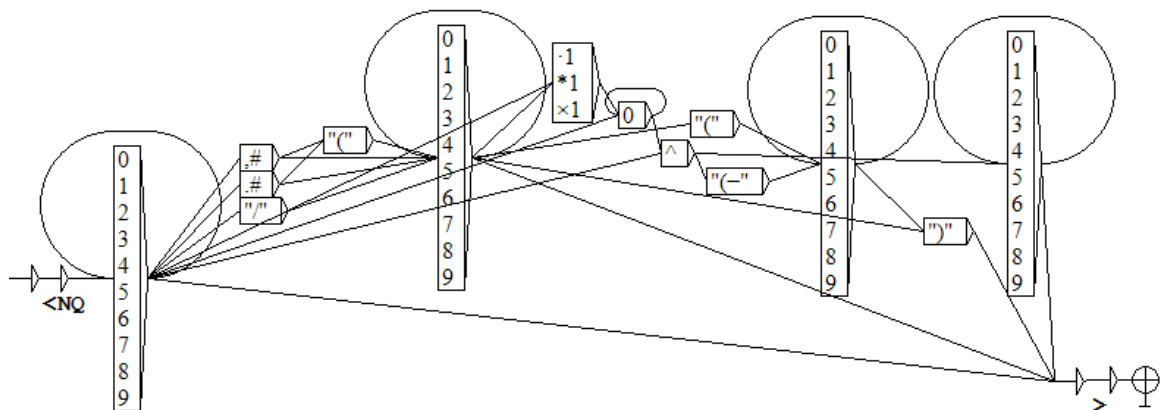
Малюнак 11 – Марфалагічны кампанент M4 для ідэнтыфікацыі адзінак вымярэння са скарачонай асновай і са скарачонам прэфісам

Before	Seq.	After
подтвержденная величина кванта Go ~ 13	кОм	[^] (-1), позволили авторам интерпретиро
уникационных соединений, например, 100	Мбит	/с в стандарте 100BASE
именно 100 000 000 бит/с, а 10	Гбит	/с в стандарте 10GBASE
на которой указан объём 1,44	Мбайт	, на самом деле вмещает
самом деле вмещает лишь 1440	Кбайт	, то есть 1,38 Мбайт в
лишь 1440 Кбайт, то есть 1,38	Мбайт	в обычном понимании. Что
с ГМД порциями по 5	Кбайт	. По данным спутниковых наблюдений
формат. скорость передачи до 64	Мбит	/с) и 137,4 МГц (метровый

Малюнак 12 – Выніковы канкарданс адзінак вымярэння са скарачонамі прэфісамі і цэлымі асновамі па запысе <NOUN+Mump> на прыкладзе рускай мовы

Before	Seq.	After
дакладнасцю да 0,2 % роўная масе 1	дм	³ хімічна чыстай воды пры
звычайна вар'іруецца зблізку 2,4	мЗв	у год. 1 Н ёсць
адзінкі вымярэння: мікрон, роўны 1	мкм	, і ангстрэм (А [*]), роўны
адзінку масы - грам (0,001 кг). 31	мкТл	(3,1×10 [^] (-5) Тл) - напружана
пф, а не 60 нф; 2 000	мкф	, а не 2 мф). Прыстаўкі
пры 0° шыраты (на экватары) 5	мТл	- сіла звычайнага магніта на
нф; 2 000 мкф, а не 2	мф). Прыстаўкі, якія адпавядаю
Ч); спектральнае разрашэнне - 10-20	нм	(ІЧ); радыяметрычнае разр
(пішучь 60 000 пф, а не 60	нф	; 2 000 мкф, а не 2 мф
р ёмістасці = (1/9)·10 [^] (-11) Ф = 1,11...	пф	. Дзесятковыя кратны і дзе

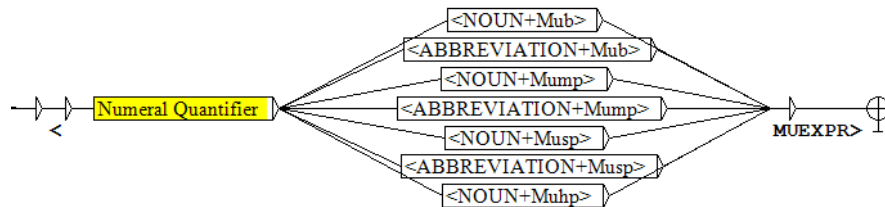
Малюнак 13 – Выніковы канкарданс адзінак вымярэння са скарачонамі асновамі і скарачонамі прэфісамі па запысе <ABBREVIATION+Musp> на прыкладзе беларускай мовы



Малюнак 14 – Сінтаксічны мованезалежны кампанент S1 для ідэнтыфікацыі колькасна-лічбавых дэскрыпцый

Before	Seq.	After
дзеяння сілы. Вага цела масай	102	г (т. е. сіла гравітацыі
для аўтамабіля, тралейбуса, прычэпа; · -	13,5	метра для аўтобуса з дзвюма
энергія фатона чырвонага бачнага святла:	2,61·10 [^] (-19)	Дж.
магнітная індукцыя ў сярэднім складае	5·10 [^] (-5)	Тл, а на экватары (шырата
пры атамным бамбаванні Хірасімы: каля	6·10 [^] 13	Дж. Энергія фатона чырвонага

Малюнак 15 – Прыклады вынікаў пошуку колькасна-лічбавых дэскрыпцый для беларускай мовы, якія атрыманыя пры дапамозе сінтаксічнага кампанента S1



Малюнак 16 – Галоўны сінтаксічны кампанент S2, які вызначае выразы з колькасна-лічбавымі дэскрыптарамі і токенамі з пазнакамі марфалагічных кампанентаў M1-M4

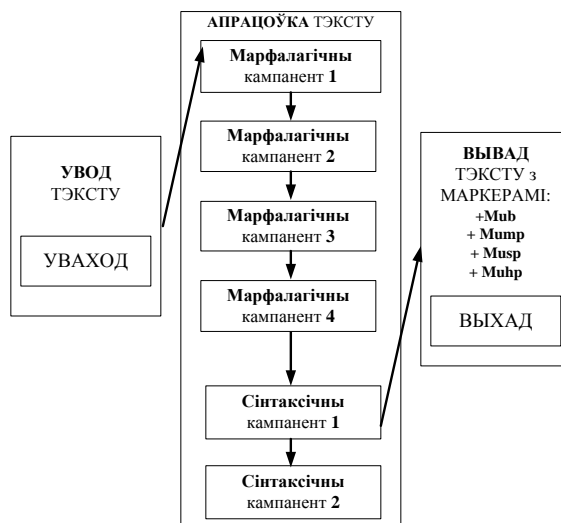
Before	Seq.	After
адзінку масы - грам (0,001 кг).	31 мкТл	(3,1×10 ⁴ (-5) Тл) - напружанасць магнітнага поля
звычайна вар'іруецца зблізка на аперу з сілай	2,4 мЗв	у год. 1 Н ёсць
дамі або нанафарадамі (пішучы	9,81 Н	. Прыбліжэнне, што 1 кг адпавядае
ёмістасць шара з радыусам	60 000 пф	, а не 60 нф; 2 000 мкф
Mbit(альбо проста Mb).	1 сантыметр	, змешчанага ў вакуум. 1 сантыметр
святло ў вакууме за (1 мегабіт	= 1000 ² біт = 10 ⁶ біт = 1000000 біт.
ны дыяпазон - 40 кэВ-3 МэВ, 2-	1 / 299 792 458) секунды	. Метр быў упершыню ўведзены
ай трубы тэлевізара - парадку	200 МэВ	, 2-30 кэВ, 0,1 Гц-300 кгц, 0-50 кгц
энергіі касмічных прамянёў - ад	20 кілаэлектронвольт	. Энергія касмічных прамянёў - ад
	1 мегаэлектронвольта	да1000 тэраэлектронвольтаў.

a)

Before	Seq.	After
организм ток не превышал	1 мА	. На человека токи статического
могут сказать «файл в	100 килобайт	»). При обозначении скоростей тел
противление величиной от 1 до	100 МОм	, чтобы протекающий через челове
кромегафарад пикотеравольт	13 йоттайоктограммов	Каждая строка содержит информ
до 64 Мбит/с) и	137,4 МГц	(метровый диапазон, формат АРТ
евонширский изумруд» массой	1383,95 каратов	. Изумруды выращивают искусств
время жизни мюонов - около	2.2 мкс	- осложняет задачу создания мюон
масса которой оказалась равной	22 фемтограммам	(1 фг = 1·10 ⁴ (-15) г). . Мюоны, как
то они оказались равными:	8.1·10 ²¹ Дж	(уменьшение массы ледников на
: - высота 670 км - наклонение	98,00 град	. Срок активного существования 1 г

б)

Малюнак 17 – Фрагменты вынікаў ідэнтыфікацыі колькасных выказаў з адзінкамі вымярэння пасля апрацоўкі беларуска- (а) і руска- (б) -моўных тэкстаў праз комплекс марфалагічных і сінтаксічных кампанентаў



Малюнак 18 – Выніковая схема шматкампанентнага комплексу ідэнтыфікацыі колькасных выказаў з адзінкамі вымярэння

Бібліяграфічны спіс

[Bekavac, 2009] Bekavac, B. Units of Measurement Detection Module for NooJ/ B. Bekavac// Conference on NooJ 2009. – Tunisia. – 2009. – P.121-127.

[Cunningham, 1999] Cunningham, H. Information Extraction: a User Guide (revised version)/ H. Cunningham// Research Memorandum CS-99-07. – Department of Computer Science, University of Sheffield. – 1999.

[Paskaleva, 2002] Paskaleva E., Angelova, G., Jankova, M., Bontcheva, K., Cunningham, H., Wilks, Y. Slavonic Named Entities in Gate/ E. Paskaleva [et al.]// Research Memorandum CS-02-01. – Department of Computer Science, University of Sheffield. – 2002.

[Mykowiecka, 2007] Mykowiecka, A., Kupść, A., Marciniak, M., Piskorski, J. Resources for Information Ex-traction from Polish texts/A. Mykowiecka// Proceedings of 3rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics. – Poznan. – 2007.

[Silberztein, 2003] Silberztein, M. NooJ Manual [Electronic resource]. – 2003. – Mode of access : <http://www.nooj4nlp.net/NooJManual.pdf>. – Date of access : 01.07.2012.

[Duško, 2007] Duško, V., Krstev, C., Koeva, S. Towards a Complex Model for Morpho-Syntactic Annotation/V. Duško[et al.]// Proceedings of the Workshop on a Common Natural Language Processing Paradigm for Balkan Languages. – Borovets, Bulgaria. – 2007. – P. 65-71.

[Гецэвіч, 2012] Гецэвіч, Ю.С. Ідэнтыфікацыя выразуў з адзінкамі вымярэння ў навукова-тэхнічных і прававых тэкстах на беларускай і рускай мовах / Ю.С. Гецэвіч, А.М. Скопінава // Развитие информатизации и государственной системы научно-технической информации (РИНТИ-2012) : доклады XI Международной конференции (Минск, 15 листопада 2012 г.). – Минск : АПН НАН Беларусі, 2012. – С. 260–265.

[BIPM – SI brochure (8th edition), 2006] BIPM - SI brochure (8th edition) [Electronic resource]. – 2006. – Mode of access : http://www.bipm.org/en/si/si_brochure/. – Date of access : 10.11.2012.

COMPONENTS FOR IDENTIFICATION OF QUANTITATIVE EXPRESSIONS WITH MEASUREMENT UNITS IN BELARUSIAN AND RUSSIAN TEXTS

Hetsevich Y.S. *, Skopinava A.M. *

**United Institute of Informatics Problems,*

National Academy of Sciences,

Minsk, Republic of Belarus

{[yury.hetsevich,skelena777](mailto:yury.hetsevich,skelena777@gmail.com)}@gmail.com

A study of an identifying process of quantitative expressions with measurements units (QEMU) in terms of word formation in thematically distinct texts for Belarusian and Russian is reported here. Models, isolated semantical components and resources for recognition of quantitative expressions with measurements units are described and algorithmically presented. Scopes of application for the obtained practical results are specified.

INTRODUCTION

The urgency of the problem is dictated by the ubiquity of quantitative expressions with measurements units and their enormous variety. Texts which contain QEMU require specific algorithms of identification and processing in such areas as corpora and database management systems, libraries, information retrieval systems, text-to-speech synthesizers, publishing institutions, intellectual systems, natural-language interfaces. Nowadays no search engine provides an interface for building up a search query with QEMU localization function. Observations of other European research workers are too general, their practical results are limited only to the definite language systems. Our research work consists in development of components and linguistic resources in order to identify and classify QEMU with them on the material of hand-crafted text corpora for the Belarusian and Russian languages.

MAIN PART

Dealing with QEMU implies many difficulties, conditioned by a great variety of numeral quantifiers, names of units, their language-dependent origin. It is extremely important to use tools that allow adjusting easily already developed rules and adding new ones. The international computer-linguistic program NooJ is one of such tools. It allows implementing sophisticated algorithms of searching for compound text fragments in

Belarusian and Russian in the form of visual executable graphs, which later form the necessary components. The basic principle for the components is the word-formative classification of QEMU:

1) QEMU with full-form stems and without prefixes (*метр, Герц, Ом*), (*eng. meter, Hertz, Ohm*);

2) QEMU with full-form stems and full-form prefixes (*нанофарады, міліампер*), (*eng. nanofarads, milliampere*);

3) QEMU with full-form stems and shortened prefixes (*кБайт*), (*eng. Kbyte*);

4) QEMU with shortened stems and without prefixes (*Дж, гз, Па*), (*eng. J, ha, Pa*);

5) QEMU with shortened stems and shortened prefixes (*км, дл, гПа*), (*eng. km, dL, hPa*).

Each type is algorithmically described by one of the four morphological components and linguistic resource (figure 2): for QEMU types 1 and 2 see figure 5, for type 3 – figure 7, for type 4 – figure 11, for type 5 – figure 10. Some results are presented in the form of concordances in figures 6, 9, 12, 13. Besides, two syntactic components are developed: the subgraph for numeral quantifiers (figures 14, 15); and the main linking syntactical component that coordinates all the others (figures 17, 18). Thus, by means of the components any text fragment can be morphologically and syntactically analyzed. It is to note, the developed linguistic resources (electronic dictionaries with basic QEMU and subgraphs with SI-prefixes; figures 2-4) are language-dependent, while the morphological and syntactical components are universal.

CONCLUSION

The obtained components are created in the form of finite-state automata through a set of morphological and syntactic grammars within the powerful linguistic processor NooJ. The finite-state automaton demonstrates how the algorithms work and indicate how they can be further updated in order to improve the accuracy. Further improvements are being planned:

- developing components which will identify numeral quantifiers expressed not only by numbers (mathematics objects), but also by numerals (parts of speech);
- providing simultaneous identification of QEMU according to the word formation peculiarities and System International classification;
- disambiguating multiple-valued expressions, for example, in such cases when algorithms "confuse" some units with each other (the same initial letter 'r' for 'год' year, 'грам' gram, 'гадзіна' hour) (r for the year, grams per hour);
- identifying front plus and minus signs, disambiguating minus, hyphen and dash signs;
- updating the base of QEMU with rare ones.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИКО-СИНТАКСИЧЕСКАЯ МОДЕЛЬ ТАТАРСКОГО ПРЕДЛОЖЕНИЯ В КОНТЕКСТЕ РЕЛЯЦИОННО-СИТУАЦИОННОЙ СИСТЕМЫ

Сулейманов Д.Ш., Гатиатуллин А.Р., Вагапов Д.Р.

НИИ «Прикладная семиотика»

Академии наук Республики Татарстан,

Казанский (Приволжский) Федеральный университет

г. Казань, Россия

dvdt.slt@gmail.com

agat1972@mail.ru

В работе описывается семантико-синтаксическая модель для представления структуры простого татарского предложения, основанная на идеереляционно-ситуационной системы. Реляционно-ситуационная система представляет собой взаимосвязанную совокупность сценариев, называемых ситуациями. Сценарий используется для описания структуры простого предложения, где элементами сценария являются синтаксемы.

Ключевые слова: реляционно-ситуационная система; семантические универсалии, синтаксемы, многословные конструкции.

ВВЕДЕНИЕ

В настоящее время в области обработки текстов, написанных на языках тюркского семейства, задача синтаксического анализа предложения продолжает оставаться одной из наиболее актуальных задач, поскольку для них нет такого множества известных разработок в области синтаксиса как для индоевропейских языков.

Эта задача является основной составляющей для многих разработок, которые ведутся в Институте Прикладной семиотики Академии наук Республики Татарстан по компьютерной обработке татарских текстов. Это такие прикладные задачи, как многоязычный интеллектуальный поиск, машинный перевод для близкородственных языков, и ряд других.

Для моделирования использованы следующие ограничения: модели строятся только для простых предложений татарского языка без причастных и деепричастных оборотов.

Семантико-синтаксическая модель татарского предложения представляет собой прагматически ориентированный контекст объектно-предикатной системы, разработанной для описания семантического аспекта структурно-функциональной модели татарской аффиксальной морфемы [Сулейманов, 2003], как оказалось, концептуально весьма близкой к реляционно-ситуационной системе, разрабатываемой группой Г.С. Осипова [Осипов, 1997].

1. Семантико-синтаксическая модель в реляционно-ситуационной системе

Реляционно-ситуационная система состоит из совокупности семантических сценариев, называемых нами семантическими универсалиями. Эти семантические сценарии по сути являются неоднородными семантическими сетями [Осипов, 1997]. Текст предложения в этой модели представляется в виде ситуации, с определенным набором участников и отношений между ними.

Авторами статьи проведена семантическая классификация контекстов в зависимости от типов отношений, участвующих в формировании глубинного смысла данного контекста. Соответственно, формируются семантические сценарии, в которых участвуют объекты и отношения, причем, сценарии строятся от простого к сложному. Простые сценарии – описывают простые типы отношений, сложные сценарии – это комбинации нескольких взаимосвязанных семантических сценариев, называемых ситуациями. Соответственно, семантические сценарии являются теми самыми семантическими универсалиями, которые используются для описания глубинного смысла определенного контекста.

Предлагаемый авторами вариант реляционно-ситуационной системы имеет следующую структуру [Сулейманов, 2003]:

1. Атрибутивные отношения

- a. Пространственные отношения
 - b. Отношения классификации
 - c. Отношения принадлежности
 - d. Признаковые отношения
 - e. Количественные отношения
 - f. Отношения сравнения
2. Темпоральные отношения
- a. Состояние
 - b. Изменение состояния
 - c. Изменение пространственных отношений
 - d. Изменение отношений принадлежности
 - e. Изменение отношений классификации
 - f. Изменение количественных отношений
 - g. Изменение качественных отношений
 - h. Воздействия на объекты
 - i. Понудительные действия
3. Отношения над ситуациями
- a. Каузальные отношения
 - b. Временные отношения
 - c. Порядковые отношения
4. Образ человека
- a. Физическое восприятие
 - b. Речь
 - c. Желание
 - d. Эмоции
 - e. Мышление
 - f. Модальные отношения

Рассмотрим примеры из данной системы. Один из примеров это *Действие* по изменению пространственного отношения. Общий вид схематического представления подобных действий приведен на рисунок 1.

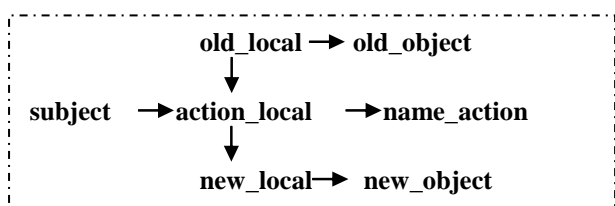


Рис. 1. Действие по изменению пространственного отношения

Подставим в общую схему конкретный пример:

Мин дивангайттым 'Ялегна диван'

Получим заполнение этой схемы, приведенное на рисунке 2.

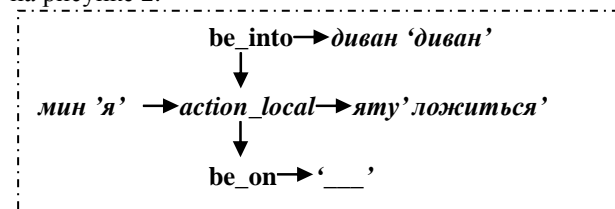


Рис.2. Пример действия по изменению пространственного отношения

Следующий пример – это действие типа воздействие. Схематическое представление действий подобного вида приведен на рис.3.

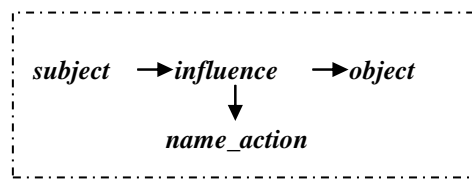


Рис.3. Действие типа воздействие

Заполним данную схему конкретным примером: *Малай кадаккасукты. 'Мальчик ударил по гвоздю'.*

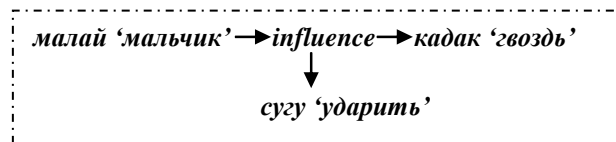


Рис.4. Пример действия типа - Воздействие

Представление текста в виде семантической сети позволяет находить смысловую близость двух разных текстов сравнением соответствующих им семантических сетей. Эта особенность используется как при интеллектуальном поиске, так и как механизм снятия неоднозначностей в задачах машинного перевода.

Участники ситуации в данной семантико-синтаксической модели представляются синтаксемами.

Согласно теории коммуникативной грамматики Золотовой [Золотова, 2004] предполагается тесная связь синтаксиса и семантики в языке, их взаимодействие при выполнении коммуникативной функции языка. Синтаксис играет важную роль в формировании осмысленных предложений. Слова-лексемы, единицы словарного состава языка в совокупности его конкретных грамматических форм, а также возможных конкретных смысловых вариантов, сами не передают смысл высказывания. Синтаксис имеет дело с осмысленными единицами, несущими свой не индивидуально-лексический, а обобщенный, категориальный смысл в конструкциях разной степени сложности. Эти единицы характеризуются взаимодействием морфологических, семантических и функциональных признаков, которые называются синтаксемами.

Синтаксема в тюркских языках характеризуется морфологической структурой – набором аффиксальных морфем. Значение синтаксемы является единицей передаваемого смысла, например, субъект – производитель действия. В конкретном предложении лексическая единица выступает в качестве единицы смысла именно как синтаксема. Таким образом, при работе с текстом необходимо оперировать не лексемами, а синтаксемами.

Общий вид ситуации, описываемой в нашей модели выглядит следующим образом:

Ситуация Q

Предикат: Синтаксема₀;

Роль 1: Синтаксема₁;

...

Роль k: Синтаксема_k.

Приведем пример заполнения этой структуры для ситуации типа Воздействие:

Influence: Синтаксема₀;

Subject: Синтаксема₁;

Object: Синтаксема₂.

Рассмотрим, что представляет собой структура синтаксемы:

Синтаксема: Название

Структура: Набор морфем, корневые и аффиксальные.

Семантика: Семантические ограничения, в виде лексико-семантической группы.

Грамматическая структура синтаксемы представляет собой ее морфемную структуру. Здесь описывается из каких корневых и аффиксальных морфем образована данная синтаксема.

Например:

Поверхностная форма: *урманнарга*

Морфемная структура: урман+ЛАр+ГА

Категориальная структура: N+Plu+Dir.

Структура синтаксемы в своем составе может иметь больше одной корневой морфемы. Это бывает в тех случаях, когда синтаксема представляет собой многословную конструкцию.

2. Синтаксемы, выражаемые многословными конструкциями

Синтаксемы в татарском языке по своей структуре могут быть образованы как отдельными словоформами, так и сочетанием из нескольких словоформ. В нашем случае, синтаксемы из нескольких словоформ - это несвободные словосочетания, составляющие единство с точки зрения семантики, которые в зарубежной литературе обозначаются как многословные выражения—MultiwordExpression, или многословные конструкции.

Под многословными выражениями нами понимаются аналитические конструкции, т.е. все конструкции, значение которых невозможно вывести из значения их составляющих.

В работе А.Н. Баранова [Баранов, 2008] отмечается, что в описании семантики, в целом наиболее правильным и единственно научным является атомистический подход. Однако этот подход не работает для описания семантики многословных выражений.

Компьютерная обработка многословных конструкций, значение которых не определяется из определения значений их составляющих, на

сегодняшний день продолжает оставаться одним из наиболее актуальных вопросов в NLP-задачах. Свидетельством этому служит ежегодный международный семинар Workshop on Multiword Expression, который действует с 2003 года [<http://multiword.sourceforge.net/>].

В нашей модели [Сулейманов, 2012] все многословные конструкции в соответствии со своими структурами делятся на два основных вида:

1. Многословные аналитические формы (МАФ).
2. Многословные аналитические конструкции (МАК).

К первому типу относятся все конструкции, в которых основа первого элемента не привязана к остальным элементам конструкции, то есть, на это слово накладываются не лексические, а только морфологические ограничения. Это означает, что для образования конструкции подобного типа на месте основного слова может быть любое слово определенной категории, но в строго заданной морфологической форме, то есть, должно завершаться на определенную морфему. Вместе с тем, вспомогательное слово имеет строго фиксированную корневую морфему. Это означает, что данная конструкция имеет универсальный характер, соответственно, в нее может быть подставлена корневая морфема такого же морфологического типа.

Ко второму типу относятся конструкции, в которых основы всех элементов конструкции фиксированы.

Структура синтаксемы, представляющей собой многословное выражение, будет выглядеть следующим образом:

Поверхностная форма: *урманга кадәр.*

Морфемная структура: урман+ГА кадәр.

Категориальная структура: N+DirS.

3. Семантически признаки. Лексико-семантические группы

Ситуации состоят из объектов, значений и отношений между ними. Причем, на сочетаемость этих объектов и отношений внутри ситуации накладываются определенные ограничения. Так, например, есть ряд ситуаций, в которых исполнителем действия может быть только человек, в нашей модели это группа отношений "Образ человека". Это означает, что определенную роль в ситуациях этого типа могут заполнять только участники, принадлежащие к определенной лексико-семантической группе.

Лексико-семантическая группа (ЛСГ) – группа слов одной части речи, объединенная одним словом-идентификатором или устойчивым словосочетанием, значение которого полностью входит в значение остальных слов группы и которое может заменять остальные слова в некоторых

контекстах.

Сходство слов, входящих в одну ЛСГ, не исчерпывается только наличием у них общих лексических признаков, а также проявляется в наличии типовой сочетаемости, связанности с определенными структурно-семантическими моделями предложений, в закономерностях соотношений парадигматики внутри группы, в единстве словообразовательных характеристик, с направлением развития многозначности, которое часто носит регулярный характер, и т.д.

Например:

абый сиңа хатында урман турында сөйләде 'в своем письме братрассказал тебе о лесе'

Say: S0: *сө йлә де (рассказал)*

Structure: V+(*)

LSG: излагать

Agent: S1: *абый (брат)*

Structure: N

LSG: человек

Address: S1: *сиң а (тебе)*

Structure: N+Dir(ГА)

LSG: человек

Tool: S3: *хатында (в своем письме)*

Structure: N+Loc([н]ДА)

LSG: средство передачи информации

Maintenance: S4: *урман турында*

Structure: Nтурында

LSG: сущность

Здесь V+(*) означает, что может быть глагол с любым набором аффиксов, морфотактически допустимых для глагола.

ЗАКЛЮЧЕНИЕ

В данной работе предложены семантико-синтаксические модели татарского простого предложения. Реализация семантико-синтаксического анализатора татарского предложения на базе этих моделей требует подготовки целого ряда лингвистических ресурсов. В их число входят базы данных с иерархическими классификациями Лексико-семантических групп, с семантическими сценариями, словарей синтаксем, а также словарей семантических ролей глаголов. На данный момент для татарского языка этих ресурсов не существует, или они находятся на стадии разработки.

Поддержан грантом РФФИ («Разработка моделей и программных средств прикладного синтаксического анализа татарских предложений»), проект № 12-07-00735-а.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Сулейманов и др., 2003] Сулейманов Д.Ш., Гатиатуллин А.Р. Структурно-функциональная модель татарских морфем. Казань, 2003. 212 с.

[Сулейманов и др.(1), 2012] Сулейманов Д.Ш., Гатиатуллин А.Р. Семантические универсалии в системе описания значений морфем // Сборник Трудов конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2011), Минск: Издательство БГУИР, 2012. – С.391-396.

[Сулейманов и др.(2), 2012] Сулейманов Д.Ш., Гатиатуллин А.Р., Галеева А.М., Сулейманова Д.Д. Контекстно-ориентированные семантические универсалии // Сборник трудов Тринадцатой национальной конференции по искусственному интеллекту с международным участием. Белгород. Россия. Т.1. – Белгород: Изд-во БГТУ, 2012. – С.275-284.

[Золотова, 2004] Золотова Г.А., Ониненко Н. К., Сидорова М. Ю. Коммуникативная грамматика русского языка. Институт русского языка РАН им. В. В. Виноградова, М. 2004 – 544 с.

[Баранов, 2008] Баранов А.Н. Против «разложения смысла»: узнавание в семантике идиом // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Бекасово, 4–8 июня 2008 г.). Вып. 7 (14). – М.: РГГУ, 2008. – С.39-44.

[Осипов и др., 2006] Осипов Г.С., Тихомиров И.А., Смирнов И.В. Ехастус – система интеллектуального метапоиска в сети Интернет. // Труды десятой национальной конференции по искусственному интеллекту с международным участием КИИ-2006. М: Физматлит, 2006. т. 3. – С. 859–866.

[Сулейманов и др., 2003] Сулейманов Д.Ш., Гатиатуллин А.Р. Модель многословных конструкций татарского языка: аналитические формы // Журнал «Казанская наука». № 12. 2012. В печати.

SEMANTICO-SYNTACTIC MODEL OF THE TATAR SENTENCE IN THE CONTEXT OF RELATIONAL-SITUATIONAL SYSTEM

Suleymanov D. Sh., Gatiatullin A. R.,
Vagapov D.R.

*Scientific Research Institute "Applied Semiotics"
Academy of Sciences of the Republic of Tatarstan,
Kazan Federal University
Kazan, Russia*

dvd.t.slt@gmail.com

agat1972@mail.ru

In this work the semantico-syntactic model for representation of the structure of the simple Tatar sentence is described. This model is based on the idea of the relational-situational system, representing a system of scenarios, called situations. The scenario is used for the description of the structure of a simple sentence, where the syntaxemes are the elements of the scenario.

Key words: semantic universals, morphemic model, object-predicate system, semantic aspect



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 32.88

МЕТОДЫ ВЕРИФИКАЦИИ ГРАММАТИКИ ЕСТЕСТВЕННЫХ ЯЗЫКОВ ФИННО-УГОРСКОЙ ГРУППЫ НА УРОВНЕ СЕМАНТИЧЕСКОГО ПРЕДСТАВЛЕНИЯ

Килеев В.В. *, Сидоркина И.Г.*

** Поволжский государственный технологический университет,
г. Йошкар-Ола, Республика Марий Эл, Россия*

slavakileev@yandex.ru

igs592000@mail.ru

В работе приводится обзор методов верификации грамматики естественных языков в контексте семантических представлений для реализации их в компьютерной системе. В работе системы верификации грамматики требуется определять семантику верифицируемого слова для проверки ее соответствия контексту. Отличие методов верификации грамматики естественных языков состоит в способе определения семантики верифицируемого слова. Кроме краткого описания каждого из методов, рассматриваются отличия в получении системой семантической информации каждым из методов.

Ключевые слова: верификация грамматики; получение семантики; множество неоднозначности; обработка естественного языка.

ВВЕДЕНИЕ

В области верификации грамматики естественных языков одной из основных целевых задач является выражение смысла (семантики) проверяемого текста. Отличие верификации грамматики от верификации орфографии заключается в том, что во время верификации грамматики проверяется насколько правильно вписывается данное слово в контекст. Таким образом, верификация грамматики осуществляет определение семантики слова и далее проверяет насколько она соответствует семантике всего предложения.

В вопросе верификации текстов естественных финно-угорских языков, проблема верификации орфографии имеет свою реализацию [Килеев и др., 2011]. В то время как проблема определения семантики и верификации грамматики еще только требует своего решения, которое должно учитывать как особенности финно-угорских языков так и особенности построения существующей системы верификации орфографии.

1. Описание проблемы верификации грамматики

Как уже было сказано выше, к вопросам верификации грамматики относится определение правильности употребления слова в контексте. Например, одной из самых частых ошибок является употребление омофонов – слов, произношение которых является одинаковым, но которые пишутся по-разному. Примером омофонов являются слова {quiet, quite} английского языка. Другим примером часто совершаемых ошибок являются слова, которые пишутся почти одинаково, например, {affect, effect}. При этом, в каждом из этих типов ошибок слова образуют множества неоднозначности (confusion sets).

Для правильного принятия решения о правильности употребления того или иного слова в контексте, системе верификации грамматики важно определить смысл каждого слова из множества неоднозначности. Лишь определив это можно сделать правильный выбор. Известно, что все методы верификации грамматики отличаются способами определения семантики каждого слова.

2. Классификация методов

2.1. Общая классификация

Фрагмент общей классификации методов верификации грамматики естественных языков

представлен на рис. 1. Ниже в статье представлено краткое описание каждого из методов и дано пояснение об особенностях получения семантики. По возможности также даются пояснения по основным недостаткам и преимуществам методов.

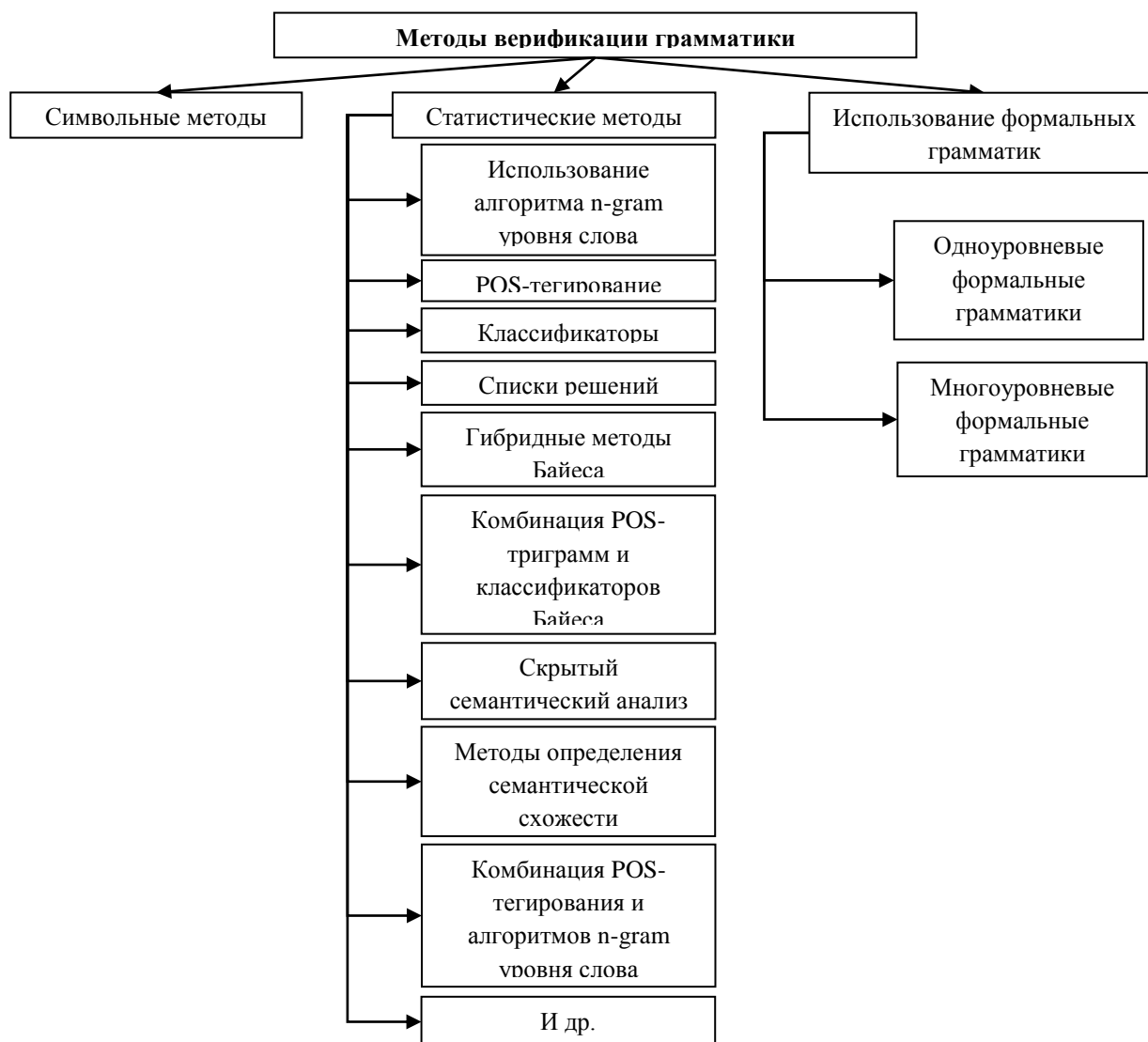


Рис. 1. Фрагмент классификации методов верификации грамматики

2.2. Символьные методы

Суть символьных методов [Naber, 2003] заключается в просмотривании всего предложения целиком и поиске грамматических аномалий. Также данные методы иногда называют методами шаблонов. Для каждого варианта совершаемой пользователем ошибки создается шаблон - представление аномалии, на основе которого осуществляется поиск ошибки по тексту. Исправление на правильный вариант также осуществляется на основе шаблона. Каждый шаблон представляет собой заданную «вручную» семантику слов из множеств неоднозначности. Формат записи шаблонов может быть разный,

• O’Hearn, et al.]. N-gram – это последовательность из n следом идущих элементов. В n-gram уровня слова, в качестве таких элементов

может использоваться, например, расширяемый язык разметки XML.

2.3. Статистические методы

В данную категорию входит большое количество методов, которые схожи между собой по принципу действия. Всем статистические методы требуют для своей работы наличие лингвистический корпусов. На основе этих корпусов строится база данных, позволяющая разделять семантики разных слов. К статистическим методам относятся следующие:

- Различные реализации n-gram уровня слова [Mays, et al., 1991] [Berlinsky-Schine, 2004] [Wilcox-выступают слова из предложения. Семантика слова w определяется тем, какие слова стоят до и после него в последовательности n-gram. Чаще всего

используются либо биграмм – последовательность из двух элементов, или триграмм – последовательность из трех элементов. Для качественной верификации грамматики, требуется лингвистический корпус. Основным недостатком, что даже при использовании лингвистического корпуса очень больших размеров, правильная триграмм-последовательность может отсутствовать в модели. Классический алгоритм триграмм не учитывает несколько ошибок в предложении, он работает при условии, что в предложении допущена только одна ошибка. Существует также улучшенный алгоритм триграмм, который может учесть несколько исправлений в одном предложении. Вместо выбора одного предложения с максимальной вероятностью, выбираются все предложения с вероятностью выше, чем у текущего. И все изменения применяются последовательно. Если изменения конфликтуют между собой, тогда выбирается то, у которого вероятность выше. Доказано, что данный алгоритм работает лучше, чем проверка грамматики в Microsoft Word 2007 [Hirst, 2008].

- POS тегирование [Marshall, 1983] [Garside, et al., 1987]. У каждого слова в предложении определяется его часть речи, которая берется за основу для представления его семантики.

- Классификаторы Байеса [Gale, et al., 1993]. Основная проблема в том, что для качественной работы данного метода современные лингвистические корпуса слишком малы.

- Списки решений (англ. decision list) [Yarowsky, 1994]. Семантика слова разрешается вручную и определяется присутствием слов-якорей вблизи верифицируемого слова. Строятся таблицы употребляемости слов из множества неоднозначности со всеми возможными словами-якорями с указанием частоты встречаемости получившихся словосочетаний в лингвистическом корпусе. Точность определения семантики слова зависит от размера корпуса.

- Гибридные методы Байеса [Golding, 1995]. Основаны на объединении метода списка решений и классификаторов Байеса. Решение о разрешении семантики строится не на основе одного самого яркого признака, а путем подсчета суммы признаков.

- Комбинация POS-триграмм и классификаторов Байеса, иногда еще называется как Три-Байес (Tribayes) [Golding, et al., 1996]. Семантика вначале определяется при помощи POS-триграмм – триграмм в качестве элементов которого выступают не сами слова о их часть речи, если во множестве неоднозначности стоят слова с одинаковой частью речи, тогда семантика определяется при помощи классификатора Байеса.

- Скрытый семантический анализ (англ. latent semantic analysis) [Jones, et al., 1997]. В этом методе строится матрица в строках которой находятся понятия, а в столбцах – тексты корпуса, на пересечение – значение функции, определяющей частоту появления понятия в тексте. После сингулярного разложения данной матрицы,

высчитывается подпространство, содержащее только самые существенные понятия текста. А для получения семантики слова строится его вектор из данного подпространства. За место сингулярного разложения можно также использовать линейный дискриминантный анализ.

- Методы определения семантической схожести [Hirst, et al., 2005] [Budanitsky, et al., 2006]. Все меры семантической схожести строятся на основе специально-размеченного корпуса WordNet.

- Смешанная модель триграмм – это комбинация три-грамм уровня слова и POS триграмм [Fossati, et al., 2007]. Смешанный триграмм – это последовательность трех элементов (e_i, e_{i+1}, e_{i+2}), где элемент e_k является либо k -ым словом в предложении, либо его частью речи. При этом накладывается условие, что в последовательности трех элементов, только один элемент может быть словом, остальные должны быть обозначениями частей речи. В остальном данный метод работает также как и классический n -gram.

- Смешанная модель триграмм с добавлением эмпирических законов для построения множества неоднозначности [Fossati, et al., 2008]. За место редакторского расстояния используется вероятности совершения ошибки. Вероятности получают из специального корпуса ошибок (Birkbeck spelling error corpus). Этот корпус содержит пары слово->неправильно написанное слово. Но проблема этого метода заключается в том, что кроме обычного лингвистического корпуса помеченного POS тегами необходим дополнительный корпус совершаемых пользователем ошибок, который даже для английского языка является не полным.

2.4. Методы, использующие формальные грамматики

Данная категория методов разделяется на две большие подкатегории: одноуровневые формальные грамматики [Heidorn, et al., 1986] и многоуровневые формальные грамматики [Bender, et al., 2004]. В одноуровневых формальных грамматиках все грамматические правила записываются в одной грамматике на одном уровне детализации. Многоуровневые формальные грамматики имеют несколько грамматик разного уровня детализации. Логический вывод идет сразу по всем грамматикам. Если в ходе вывода обнаруживается несоответствие грамматик разных уровней детализации, то считается что в этом случае обнаружена ошибка.

3. Выводы

Хотя статистические методы являются одними из самых часто используемых, они требуют для своей работы наличие больших лингвистических корпусов. Так как для восточных финно-угорских языков лингвистические корпуса на данный момент отсутствуют, то при разработке системы верификации грамматики финно-угорских языков

предложено использовать метод на основе модификации формальных грамматик.

Библиографический список

- [Килеев и др., 2011] Килеев, В.В. Лингвистические особенности архитектуры компьютерной системы верификации орфографии финно-угорских языков / В.В. Килеев, И.Г. Сидоркина // Вестник Волжского университета имени В.Н. Татищева. Научно-теоретический журнал. – Серия «Информатика». – Вып. 18. – Тольятти: Волжский университет им. В.Н. Татищева, 2011г. – С. 115-119.
- [Bender, et al., 2004] Bender E. [и др.] Arboretum: Using a precision grammar for grammar checking in CALL [Конференция] // In Proceedings of the InSTIL/ICALL Symposium. - 2004.
- [Berlinsky-Schine, 2004] Berlinsky-Schine A. Context-based detection of 'real word' typographical errors using markov models [Report] : Technical report. - Ithaca, NY : Cornell University, 2004.
- [Budanitsky, et al., 2006] Budanitsky A. and Hirst G. Evaluating wordnet-based measures of lexical semantic relatedness [Journal] // Computational Linguistics. - 2006. - 32(1). - pp. 13-47.
- [Fossati, et al., 2007] Fossati D. and Di Eugenio B. A mixed trigrams approach for context sensitive spell checking [Conference] // CICLing-2007, Eighth International Conference on Intelligent Text Processing and Computational Linguistics. - Mexico City, Mexico : [s.n.], 2007.
- [Fossati, et al., 2008] Fossati D. and Di Eugenio B. I saw TREE trees in the park: How to correct real word spelling mistakes. [Conference] // LREC 2008, 6th International Conference on Language Resources and Evaluation. - Marrakech, Morocco : [s.n.], 2008.
- [Gale, et al., 1993] Gale W. A., Church K. W. and Yarowsky D. A method for disambiguating word [Journal] // Computers and the Humanities. - 1993. - 26. - pp. 415-439.
- [Garside, et al., 1987] Garside R., Leech G. and Sampson G. The Computational Analysis of English: a corpus based approach [Book]. - [s.l.] : Longman, 1987.
- [Golding, 1995] Golding A. A bayesian hybrid method for context-sensitive spelling correction [Journal] // In The Third Workshop on Very Large Corpora. - 1995. - pp. 39-53.
- [Golding, et al., 1996] Golding A. and Schabes Y. Combining trigram-based and feature-based methods for contextsensitive spelling correction [Conference] // 34th Annual Meeting of the Association for Computational Linguistics. - 1996.
- [Heidorn, et al., 1986] Heidorn G. E. [et al.] The EPISTLE text-critiquing system [Journal] // IBM Systems Journal. - 1986. - 21(3). - pp. 305-326.
- [Hirst, et al., 2005] Hirst G. and Budanitsky A. Correcting real-word spelling errors by restoring lexical cohesion [Journal] // Natural Language Engineering. - 2005. - 11. - pp. 87-111.
- [Hirst, 2008] Hirst G. An Evaluation of the Contextual Spelling Checker [Доклад]. - Toronto, Canada : Department of Computer Science, 2008.
- [Jones, et al., 1997] Jones M. P. and Martin J. H. Contextual spelling correction using latent semantic analysis [Conference] // Fifth Conference on Applied Natural Language Processing. - 1997.
- [Marshall, 1983] Marshall I. Choice of grammatical word-class without global syntactic analysis: tagging words in the LOB corpus // Computers and the Humanities. - 1983. - 17. - pp. 139-150.
- [Mays, et al., 1991] Mays E., Damerau F. J. and Mercer R. L. Context based spelling correction [Journal] // Context based spelling correction. Information Processing and Management. - 1991. - 27(5). - pp. 517-522.
- [Naber, 2003] Naber D. A Rule-Based Style and Grammar Checker. Diploma thesis, University of Bielefeld.
- [Wilcox-O'Hearn, et al.] Wilcox-O'Hearn A., Hirst G. and Budanitsky A. Real-word spelling correction with trigrams: A reconsideration of the Mays, Damerau, and Mercer model. [Conference] // CICLing-2008, 9th.
- [Yarowsky, 1994] Yarowsky D. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French // Proceedings of the 32nd Annual Meeting of the Association for computational Linguistics. - 1994. - pp. 88-95.

NATURAL FINNO-UGRIC LANGUAGE GRAMMAR VERIFICATION METHODS AT THE LEVEL OF SEMANTIC REPRESENTATION

Kileev V.V.* , Sidorkina I.G.*

*Volga State University of Technology, Yoshkar-Ola, Republic of Mari El, Russia

slavakileev@yandex.ru

igs592000@mail.ru

In this paper review of natural language grammar verification methods is considered in the context of semantic representation to implement it in the system. In the work of grammar verification system it is required to determine semantics of the verified word to check it for correspondence to the context. The difference between grammar verification methods is lied in the ways of determining the verified word's semantics. The ways of determining verified word's semantics are examined in addition to the short description of each method

INTRODUCTION

In the field of natural language grammar verification one of the main targeted task is to determine meaning (semantics) of the verified text. The difference between grammar and spell checking is that in grammar checking the correspondence of certain word to the context is determined. Thereby grammar verification determines semantics of the word and then checks whether it corresponds to semantics of the whole sentence.

MAIN PART

The fragment of general classification of grammar verification methods is given on the fig. 1. There are three main types of grammar verification methods: Symbol methods, Statistical methods and methods that use formal grammar.

In symbol methods templates for each mistake type is manually created. The various statistical methods require large text corpora to work correctly.

Methods that use formal grammar may use one grammar to store all grammatical rules and may use several grammars with different level of detail. In the second case the logical inference is done on all grammars simultaneously. The error is detected when the results of the inference on different grammars are not equal.

CONCLUSION

Though statistical methods are widely used nowadays they require large text corpora. Because of the fact that for Eastern Finno-Ugric languages there are no such text corpora it is proposed to use methods based on modified formal grammars in creation of Finno-Ugric languages grammar verification system.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89

ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ МОРФОЛОГИЧЕСКИХ ПРАВИЛ КАЗАХСКОГО ЯЗЫКА В ВИДЕ СЕМАНТИЧЕСКИХ ГИПЕРГРАФОВ

Шарипбаев А.А., Бекманова Г.Т., Муқанова А.С., Ергеш Б.Ж.

** Евразийский национальный университет имени Л.Н. Гумилева, г. Астана, Казахстан*

sharalt@mail.ru

gulmira-r@mail.ru

asel_ms@bk.ru

saturn_banu@mail.ru

В данной работе построены онтологические модели морфологических правил казахского языка в виде семантических гиперграфов. В этих графах вершины представляют семантические признаки (морфологические понятия), а ребра – отношения между этими признаками. С помощью такого гиперграфа описываются структуры словоформ представлены в виде деревьев, которые преобразуются в линейные скобочные записи. Линейные скобочные записи являются формальными моделями морфологических правил. Программная реализация линейных скобочных записей позволили автоматизировать синтез всевозможных словоформ морфологический анализ казахского языка.

Ключевые слова: онтология, морфологические правила, семантический гиперграф, морфологический анализ.

ВВЕДЕНИЕ

Агглютинативные языки (от лат. Agglutinatio — приклеивание) — языки, имеющие строй, при котором доминирующим типом словоизменения является агглютинация («приклеивание») различных формантов (суффиксов или префиксов), причём каждый из них несёт только одно значение.

Агглютинативные языки — тюркские, финно-угорские, монгольские, тунгусо-маньчжурские, корейский, японский, часть индейских и некоторые африканские языки.

Казахский язык относится к тюркской группе языков и характеризуется большим числом словоформ для каждого слова, образованных путем добавления к его концу суффиксов и окончаний. Для него определен строгий порядок аффиксов. Вначале к корню слова прибавляются суффиксы затем окончания множественности, притяжательные окончания, падежные окончания, окончания спряжения [Қазак грамматикасы, 2002].

В настоящее время онтология является мощным и распространенным инструментом моделирования отношений между объектами различных предметных областей. Принято классифицировать онтологии по степени зависимости от задач или

прикладной области, по модели представления онтологических знаний и его выразительным возможностям и другим параметрам [Gruber,1993][Gruber1995]. Прикладные онтологии описывают концепты, которые зависят как от онтологии задач, так и от онтологии предметной области.

Прикладная онтология разрабатывается на основе общих принципов построения онтологий, но с учетом использования в качестве модели представления знаний семантических гиперграфов. Данный формализм позволяют определить онтологию O в виде тройки: (V, R, K) где V – множество понятий проблемной среды (вершины гиперграфа), R – множество отношений между понятиями (дуги и ребра гиперграфа), а K – множество имен понятий и отношений в данной предметной области.

Одним из формальных средств представления знаний является язык семантических гиперграфов, в котором можно в зависимости от типов связей реализовывать классифицирующие, функциональные, ситуационные, структурные сети и сценарии. Семантический гиперграф является расширением семантических сетей, где естественным образом представляются n -арные отношения, которые позволяют задавать не только

атрибуты объектов, но и представлять их структурные, «целостные» описания [Berge, 1985] [Vizing, 2007].

Гиперграф H определяется парой (V, R) , где $V = \{v_i\}$ – множество вершин; $R = \{r_j\}$ – множество ребер; $i = 1, 2, 3..n$; $j = 1, 2, 3..m$; каждое ребро представляет собой пару из элементов множества V , т.е. $r = \{(v_{j_i}, v_{j_s})\}$, $j_s \neq j_t, s, t$ – натуральные числа.

1. Онтологические модели морфологических правил

1.1. Имя существительное

Для имен существительных в качестве семантических признаков начальных форм выступают одушевленность (jand) и неодушевленность (jans) имен существительных. В зависимости от этого признака и определяется траектория словоизменения имени существительного. Имя существительное в казахском языке спрягается (jikt) и изменяется по падежам (sept), а также числам (kopt) и имеет притяжательную форму (taul). На рисунке 1 показана онтологическая модель имени существительного с учетом семантических признаков.

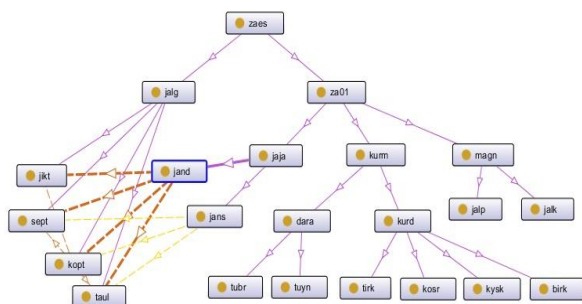


Рисунок 1 – Онтологическая модель имени существительного

Если представить онтологию в виде гиперграфа, то его вершины и ребра будут:

$V = \{jand, jans\}$,

$E = \{(jand, jikt), (jand, sept), (jand, kopt), (jand, taul), (jans, sept), (jans, kopt), (jans, taul)\}$.

Из указанного семантического гиперграфа можно получить формальные правила с помощью скобочной записи. Количество формальных правил для имени существительного 4500. Далее для одного одушевленного существительного с помощью формальных правил автоматически генерируется 93 словоформы (словарных статей), а для неодушевленного существительного генерируется 82 словоформы. Также имя существительное возможно образовать из других частей речи.

Пример словоизменения одушевленного

существительного «бала» - «ребенок» содержит все словоформы данного существительного и их морфологическую информацию, которая содержит в сокращенном обозначении информацию о том в каком числе, падеже находится существительное, от какого лица происходит действие и его принадлежность тому или иному лицу. В таблице 1 приведено изменение существительного «бала» в дательном падеже.

Таблица 1 – Изменение существительного «бала» в родительном падеже.

Правила	Пример	Объяснение
((зежа01)ға)!ба	((бала)ға)	((зе- существительное, жа- одушевленное, 01- признак твердости, ға- окончание дательного падежа))! ба – родительный падеж

1.2. Имя прилагательное

Для имен прилагательных в качестве семантических признаков начальных форм выступают возможности образования из него сравнительной и/или превосходной степени прилагательного, а также выступают такие признаки как относительные (kats) и качественные (sapa), простые (dara), сложные (kurd), производные (tuyn) и т.д. Онтологическая модель имени прилагательного представлена на рисунке 2.

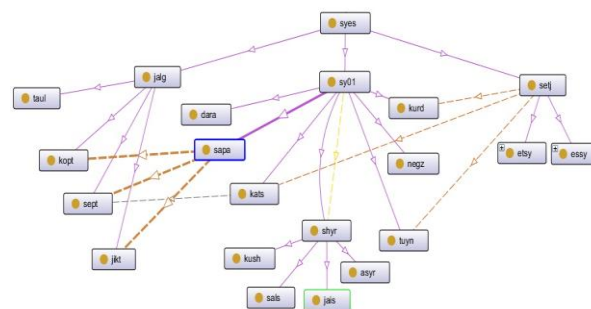


Рисунок 2 – Онтологическая модель имени прилагательного

Определить возможно ли из данного прилагательного образовать сравнительную степень прилагательного и с помощью каких конкретно суффиксов, может только эксперт. Это касается и возможности использования вспомогательных слов при образовании превосходных степеней прилагательных. В данном случае разметку семантических признаков в базе знаний осуществлял специалист-лингвист.

Если представить онтологию в виде гиперграфа, то его вершины и ребра будут:

$V = \{kats, sapa\}$,

$E = \{(kats, sept), (sapa, sept), (sapa, kopt), (sapa, jikt)\}$.

Разряды прилагательного субстантивируясь, изменяется по падежам, спрягается по лицам и принимает аффиксы принадлежности. В таблице 2

приведено изменение прилагательного «ақылды» - «умный».

Таблица 2 – прилагательное «ақылды» - «умный».

Правила	Пример	Объяснение
((сы01)ның)!іл	(ақылды)ны- «умного»	((сы- имя прилагательн ое, 01 - признак твердости), ның - окончание родительного падежа)! іл - родительный падеж
((сы01)мын)!жі11	(ақылды)мын -«я умный»	((сы- имя прилагательн ое, 01 - признак твердости), мын - окончание спряжения ед.ч., 1 лица)! жі11 - спряжение. ед.ч. 1 лицо

С помощью добавления 135 суффиксов образуется имя прилагательное из других частей речи. В результате из 40000 слов словаря генерируется 66000 прилагательных.

1.3. Имя числительное

Имена числительные по составу в казахском языке разделяют на простые (dara) и сложные (kurd). Например, простые: бір - один, он- десять, жүз-сто, мың-тысяча; сложные: он бес- пятнадцать, бес жүз- пятьсот, елу мың - пятьдесят тысяч.

Словообразование сложных имен числительных возможно реализовать автоматически, поскольку в большинстве случаев они образуются из простых числительных путем всевозможных сочетаний разряда числительного и простых числительных.

Пример. Образование сложного числительного «он бір» - «одиннадцать» происходит путем соединения числительных «он» - «десять» и «бір» - «один». Сложное числительное «жүз он бір» - «сто одиннадцать» происходит путем соединения числительных «жүз» - «сто», «он» - «десять» и «бір» - «один» .

По значению имена числительные делятся на шесть групп, которые образуются из простого или сложного числительного путем присоединения соответствующих окончаний или суффиксов. В большинстве случаев все группы числительных автоматически образуются из количественных

путем присоединения суффиксов. Онтологическая модель имени числительного представлена на рисунке 3.

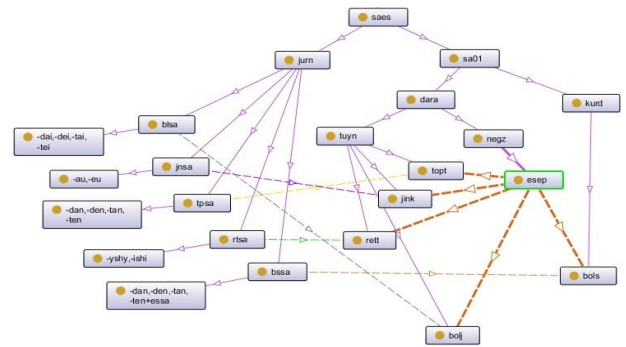


Рисунок 3 – Онтологическая модель имени числительного

Если представить онтологию в виде гиперграфа, то его вершины и ребра будут:

$$V = \{esep, jurm\},$$

$$E = \{(esep, topt), (esep, jink), (esep, rett), (esep, bolj), (esep, bols)\}.$$

1.4. Глагол

Глагол наряду с именем существительным сложная для словообразования и словоизменения часть речи. Словообразование и словоизменение происходит как автоматически, так и по результатам заполненной лингвистом базы знаний.

Необходимо отметить, что глаголы словоизменяются по лицам и числам, а также происходит словообразование новых видов глаголов из других частей речи. Онтологическая модель глагола представлена на рисунке 4.

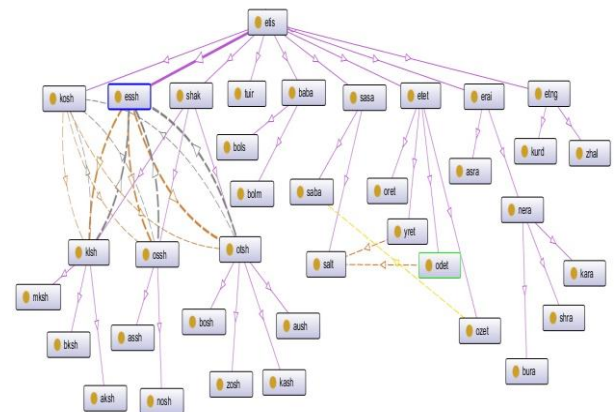


Рисунок 4 – Онтологическая модель глагола

Если представить онтологию в виде гиперграфа, то его вершины и ребра будут:

$$V = \{essh, kosh, saba, salt\},$$

$$E = \{(essh, klsh), (essh, ossh), (essh, otsh), (kosh, klsh), (kosh, ossh), (kosh, otsh), (saba, oset), (salt, yret), (salt, odet)\}.$$

Ниже приведены формальные правила словоизменения и словообразования глагола:

$$(((етот01)п) отыр)мын)!окжі11$$

(((етот01)п) отыр)мыз)!окжі11
 (((етот01)п) отыр)сын)!окжі22
 (((етот01)п) отыр)сындар)!окжі22
 (((етот01)п) отыр)сыз)!окжі2*
 (((етот01)п) отыр)сыздар)!окжі2*
 (((етот01)п) отыр)!окжі33
 (((ет01)т)кыз)дыр)!өг
 (((ет01)т)кыз)ғыз)!өг
 (((ет01)т)тыр)т)кыз)!өг
 (((ет01)т)тыр)т)кыз)дыр)!өг
 (((ет01)т)кыз)дыр)т)?г
 (((ет01)т)тыр)т)кыз)дыр)т)!өг
 ((ет01)т)ты)!өе
 ((ет01)т)қан)!өе
 ((ет01)т)атын)!өе
 ((ет01)т)ар)!ке

С помощью формальных правил образуются новые глаголы и отглагольные формы из других частей речи. В результате из 40000 слов словаря генерируется 395000 глаголов.

1.5. Местоимение

В качестве семантических признаков начальных форм для местоимения является деление в зависимости от его значения на 7 групп: личные, указательные, вопросительные, возвратные, неопределенные, отрицательные, определительные.

Онтологическая модель местоимений представлена на рисунке 5.

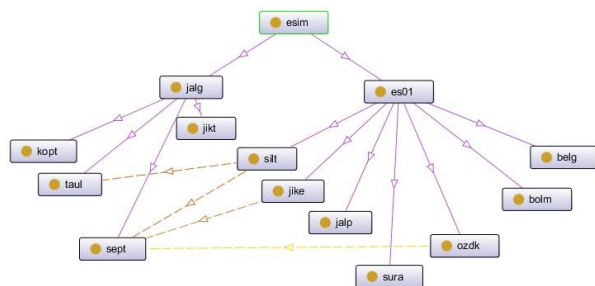


Рисунок 5 – Онтологическая модель местоимений

Личные, указательные и возвратные местоимения склоняются по правилам, определенным для каждой группы. Местоимения других групп склоняются лишь частично, некоторые местоимения вообще не склоняются. Для них определен семантический признак склонения или несклонения. Притяжательная форма (тәуелдік жалғау (taul)) и форма спряжения (жіктік жалғау (jikt)) существует только для некоторых местоимений.

Таким образом, в качестве семантических признаков местоимений можно выделить склонение местоимений, существование притяжательной формы, формы спряжения и принадлежность к той или иной группе местоимений.

Если представить онтологию в виде гиперграфа, то его вершины и ребра будут:

$V = \{silt, jike, ozdk\}$,

$E = \{(silt, taul), (silt, sept), (jike, sept), (ozdk, sept)\}$.

ЗАКЛЮЧЕНИЕ

Построены онтологические модели морфологических правил казахского языка, что позволило записать формальные правила словоизменения и словообразования каждой части речи. Программная реализация этих правил позволила из 40 000 начальных форм слов с размеченными семантическими признаками автоматически генерировать более 3 200 000 словоформ (словарных статей). Тем самым создан морфологический анализатор казахского языка, который сохраняет в памяти только небольшое количество начальных форм слов с размеченными семантическими признаками, а все возможные правильные словоформы получаются автоматически в соответствии с формальными правилами. Результаты работы будут использованы при создании всевозможных программ обработки казахского языка (трансляторов, семантических поисковиков, речевых технологий и др.).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Қазақ грамматикасы, 2002] Қазақ грамматикасы. Фонетика, сөзжасам, морфология, синтаксис// Астана-2002., С. 20-25
- [Gruber, 1993] Gruber T.R. A Translation Approach to Portable Ontology Specifications / Gruber T.R.// Knowledge Acquisition, 1993, P.199-220
- [Gruber, 1995] Gruber T.R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing / Gruber T.R. // International Journal Human-Computer Studies. – 1995, - Vol. 43 - P.907-928
- [Berge, 1985] Graphs and Hypergraphs / Berge C.C.; Elsevier Science Ltd., 1985
- [Vizing, 2007] Vizing V.G. About a coloring of insidator in the hypergraph / Vizing V.G. // Diskretn. Anal. Issled. Oper., Ser., 2007, № 1, P. 40-45

ONTOLOGICAL MODELS OF MORPHOLOGICAL RULES OF KAZAKH LANGUAGE IN THE FORM OF SEMANTIC HYPERGRAPHS

Sharipbayev A.A., Bekmanova G.T., Mukanova A.S., Yergesh B.Zh.

L.N. Gumilyov Eurasian National University, Astana, Kazakhstan

sharalt@mail.ru

gulmira-r@mail.ru

asel_ms@bk.ru

saturn_banu@mail.ru

This paper illustrates construction of ontological models of morphological rules of Kazakh language in the form of semantic hypergraphs. In these graphs the nodes represent the semantic features, and the edges represent the relationship between these features. With such a hypergraph structure of word forms are described in the form of trees, which are converted into linear parenthesis notation. Linear parenthesis notations are the formal models of morphological rules.



УДК 004.823

ПОДХОД К АВТОМАТИЗАЦИИ АНАЛИЗА И СИНТЕЗА ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ

Заболеева-Зотова А.В. *, Орлова Ю.А. *, Петровский А.Б. **

**Волгоградский государственный технический университет,
г. Волгоград, Российская Федерация*

zabzot@gmail.com

yulia.orlova@gmail.com

***Институт системного анализа РАН, г. Москва, Российская Федерация*
pab@isa.ru

В работе рассматривается подход к автоматизации начальных этапов проектирования программного обеспечения. Разработан программный инструментарий, предназначенный для автоматического семантического анализа технической документации, автоматического построения моделей, синтеза структуры и естественно-языкового описания программного обеспечения.

Ключевые слова: проектирование; семантический анализ; синтез структуры; техническая документация.

ВВЕДЕНИЕ

Проектирование программного обеспечения (ПО) представляет собой плохо формализованный трудоемкий процесс, требующий от пользователя глубокого знания предметной области и навыков в проектировании. Наиболее известные из коммерческих продуктов, используемых для проектирования программного обеспечения, предназначены, в основном, для визуализации промежуточных и конечных результатов процесса проектирования. Некоторые из них позволяют полностью автоматизировать последние этапы проектирования: генерация кода, создание отчетной и сопровождающей документации и т.д. В тоже время задача анализа технического задания, выделение на его основе функциональной структуры программы, задача синтеза функциональной структуры программы на основе построенной модели цели практически не решаются. Ни одно из известных CASE-средств не позволяет пользователю проектировать программное обеспечение даже на ограниченном естественном языке [Заболеева-Зотова и др., 2010b].

Повышение эффективности проектирования многокомпонентного программного обеспечения может быть обеспечено за счет автоматизации начальных этапов процесса. Для этого необходимо:

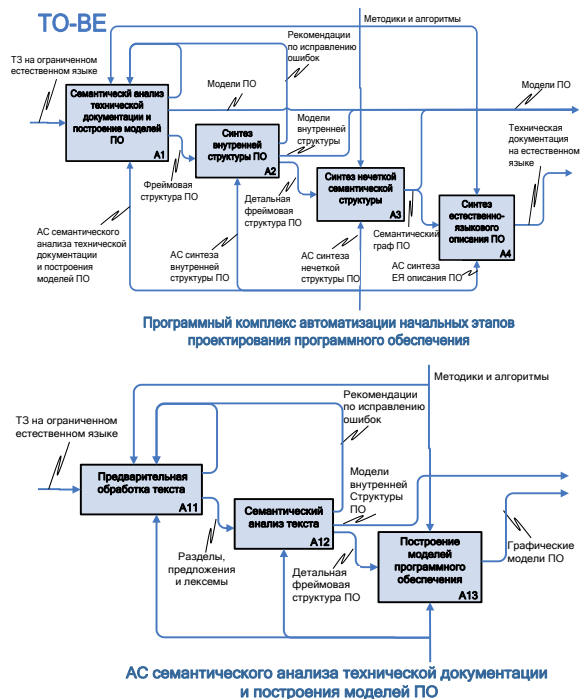


Рисунок 1. Процесс проектирования программного обеспечения

– провести анализ процесса концептуального проектирования многокомпонентного ПО, моделей семантического анализа текста и формализмов представления знаний;

- formalizovat' procedury avtomatizatsii nachalnykh etapov proektirovaniya mnogo-komponentnogo programmnoho obespecheniya;
- razrabotat' algoritmy semanticheskogo analiza tekhnicheskoy dokumentatsii i sinteza struktury mnogo-komponentnogo programmnoho obespecheniya, ispol'zuya apparat multymnozhestv;
- realizovat' razrabotannyye formalizmy, metodiku i algoritmy v vide avtomatizirovannoy sistemy estestvenno-yazykovogo opisaniya PO;
- provesti aprobatsiyu razrabotannykh sredstv pri proektirovaniya mnogo-komponentnogo PO.

Na risунке 1 приведена диаграмма «to be», которая служит иллюстрацией процесса проектирования программного обеспечения с использованием предлагаемой концепции.

1. Анализ существующих CASE-систем

В современных информационных технологиях важное место отводится инструментальным средствам, системам разработки и сопровождения ПО. Эти технологии и среды образуют CASE-системы. Широко известные CASE-системы, такие как BPWin, ERWin, OOWin, Design/IDEF, CASE-Аналитик, Silverrun, Rational Rose, Vantage Team Builder, S-Designor и другие, позволяют частично автоматизировать процесс проектирования программного обеспечения. Однако, как показал анализ, данные системы автоматизируют конечные этапы проектирования программного обеспечения, такие как создание отчетной и сопровождающей документации, генерация кода и т.д. Начальные этапы проектирования – формирование и анализ технической документации, синтез внутренней структуры программного обеспечения – выполняются аналитиком.

Важное значение в процессе разработки ПО имеют также средства спецификации проектов ПО. На начальных этапах проектирования ПО строится функциональная модель системы, которая описывает совокупность выполняемых системой функций. Построение функциональных спецификаций выполняется аналитиком на основе текста технической документации и является в настоящий момент не автоматизированной процедурой.

Как показали специальные исследования, 41% обнаруженных ошибок в сложном проекте программного обеспечения составляют ошибки, сделанные на начальных этапах проектирования. Причиной 56% всех ошибок являются ошибки, сделанные на стадии анализа требований [Заболеева-Зотова и др., 2010a]. Такие ошибки чрезвычайно дорого исправлять.

Проведенный анализ позволяет сделать следующие выводы:

- Автоматизированы в основном отдельные этапы проектирования.

- Отсутствуют методы автоматизированного анализа технической документации.
- Отсутствует автоматизация процесса построения моделей программного обеспечения.
- На начальных этапах проектирования программного обеспечения допускается много ошибок.

Таким образом, автоматизация начальных этапов проектирования программного обеспечения остается открытой задачей, требующей для своего решения создания специальных программных средств.

2. Методика автоматизированного анализа и синтеза технической документации

Часть компонентов технической документации содержит информацию, которая по своему характеру является нечеткой, что обусловлено вариативностью и подвижностью границ языковой нормы и статистическим характером отдельных видов информации. Неточность информации, содержащейся в компонентах технической документации, относится к семантическому и предметно-зависимому уровню и обусловлена сложностью процесса формализации описываемых явлений. Рекомендации по проведению такой формализации формулируются на ограниченном естественном языке (ЕЯ) и могут трактоваться по-разному различными специалистами в зависимости от языковой интуиции человека,

Практически непреодолимой причиной неполноты лингвистической информации является открытость и постоянное развитие ЕЯ: появление новых языковых единиц, изменение свойств существующих единиц и правил их сочетаемости. Такая динамика особенно заметна в подязыках новых предметных областей с неустоявшейся терминологией.

Еще одной причиной неполноты лингвистической информации является наличие нюансов и языковых особенностей отдельных носителей языка, описать и формализовать которые на сегодняшний день не представляется возможным.

Часть информации, содержащейся в технической документации, особенно в техническом задании, может быть ошибочной. Ошибочная информация отличается от неточной тем, что для неточной информации известно, насколько она может не соответствовать действительности. Ошибочная информация может быть маркирована даже как точная, но в то же время полностью противоречить реальной ситуации.

Предлагаемая методика анализа и синтеза технической документации (рис. 2) содержит формализмы, необходимые для представления семантики требований к программному обеспечению на ранних этапах проектирования.

В соответствии с методикой разрабатываемая программная система рассматривается как черный ящик, а предъявляемые к ней требования представляются в виде спецификации функций и определения потоков входных и выходных воздействий.



Рисунок 2. Методика анализа и синтеза технической документации

Методика анализа и синтеза текста технической документации состоит из четырех этапов: семантическая обработка текста, создание фреймовой структуры, создание моделей программного обеспечения, описанного в технической документации, синтез текста описания построенных моделей.

Для семантической обработки текста разработана семантическая модель текста технической документации, включающая требования, сформулированные в виде документов на ограниченном естественном языке. Фреймовая структура является внутренним представлением требований. Модель программного обеспечения задается в виде описания требований на графических языках Data Flow Diagrams и UML.

Семантическая модель текста технической документации содержит разработанные расширенные нечеткие атрибутивные грамматики над фреймовой структурой формальных документов «Техническое задание» и «Технический проект».

Расширенная нечеткая атрибутивная грамматика, необходимая для автоматизированного анализа текста технического задания, определена в виде:

$$AG=(N, T, P, S_0, B, F, A, D(A))$$

где N – конечное множество нетерминальных символов; T – множество терминальных символов, непересекающееся с множеством N ; P – конечное множество правил; S_0 – выделенный символ из множества N , называемый начальным символом; B – множество лингвистических переменных $\beta_{k,i}$, соответствующих терминальным символам T (переменная i на k уровне); F – множество функций принадлежности $f_{k,i}$, определяющих степень принадлежности $\mu_{k,i}$ лингвистических переменных

$\beta_{k,i}$; $A=Asin \cup Asem$ – множество атрибутов, где $Asin$ – синтаксические атрибуты, $Asem$ – семантические атрибуты; $D(A)$ – конечное множество семантических действий.

Лингвистические переменные из множества $B=\{\beta_{k,i}\}_{k,i}$ используемые для анализа текста технического задания описывается следующей пятеркой:

$$\beta_{k,i} = (\beta, T(\beta), U, G, M)$$

где β – название лингвистической переменной (наименование и область применения, основание для разработки, назначение разработки, технические требования к программному изделию, стадии и этапы разработки и т.д.); $T(\beta)$ – языковые выражения. Для лингвистических переменных верхнего уровня они являются лингвистическими переменными, соответствующими терминалам правой части правила. Для лингвистических переменных нижнего уровня – нечеткими переменными, то есть выражениями естественного языка. U – универсум, $T(\beta) \subset U$. G – правила морфологического и синтаксического описания языковых выражений, которые определяют атрибуты $Asin$. M – семантическое правило для лингвистических переменных, которое индуцируется морфологическими и синтаксическими правилами, так как смысл термина в T частично определяется его синтаксическим деревом, и семантическими атрибутами $Asem$.

Методы представления связей между правилами транслируются на язык нечеткой математики. При этом связи представляются нечеткими отношениями, предикатами и правилами, а последовательность преобразований этих отношений – как процесс нечеткого вывода.

Лингвистические переменные верхнего уровня являются составными, то есть включают лингвистические переменные нижнего уровня. Благодаря этому можно построить дерево лингвистических переменных и установить зависимость между ними.

Функции принадлежности из множества $F=\{f_{k,i}\}_{k,i}$ лингвистических переменных $\{\beta_{k,i}\}_{k,i}$ необходимы для построения нечеткого вывода. В частности, каждому правилу грамматики из множества P ставится в соответствие функция принадлежности $f_{k,i}$. Эта двойственная система подстановок используется для вычисления смысла лингвистической переменной.

В грамматике используются следующие синтаксические атрибуты $Asin$: 'Название' – текст, представляющий собой наименование раздела; 'Содержимое' – текст, представляющий собой содержимое раздела; 'Клауза' – фрагмент; 'Клауза ТИРЕ' – фрагмент с тире; 'Группа ГЕНИТ_ИГ' – именительная группа, связанная родительным падежом и др.

Семантические атрибуты, используемые в грамматике, содержат название атрибута $Asem$ и

следующие семантические действия $D(A)$: “Фрейм СИСТЕМА=Создание” – создается фрейм СИСТЕМА; ”Слот НАЗВАНИЕ СИСТЕМЫ=Присваивание” – значение присваивается слоту НАЗВАНИЕ СИСТЕМЫ; “Фрейм ПОТОК ДАННЫХ=Создание” – создается фрейм ПОТОК ДАННЫХ и др.

3. Информационная модель

Рассмотрим систему, описанную в техническом задании, как совокупность входных воздействий, функций и выходных воздействий:

$$S = (N_S, F_S, I_S, O_S),$$

где N_S – название системы, F_S – множество функций системы, I_S – мультимножество входных воздействий, O_S – мультимножество выходных воздействий. Входные воздействия такой системы S представляют собой мультимножество

$$I_S = \{k_I^1 \cdot I_S^1, \dots, k_I^n \cdot I_S^n, \dots, k_I^N \cdot I_S^N\},$$

где функция кратности k_I^n характеризует число одинаковых входов вида I_S^n .

Выходные воздействия такой системы S представляют собой мультимножество

$$O_S = \{k_O^1 \cdot O_S^1, \dots, k_O^m \cdot O_S^m, \dots, k_O^M \cdot O_S^M\},$$

где функция кратности k_O^m характеризует число одинаковых выходов вида O_S^m .

Множество функций системы F_S реализуются подсистемами $F_S^1, \dots, F_S^l, \dots, F_S^L$, каждая из которых описывается следующим образом:

$$F_S^l = (N_F^l, I_F^l, D_F^l, G_F^l, H_F^l, O_F^l),$$

где N_F^l – имя функции F_S^l , I_F^l – мультимножество входных воздействий функции F_S , D_F^l – название действия, выполняемого функцией, G_F^l – объект, над которым выполняется действие, H_F^l – ограничения на функцию, O_F^l – мультимножество выходных воздействий функции F_S .

Мультимножества I_S , O_S , I_F^l , O_F^l представляют потоки данных, которые состоят из единичных порций, имеющих одинаковую структуру:

$$DF = (N_{DF}, D_{DF}, T_{DF}),$$

где N_{DF} – название потока данных, D_{DF} – направление потока данных, T_{DF} – тип данных в потоке.

Операции над мультимножествами [Петровский, 2000], [Петровский, 2003], [Петровский, 2004] можно использовать синтеза структуры программных системам.

При синтезе текста описания построенных моделей программного обеспечения требования к программной системе представляются в виде набора диаграмм потоков данных на графическом языке Data Flow Diagrams. В диаграммах отображается общая структура системы с указанием ее входных, выходных потоков и функций, выполняемых системой, с их входными, выходными

потоками.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 11-07-00398, 12-07-00266, 12-07-00270).

Библиографический список

[Заболеева-Зотова и др., 2010a] Автоматизация семантического анализа текста технического задания: монография. / Заболеева-Зотова А.В., Орлова Ю.А. - Волгоград, ИУНЛ. 2010. - 155 с.

[Заболеева-Зотова и др., 2010b] Автоматизация начальных этапов проектирования программного обеспечения / Заболеева-Зотова А.В., Орлова Ю.А. // Изв. ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах»: межвуз. сб. науч. ст. - Волгоград, ВолгГТУ. 2010. - Вып. 8, № 6. - С. 121-124.

[Петровский, 2000] Петровский А.Б. Комбинаторика мультимножеств. // Доклады Академии наук. 2000, Т.370, №6, С.750-753.

[Петровский, 2003] Петровский А.Б. Пространства множеств и мультимножеств. – М.: Едиториал УРСС, 2003.

[Петровский, 2004] Петровский А.Б. Многокритериальное принятие решений по противоречивым данным: подход теории мультимножеств. // Информационные технологии и вычислительные системы. 2004, №2, С.56-66.

[Розалиев и др., 2010] Применение нейронных сетей и грануляции при построении автоматизированной системы определения эмоциональных реакций человека / Розалиев В.Л., Бобков А.С., Федоров О.С. // Изв. ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах»: межвуз. сб. науч. ст. - Волгоград, ВолгГТУ. 2010. - Вып. 9, № 11. - С. 63-68.

APPROACH TO AUTOMATION OF ANALYSIS AND SYNTHESIS OF TECHNICAL DOCUMENTATION

Zaboleeva-Zotova A.V.*, Orlova Y.A.*,
Petrovsky A.B.**

*Volgograd State Technical University,
Volgograd, Russian Federation
zabzot@gmail.com
yulia.orlova@gmail.com

**Institute for Systems Analysis,
Russian Academy of Sciences,
Moscow, Russian Federation
pab@isa.ru

The paper considers an approach to automation of the initial stages of software design. Program tools, which provide automated semantic analysis of technical documentation, automated construction of models, synthesis of structure and natural language description of the program software, are developed.

The method of analysis and synthesis of text and technical documentation consists of four phases: semantic word processing, creating frame structure, the creation of software models, as described in the technical documentation, the synthesis of the description text of the constructed models.

This work was partially supported by the Russian Foundation for Basic Research (projects 11-07-00398, 12-07-00266, 12-07-00270).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.82

ИДЕНТИФИКАЦИЯ ПРОСТРАНСТВЕННО-ВРЕМЕННЫХ СВЯЗЕЙ МЕЖДУ ВЫСКАЗЫВАНИЯМИ В ЗАДАЧАХ СЕМАНТИЧЕСКОГО АНАЛИЗА ТЕКСТА

Дмитриев А.С., Заболеева-Зотова А.В.

*Волгоградский государственный технический университет,
г. Волгоград, Российская Федерация*

dmitrialeksan@yandex.ru

zabzot@gmail.com

В работе представлена методика интеллектуального анализа, в основе которой лежит обработка пространственной и темпоральной информации в тексте на естественном языке. Описывается применение пространственно-темпоральной логики для идентификации связи между событиями в тексте на естественном языке. Идентификация связи между событиями в тексте производится при помощи методов машинного обучения, в частности логико-марковских сетей.

Ключевые слова: семантический анализ; коммуникативная грамматика; пространственные отношения; темпоральные отношения; комбинированная логика; машинное обучение.

ВВЕДЕНИЕ

Общеизвестно, что в последнее время существует большое количество попыток (и иногда довольно успешных) по созданию современных интеллектуальных систем семантического анализа текстов и речи на естественных языках. Существующие системы с разным успехом справляются с поставленными перед ними задачами, но их общий и главный недостаток состоит в том, что семантика текста строится в основном по конечным высказываниям и предложениям. При этом смысл всего текста рассматривается крайне редко. Такой подход чреват сильным искажением описанной информации в тексте, если, например, какой-либо факт расписан в нескольких последовательных или разбросанных по тексту предложениях.

Естественно, для решения данной проблемы необходимо использовать методику построения сюжетных линий, или по-другому, цепочки событий. Именно события в тексте играют в связке различных предложений ключевую роль. Как правило, события описываются глаголами или отглагольными образованиями. Определив все события текста, используя метод анализа текста на основе коммуникативной грамматики, можно попытаться выстроить между ними причинно-следственную связь, которая позволит достаточно

точно описать семантику всего текста, а не отдельно взятых предложений.

Исходя из вышесказанного, было принято решение о разработке системы, способной распознавать взаимосвязи между событиями в тексте, используя пространственно-временные отношения между данными событиями для повышения семантической связности естественно-языкового (ЕЯ) текста.

1. Способы выражения пространственных и темпоральных отношений в тексте на ЕЯ

1.1. Выражение пространственных отношений

Согласно работе Всеволодовой [Всеволодова и др., 2008], определение пространственных отношений в тексте производится на основе так называемых оппозиций. На первом уровне разбиения всех существующих в русском языке именных локативных групп наиболее важным является отношение локализуемого предмета к локуму, которое основано на нахождении предмета в какой-либо момент времени в пределах локума. В том случае, если предмет в любой момент времени есть, был или будет в пределах локума, то можно говорить о семе сопространственности: «в поля, в поле, из поля, полем, через поле, по полю». Иначе,

если локализуемый предмет не находится, или не будет находиться в пределах локума, то можно говорить об отношении несопространственности: «около поля, у поля, в двух километрах от поля, к полю, мимо поля».

Остальные уровни описания пространственных отношений в русском языке представлены на рисунках 1 и 2.



Рис. 1. Описание отношения соппространственности



Рис. 2. Описание отношения несопространственности в русском языке.

1.2. Выражение темпоральных отношений

В построении временного порядка используется множество грамматических категорий. К ним относятся видо-временные формы глаголов, наречия времени, лексико-семантическая информация и представление о познаваемом мире.

Большинство событий описываются через конструкцию глаголов. Известно, что видо-временные формы глаголов налагают ограничения на временной порядок событий (прошедшее, настоящее, будущее, совершенный и несовершенный вид).

Для событий, описывающихся не через конструкцию глаголов, т.е. с помощью имени существительного, эти параметры не учитываются.

Кроме этих параметров, события могут быть классифицированы различными аспектуальными классами (из которых основными являются состояния, процессы, моментальные события, и события в развитии), способами глагольного действия.

Также они могут быть классифицированы модальностями (т.е. возможность, вероятность или обязанность выполнения действия) – «На этой неделе можно ожидать подъема котировок основных акций».

Полярностями (т.е. положительность или отрицательность выполнения действия) – «Пока не будет остановлен огонь, ни о каком перемирии речи идти не может».

В таблице 1 представлены 4 способа временного поведения глагола в предложениях.

Для связи двух событий используются временные союзы. Они часто появляются в сложных предложениях, и описывают отношения между частями. Например: перед тем как, после, во время, с тех пор, когда, пока, ...

Кроме временных союзов, существуют ещё другие лексические единицы, которые влияют на временной порядок событий. Это временные предлоги, наречие, местоимение и частицы (таблица 2).

Таблица 1 – Способы глагольного действия

Действия	Значение	Приставки или суффиксы к глаголам
Начинательный	Обозначает начало некоторого действия	по- (пойти, полететь, поскакать)
Пердуративный	Обозначает действие, которое заполняет некоторый промежуток времени	про- (проработал, проговорил, просидеть, пролежать, простоять,...)
Финитивный (завершенный)	Обозначает прекращение некоторой деятельности	от- (Отговорил; Отцвели)
Кумулятивный	Обозначает «накопление результата» действия	на- (накупить, натворить, накопить,...)

Таблица 2 – Временные индикаторы

Предлог	с, до, по, перед, под, при, спустя,...
Наречие	сначала, раньше, отныне, скоро, потом
Местоимение	который, какой, кто, что, чей
Частицы	еще, уже, и, а, ...
Другие	как только, когда-то, чтобы,...

2. Коммуникативная грамматика с использованием категорий пространства и времени

В настоящей работе для обработки текстов на естественном языке с целью выявления и извлечения пространственных и темпоральных данных используется коммуникативная грамматика русского языка, разработанная Золотовой Г.А.

Основная идея данной грамматики состоит в том, что синтаксис и семантика тесно взаимосвязаны в рамках анализа смысла предложений.

Анализ текста с использованием коммуникативной грамматики русского языка состоит из следующих этапов:

- Лемматизация
- Морфологический анализ
- Синтаксический анализ
- Семантический анализ (с использованием семантического словаря и лингвистических правил)

Главная характерная особенность синтаксического анализа состоит в том, что для дальнейшего семантического анализа составляются все допустимые деревья синтаксических зависимостей рассматриваемого высказывания. В дальнейшем на этапе семантического анализа производится отбор наиболее подходящего по смыслу дерева зависимостей.

Основным термином коммуникативной грамматики является синтаксема. Она представляет собой слово или словосочетание, значение которого определяется в зависимости от категориального значения слова и морфологической формы, которые в свою очередь реализуются в определенной синтаксической позиции. Смысл предложения (высказывания) определяется совокупностью значений входящих в него синтаксем и отношений между ними.

Для поиска значений синтаксем и отношений между ними используется реляционно-ситуационный анализ. В основе его работы лежит поиск предикатных слов (глаголов и отглагольных образований) на основе семантического словаря, после чего производится поиск значений синтаксем путем сопоставления предикатного слова и составленных синтаксическим анализатором деревьям зависимости, с применением семантического словаря со словарными статьями по синтаксемам и лингвистических правил построения синтаксем. Лингвистические правила и семантический словарь составляются экспертами-лингвистами.

Для реализации реляционно-ситуационного анализа используется предложенная Осиповым Г.С. [Осипов и др., 2008] и др. в интеллектуальной поисковой системе Eхastus неоднородная семантическая сеть с расширенным семейством отношений. Вершинами этой сети являются значения синтаксем, ребрами – отношения между синтаксемами.

После установки значения синтаксем, определяются отношения, в которые входят те или иные пары синтаксем. При этом происходит замыкание – отношения в семантической сети дополняются новыми связями.

У описанной методики анализа естественно-языкового текста, несмотря на то, что анализ на

основе коммуникативной грамматики считается одним из передовых в семантическом анализе, есть существенный недостаток. Этот анализ способен выявить семантику в рамках одного предложения или высказывания. Провести анализ и составить семантическую модель всего текста данная методика не может.

Исходя из этого, было принято решение расширить анализ естественных языков на основе коммуникативной грамматики дополнительными категориями событийности, которые выражаются на всем протяжении текста. Это позволит создавать более полные семантические модели текстов, где раскрывается смысл не каждого отдельно взятого предложения, а, по возможности, большей части анализируемого текста.

Для установления связи между предложениями используются пространственно-темпоральные отношения между предикатными словами и синтаксемами в предложениях. Структурно это представляет собой расширение неоднородной семантической сети.

Например: «Вася опаздывал в школу. Он срезал путь через дворы» (Рисунок 3).

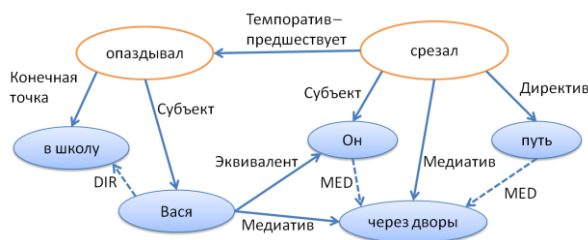


Рис. 3. Пример пространственно-темпоральной связи между семантическими моделями двух предложений

На рисунке сплошными линиями показаны значения синтаксем, а штриховыми отношения между синтаксемами. Для установки семантической взаимосвязи между двумя предложениями используется категория событийности. В тексте повествование идет, как правило, или последовательно, или колеблется относительно определенной сюжетной линии. Соответственно, в тексте прослеживается цепочка событий, попеременно сменяющих друг друга. Для выявления событийности используется изменение пространственных и временных данных, связанных с указанными событиями.

В отличие от построения семантики одного высказывания определение взаимосвязи событий между разными предложениями является более сложной задачей. Воспользуемся для решения данной проблемы механизмом машинного обучения на основе логико-марковских сетей. Для составления событийной цепочки необходимо выявить темпоральные и пространственные характеристики из текста, которые извлекаются с помощью описанного выше метода на основе коммуникативной грамматики, и затем, оперируя

разработанной в рамках данной работы пространственно-темпоральной логикой и методами классификации на основе логико-марковских сетей, выявить принадлежность события к той или иной цепочке событий.

3. Комбинированная пространственно-темпоральная логика

Для определения смысла конечных предложений используется неоднородная семантическая сеть, выявляющая события в рамках одного предложения. Получив семантическую сеть, мы получаем большой набор логических правил, среди которых содержатся правила, описывающие пространственную и временную информацию. Но эта информация довольно слабо связана между собой, поэтому целесообразно использовать комбинированную логику темпоральных и пространственных отношений, для того, чтобы описываемые в предложениях события представляли собой максимально четкую модель в пространстве и времени.

Кроме того, использование комбинированной пространственно-временной (или по-другому топо-темпоральной) логики снижает потребление вычислительных ресурсов. Использование каждой логики в отдельности с дальнейшими попытками связать результат их работы с событиями требует существенных временных затрат. Так или иначе, задача построения эффективной и достаточно полной топо-темпоральной логики требует тщательного исследования, поскольку разные ее комбинации относятся к принципиально разным классам сложности.

3.1. Пространственная логика RCC-8

За основу анализа пространственных отношений взята топологическая логика. Логика топологических пространств в настоящий момент является одним из самых успешных подходов в описании пространственных отношений в искусственном интеллекте. Но до сих пор не было составлено эффективной модели взаимодействия пространственных и временных отношений, поскольку обычное сложение пространственных и временных логик не дает желаемого результата. Связано это в первую очередь с проблемами достижимости и сохранения условий динамических систем. Наиболее полное исследование, посвященное комбинированию пространственных и темпоральных логик, проведено в исследованиях David Gabelaia, Roman Kontchakov, Agi Kurucz, Frank Wolter, Michael Zakharyashev и др.

В работе используется пространственная логика, основанная на пропозициональной логике, в которой унарные предикаты обозначают пространственные объекты, а топологические отношения между ними представляются с помощью внутренних операторов и операторов замыкания,

кванторами общности и существования в пространстве и обычными логическими операциями. Данная логика называется модальной логикой и рассматривается как логика топологических пространств. Обозначается как $S4_u$.

Пространственные термы этой логики представляют собой выражения следующего вида:

$$\tau = p_i \mid \bar{\tau} \mid \tau_1 \cap \tau_2 \mid \tau_1 \cup \tau_2 \mid I\tau \mid C\tau, \quad (1)$$

где p_i пространственные переменные, I и C – операторы включения и замыкания.

Топологическая модель представляет собой структуру следующего вида:

$$Mod = (P, p_0^{Mod}, p_1^{Mod}, \dots), \quad (2)$$

где $P=(U,I)$ – топологическое пространство, а $p_i^{Mod} \subseteq U$ для любого i .

В работе для описания приведения к одной размерности описываемых пространственных объектов используется понятие регулярных замкнутых множеств (или просто «Регионы»).

Для описания регионов используется язык RCC-8. Синтаксис RCC-8 состоит из переменных, обозначающих регионы r, s, \dots и восьми бинарных предикатов:

- $DC(r,s)$ регионы r и s не связаны;
- $EC(r,s)$ r и s внешне связаны;
- и др.

Каждый из этих операторов может использоваться совместно с логическими операциями.

Аргументы RCC-8 предикаты - региональные переменные, интерпретируемые как регулярные замкнутые множества (регионы) топологических пространств. Расширить RCC-8 можно путем представления ее фрагментом логики $S4_u$ (Например):

$$DC(r, s) = (p_r \cap p_s = \emptyset) \quad (3)$$

и др.

3.2. Темпоральная логика LTL

Временная логика в работе представлена линейной темпоральной логикой LTL. Темпоральная логика является подходом к рассуждению о времени, используя темпоральные связи без явного указания количества времени. Самый популярный вариант этой логики это LTL, которая успешно используется в тестировании и верификации программного обеспечения.

Размеченное течение времени для LTL является любой строгой линейной последовательностью $(W, <)$, с временными точками $w \in W$ и отношением предшествования $<$. LTL формулы построены из пропозициональных переменных p_0, p_1 используя логические операторы и темпоральный оператор U -

“пока”. Например xUy означает, что “ x справедливо, пока имеет место быть y ”. Другие темпоральные связки Rf - когда-то в будущем, Ff - всегда в будущем Nf - в следующий момент.

LTL-модель представляет собой структуру следующего вида:

$$Mod = (T, p_0^{Mod}, p_1^{Mod}, \dots), \quad (4)$$

где $T=(W, <)$, а $p_i^{Mod} \subseteq U$ для любого i .

3.3. Комбинированная топо-темпоральная логика

Далее рассмотрим комбинирование пространственной и темпоральной логики. Передвижение пространственных объектов во времени представляется в виде модели “снимок экрана”, т.е. в каждый момент времени фиксируется текущее положение объекта. Топологико-темпоральная модель это пара $Mod=(P,DT)$, где $P=(U, I)$ - топологическое пространство, а DT это множество пространственных точек p в каждый момент времени $n \in N$.

Язык применяемой логики должен придерживаться нескольких следующих правил:

- Язык должен быть способным выразить изменения во времени значений истинности для простых пространственных суждений;
- Язык должен быть в состоянии выразить изменения или развитие пространственных объектов за несколько фиксированных конечных промежутков времени;
- Язык должен быть в состоянии выразить изменения или развитие пространственных объектов на всем протяжении времени.

В работе в целях повышения выразительности языка используется максимальный подход - комбинация всех 3 правил. Этот подход позволяет неограниченно использовать логические, топологические и темпоральные операторы для построения пространственно-темпоральных термов.

Для пространственно-темпорального языка LTL* RCC-8 вводится следующие термы и формулы [Gabelaia и др., 2005]:

$$\begin{aligned} \rho &= CIp \mid CI(\rho_1 U \rho_2), \\ \tau &= \rho \mid \bar{\rho} \mid \tau_1 \cap \tau_2 \mid \tau_1 \cup \tau_2 \mid I\rho, \\ \varphi &= Q(\rho_1, \rho_2) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 U \varphi_2, \end{aligned} \quad (5)$$

где ρ - региональный терм, φ - формула комбинированной логики и Q - обозначает 8 предикатов RCC-8.

4. ИДЕНТИФИКАЦИЯ ПРОСТРАНСТВЕННЫХ И ВРЕМЕННЫХ СВЯЗОК С ИСПОЛЬЗОВАНИЕМ ЛОГИКО-МАРКОВСКИХ СЕТЕЙ

Идентификация временных и пространственных

отношений производится посредством методики машинного обучения. Для этого система на вход получает два типа корпуса текстов. Первый корпус - аннотированный синтаксически размеченный корпус, второй - неразмеченный, по которому вручную расставляются метки с обозначением временных и пространственных отношений. Для обучения системы для выстраивания взаимосвязи между пространственно-временными категориями и событиями в тексте используется механизм логико-марковских сетей, позволяющий оперировать в своих вершинах обычными логическими формулами, представляющими комбинированную пространственно-временную логику, с последующим разбиением сети на атомарные выражения, используемые в обыкновенной Марковской сети. Марковская сеть выполняет задачу классификации по присваиванию тех или иных событий к определенной цепочке событий.

MLNs может ответить на запросы произвольной форме «Какова вероятность того, что формула F_1 верна с учетом того, что формула F_2 верна?» Если F_1 и F_2 являются двумя формулами в логике первого порядка и C является конечным набором констант, включая любые константы, которые появятся в F_1 или F_2 и L является MLNs, тогда:

$$\begin{aligned} P(F_1 \mid F_2, L, C) &= P(F_1 \mid F_2, M_{L,C}) \\ &= \frac{P(F_1 \wedge F_2 \mid M_{L,C})}{P(F_2 \mid M_{L,C})} \\ &= \frac{\sum_{x \in X_{F_1} \cap X_{F_2}} P(X = x \mid M_{L,C})}{\sum_{x \in X_{F_2}} P(X = x \mid M_{L,C})} \end{aligned} \quad (6)$$

где X_{F_i} является множество «миров» (интерпретаций) где выполняется формула F_i

Подставляя в сеть утверждения, которые необходимо проверить (например, принадлежит ли событие к данной цепочке событий) мы, опираясь на обученную сеть, можем получить с некоторой степенью достоверности ответ на принадлежность события к определенной цепочке событий.

Заключение

Описанная методика призвана повысить качество работы существующих семантических систем обработки текстов, в частности интеллектуального поиска, вопрос-ответных систем и др. Она также может использоваться в задачах семантического поиска, поскольку использование расширенных категорий событийности позволяет извлекать смысл не только из конкретных предложений, но также из связанных текстов, в которых прослеживается четкая последовательность событий.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 12-07-00266, 12-07-00270).

Библиографический список

[Бердник, 2007] Бердник, В.Л. Семантический анализ высказываний идентификации сущности / Бердник В.Л., Заболеева-Зотова А.В. // Известия ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах": межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2007. - Вып.3, №9. - С. 43-46.

[Всеволодова и др., 2008] Всеволодова, М.В. Способы выражения пространственных отношений в современном русском языке / М.В. Всеволодова, Е.Ю. Владимирский. - М.: Книжный дом «Либроком», 2009. - 288 с.

[Дмитриев, 2011] Дмитриев, А.С. Извлечение пространственно-временных отношений из текста на естественном языке / Дмитриев А.С. // Интегрированные модели и мягкие вычисления в искусственном интеллекте : сб. науч. тр. VI междунар. науч.-практ. конф. (Коломна, 16-19 мая 2011 г.). В 2 т. Т. 2 / Рос. ассоциация искусственного интеллекта [и др.]. - М., 2011. - С. 883-889.

[Кондрашина и др., 1989] Кондрашина, Е.Ю. Представление знаний о времени и пространстве в интеллектуальных системах. / Е.Ю. Кондрашина, Л.В. Литвинцева, Д.А. Поспелов; под ред. Д.А. Поспелова. - М.: Наука, 1989. - 328 с.

[Осипов и др., 2008] Осипов, Г.С. Реляционно-ситуационный метод поиска и анализа текстов и его приложения / Осипов Г.С., Тихомиров И.А., Смирнов И.В. // Искусственный интеллект и принятие решений / Институт системного анализа РАН. - М., 2008. - №2. - С. 3-10.

[Осипов, 1997] Осипов, Г.С. Приобретение знаний интеллектуальными системами/Г.С. Осипов -М., Наука Физматлит, 1997. - 109с.

[Фамхынг и др., 2008] Фамхынг, Д.К. Применение нечёткой нейронной сети к обработке временной информации в тексте на русском языке / Фамхынг Д.К., Захаров С.С. // AIS'08. CAD-2008. Интеллектуальные системы. Интеллектуальные САПР (пос. Дивноморское, 3-10 сент. 2008 г.): тр. междунар. науч.-техн. конференций / ФГОУ ВПО "Юж. федерал. ун-т" [и др.]. - М., 2008. - Т. 3. - С. 16-22.

[Gabelaia и др., 2005] Gabelaia, D. Combining Spatial and Temporal Logics: Expressiveness Vs. Complexity / David Gabelaia, Roman Kontchakov, Agi Kurucz, Frank Wolter, Michael Zakharyashev // Journal of artificial intelligence research. - 2005. - P.167-243

IDENTIFICATION OF SPACE-TEMPORAL CONNECTIONS BETWEEN STATEMENTS IN PROBLEMS SEMANTIC ANALYSIS TEXT

Dmitriev A.S., Zaboleeva-Zotova A.V.

*Volgograd, Russian Federation Volgograd State
Technical University,
Volgograd, Russia*

dmitrialeksan@yandex.ru

zabzot@gmail.com

This paper presents a methodology mining, which is based on processing of spatial and temporal information in natural language text. Describes the use of spatial-temporal logic to identify the connection between the events in natural language text. Identifying relationships between events in the text produced by the methods of machine learning, in particular Markov-logic networks.

It is well known that in recent years there are a large number of times (and sometimes quite successful) to create a modern intelligent systems of semantic text analysis and natural language speech. Existing systems with varying degrees of success to cope with their tasks, but they are common and the main drawback is that the

semantics of the text is based mainly on the final statements and proposals. In this case, the meaning of the text is not considered. This approach has strong distortion of information described in the text, for example, if a fact is painted in several successive or scattered through the text sentences.

Naturally, to solve this problem, you must use a method of constructing story lines, or in other words, the chain of events. That event in the text play in a bunch of different offers key role. Typically, the events described by verbs or verbal formations. Defining the events of text using text analysis method based on communicative grammar, you can try to build between cause and effect, which will quite accurately describe the semantics of the text, rather than individual sentences.

Based on the foregoing, it was decided to develop a system that can recognize the relationship between the events in the text, using spatio-temporal relationships between these events to enhance the semantic coherence of text in a natural language.

For this system to the input receives two types of corpus. The first building - annotated syntactically marked up body, the second - unmarked on which labels are placed manually with the designation of temporal and spatial relations. For teaching for building relationships between spatial and temporal categories and events in the text, the mechanism of Markov-logic networks, allowing the use of the tops in their usual logical formulas.

The described technique is designed to improve the performance of existing semantic text processing systems, in particular intelligent search, question-answering systems, etc. This system can be used in the problems of semantic search, as the use of advanced categories eventfulness can extract meaning not only of the specific proposals, but also from coherent texts in which there is a clear sequence of events.

This work was partially supported by the Russian Foundation for Basic Research (projects 12-07-00266, 12-07-00270).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.912

АВТОМАТИЗАЦИЯ РЕФЕРИРОВАНИЯ НОВОСТНЫХ ИНТЕРНЕТ-ТЕКСТОВ

Солошенко А.Н., Орлова Ю.А., Дмитриев А.С.

Волгоградский государственный технический университет,

г. Волгоград, Россия

nastyasolan@gmail.com

yulia.orlova@gmail.com

dmitrialeksan@yandex.ru

Реферирование текста в последние годы получило значительную актуальность в связи с развитием Internet и каталогов информационных ресурсов. Данная статья посвящена проблеме составления обзорных статей по текстам интернет-новостей. В статье рассматривается построение и структура новостного текста, этапы его анализа, принципы и методика составления рефератов по новостным статьям.

Ключевые слова: автоматизация реферирования, информационные ресурсы, новостные интернет-тексты, методы реферирования.

Введение

Ежедневно в сети Интернет появляются миллионы страниц новостных текстов. Сотни тысяч людей описывают события и явления, берут интервью, создают посты в блогах. Так, например, только число ежедневных сообщений в Twitter приблизилось к отметке 400 миллионов записей в день, или почти 4,5 тысячи сообщений в секунду. Обработка информационного материала вручную требует колоссальных человеческих ресурсов, трудовых и временных затрат, поэтому и возникла задача создания методики для автоматизации реферирования статей. Потребности в средствах автоматического реферирования и аннотирования испытывают: корпоративные системы документооборота, поисковые машины и каталоги ресурсов Internet, автоматизированные информационно-библиотечные системы, каналы вещания, службы рассылки новостей и др.

В России, как и за рубежом, данному направлению исследований придается очень большое значение [Браславский П. и др., 2008; Гридина Е.А., 2011; Заболеева-Зотова А.В. и др., 2008; Орлова, Ю.А. и др., 2011; Тарасов С.Д., 2008]. В настоящее время известно большое количество систем автоматического реферирования текстов. Среди отечественных это TextAnalyst, Content Analyzer, технологии AOT, RCO, редактор Microsoft Word, МедиаЛингва Аннотатор, система Яндекс

Новостей, среди зарубежных - Extractor , QDA Miner (включающий пакеты WordStat и Simstat), системы Inxight Summarizer (компонент поискового механизма AltaVista), Intelligent Text Miner (IBM). Однако многие ведущие системы разработаны на Западе и ориентированы исключительно на обработку западноевропейских языков, что делает их непригодными для анализа текстов на русском языке. Возможности некоторых отечественных систем ограничены выделением и выбором оригинальных фрагментов из исходного документа и соединением их в короткий текст на основе только статистических методов реферирования.

Для упрощения изучения существующих и создания новых интернет-текстов необходимо представление статей в сжатом виде, но с сохранением их смысла. Поэтому для достижения наиболее оптимального результата необходимо совместить несколько алгоритмов реферирования. Сперва рассмотрим специфику новостных статей, их структуру, а затем уже непосредственно методику автореферирования.

1. Структура новостных интернет-текстов

Структура новостного текста может варьироваться в зависимости от задач и определяется авторами индивидуально. Проанализировав ряд статей, представленных на известных новостных сайтах, таких как сайт газеты

Комсомольская правда, Коммерсант.ru, Эксперт и др., построим обобщенную структуру текста новости (рисунок 1).



Рисунок 1 - «Перевернутая пирамида» - структура новостных текстов

Как правило, для такого типа текстов характерна пирамидальная структура. В ее основу заложен принцип «перевернутой пирамиды», который требует размещение основной информации в самом начале материала и последующее ее раскрытие далее по тексту в деталях. [Электронный ресурс 2]

Данная структура соблюдает все каноны построения новостных статей: в тексте отражена только одна главная тема, статья состоит из введения (основная мысль), развития (вспомогательная полезная информация) и заключения. Как правило, заголовок сразу «цепляет» внимание сутью текста, а продолжение вызывает интерес.

Поэтому, зная особенности построения интернет-текстов, можно определить, какие проблемы ставит автор текста, выделить ключевые факты новостной статьи, определить объекта и субъекта новости. Далее в работе будет рассмотрена методика автоматизации реферирования таких текстов.

2. Обзор существующих систем автоматического реферирования

На международном рынке представлено множество программных продуктов, которые позволяют создавать авторефераты для статей. Так как планируется разработать программный продукт для автоматизации реферирования новостных интернет-текстов на русском языке, то наибольший интерес представляют отечественные аналоги. Их можно оценить по следующим практически важным критериям: поддержка русского языка, входные форматы данных, работа со словарями, функция выделения ключевых фраз, составления связного реферата, свободный доступ (freeware лицензия), функция задания коэффициента сжатия текста, удобство интерфейса.

Рассмотрим наиболее известные системы с точки зрения выделенных критериев.

Content Analyzer предназначен для анализа содержания тематических Web-страниц в реальном времени, динамического выделения списков ключевых слов и словосочетаний, построения автореферата текста документа. Определяет и рассчитывает следующие основные характеристики: частота и вес термина/словосочетания в документе, усредненный вес слов словосочетания/предложения. Работа программы основана на использовании алгоритма TF-IDF для выделения ключевых слов.

TextAnalyst позволяет построить дерево терминов, установив связи между ключевыми понятиями, выделяемыми из текста, предоставляет функцию автоматического реферирования текста - формирования его смыслового портрета в терминах наиболее информативных фраз. Основан на использовании нейронных семантических сетей. [Гридина Е.А., 2011]

Такая технология анализа и поиска текстовой информации как RCO (Russian Context Optimizer) позволяет производить морфологический, синтаксический и семантический анализ текста, их анализ и классификацию, автоматическое реферирование. Технология АОТ же (Автоматическая Обработка Текста) состоит из модулей графематического, морфологического, семантического анализа текста. Обе системы работают со словарями и тезаурусами.

Необходимо также упомянуть о пакете MS Word 2003, 2007, в котором реализована функция AutoSummarize. Предусмотрена возможность указать, какую часть от объема документа должно составлять резюме, однако работа осуществляется только с текстами на английском языке. Отдельно стоит сказать о коммерческой системе Яндекс Новости, позволяющей автоматически группировать данные в новостные сюжеты и составлять аннотации статей на основе кластера новостных документов.

Как наименее функциональный инструмент для реферирования текстов можно отметить редактор MS Word. Составление автореферата основано здесь на поиске наиболее часто встречающихся слов, поэтому не гарантируется получения осмысленного или удобочитаемого текста. TextAnalyst как программный продукт, основанный на алгоритмах создания семантических сетей, проявляет наибольшую гибкость при работе с текстом и составлении его смыслового портрета, как, впрочем, и Content Analyzer. Поэтому наибольшие перспективы в области автоматизации реферирования видятся в развитии взаимодействия и совмещения статистических алгоритмов извлечения ключевых слов и алгоритмов формирования семантической связности блоков текста.

3. Методика реферирования интернет-текстов

Существуют статистические, позиционные, логико-семантические методы реферирования. Каждый метод имеет свои достоинства и недостатки, поэтому стоит использовать их комбинированно.

Решение задачи реферирования текста можно разбить на несколько этапов.

Сначала осуществляется предварительная обработка текста, включающая:

- Перевод текста во внутреннее представление - графематический анализ (выделение из исходного текста лексических единиц - определение границ предложений, слов)
- Нормализацию слов с помощью морфологического анализа. Цель состоит в выделении основ слов, то есть словоформ с отсечёнными окончаниями
- Построение частотного словаря словоформ (как раз после объединения грамматических форм одного и того же слова)
- Вычисление весовых коэффициентов слов, предложений в абзацах в зависимости от функциональной структуры. Выбор наиболее весовых терминов, блоков
- Определение темы текста. На основе заголовка формирование для каждой темы списка предложений, которые ее характеризуют (определение веса каждого предложения путем суммирования частот появления в нем слов и словосочетаний, определяющих темы).

Далее осуществляется выделение семантически связанных групп предложений, удаление незначущих абзацев/ предложений в соответствии с заданным коэффициентом сжатия. По достижении заданного порога сокращения текста процесс реферирования останавливается, получаем конечный реферат, необходимый пользователю.

Методика реферирования представлена ниже в виде блок-схемы (рисунок 2).

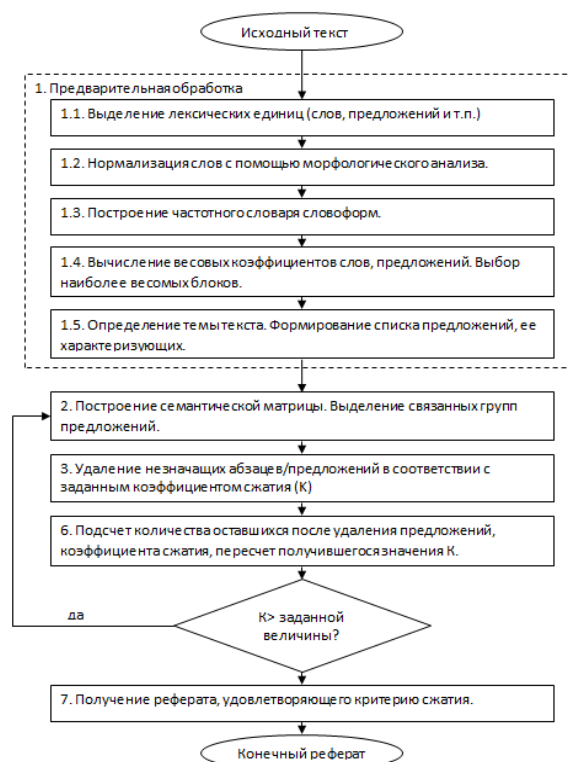


Рисунок 2 - Методика составления реферата

Таким образом, можно выделить следующие необходимые для реферирования текста компоненты:

- модуль графематического анализа;
- модуль морфологического анализа;
- модуль выделения ключевых слов;
- модуль составления связного реферата.

На начальном этапе вырабатывается информация, необходимая для дальнейшей обработки морфологическим и прочими модулями. В задачу графематического анализа входят: разделение входного текста на слова и разделители, выделение предложений из входного текста, абзацев, заголовков и др. Основан на правилах для естественного языка.

Следующим этапом является морфологический анализ, цель которого - построение морфологической интерпретации слов входного текста. Все методы можно поделить на словарные и вероятностно-статистические (без использования словаря). В вероятностно-статистических методах анализ словоформы построен на правилах поиска и сочетания единиц разных лексиконов (лексиконы префиксов, суффиксов, окончаний, основ, баз, корней), приводящий к унификации гипотез. Недостатками являются большой объем лексиконов, плохая работа на малой выборке, отсутствие точных лингвистических методов. Словарный же метод основан на подключении словаря, тезаруса, дает максимально полный анализ словоформы.

Поэтому для данного блока был выбран словарный метод на базе словаря АОТ «Диалинг». Он написан на С++, обладает развитой системой добавления новых слов. Для каждого слова входного текста выдается множество морфологических интерпретаций следующего вида:

- морфологическая часть речи (существительное, глагол, местоимение и т.д.);
- лемма – каноническая форма лексемы (например, существительное в именительном падеже единственного числа, или глагол–инфинитив);
- множество наборов граммем – элементарных описателей, относящих слово-форму к какому-либо морфологическому классу (мр, жр, ср - мужской, женский, средний род; од, но - одушевленность, неодушевленность; ед, мн - единственное, множественное число; им, рд, дт, вн, тв, пр, зв - падежи: именительный, родительный, дательный, винительный, творительный, предложный, звательный).

Структура словаря и алгоритм морфологического анализа представлены на рисунках ниже (рисунки 3 и 4 соответственно).

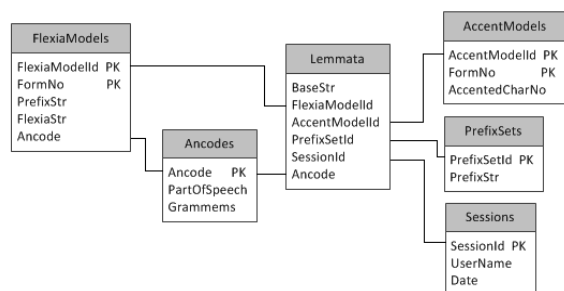


Рисунок 3 - Структура словаря в виде реляционной схемы

Так, таблица Lemmata содержит перечень всех лемм данного словаря, для каждой леммы даны ее свойства: псевдооснова слова (BaseStr), ссылка на набор окончаний (FlexiaModelId), ударений (AccentModelId), приставок (PrefixSetId), на общие граммы данной леммы (Ancestor) (может быть пустым).

Таблица FlexiaModels содержит перечень возможных окончаний всех лемм. Поле PrefixStr содержит префикс данной словоформы, FlexiaStr – окончание словоформы, Ancestor – морфологическую интерпретацию данной словоформы (PartOfSpeech содержит часть речи, а поле Grammems набор граммем). [Сокирко А.В., 2004]

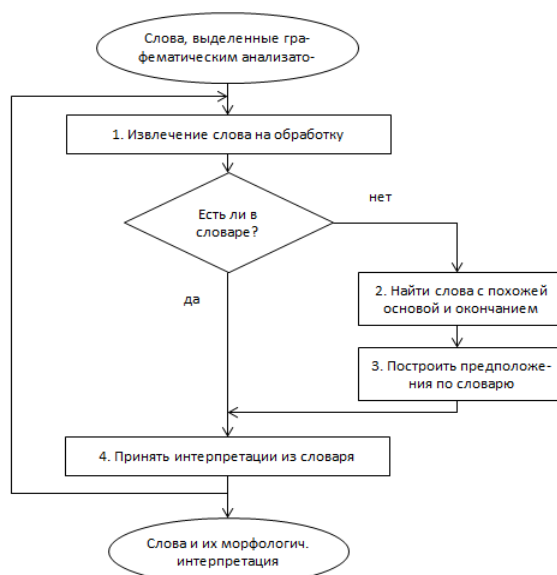


Рисунок 4 - Алгоритмическая модель морфологического анализа

Для выделения ключевых слов в настоящее время разработано множество методов. Будем использовать алгоритмы TF-IDF и C-Value.

TF-IDF (от англ. TF – term frequency, IDF – inverse document frequency) – статистическая мера, используемая для оценки важности слова в контексте документа. Вес некоторого слова пропорционален количеству употребления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции. TF – отношение числа вхождения некоторого слова к общему количеству слов документа.

$$TF = \frac{n_i}{\sum_k n_k} \quad (1)$$

IDF – инверсия частоты, с которой слово встречается в документе, ее учёт уменьшает вес широкоупотребительных слов.

$$IDF = \log \frac{|D|}{|(d_i \supset t_i)|} \quad (2)$$

где $|D|$ – количество документов в корпусе; $|(d_i \supset t_i)|$ – количество документов, в которых встречается t_i (когда $n_i \neq 0$).

Мера TF-IDF является произведением двух сомножителей: TF и IDF.

Метод C-Value позволяет сопоставить каждой извлечённой из текста именной группе значение терминологичности, вычисляемое по формуле:

$$C-Value(a) = \begin{cases} \log_2 |a| \cdot freq(a), & \text{если не вложен} \\ \log_2 |a| \cdot freq(a) - \frac{1}{P(T_a)} \cdot \sum_{b \in T_a} freq(b) \end{cases} \quad (3)$$

где a – кандидат в термины, $|a|$ – длина словосочетания в количестве слов, $\text{freq}(a)$ – частотность a , Ta – множество словосочетаний, содержащих a , $P(Ta)$ – количество словосочетаний, содержащих a .

Путём сортировки списка кандидатов в термины по убыванию значения C -value можно получить список ключевых фраз, наиболее адекватных исходному тексту. [Браславский П. и др., 2008]

Один из самых известных методов реферирования - метод составления выдержек, предполагает акцент на выделение характерных фрагментов (как правило, предложений). Его основу составляет процедура назначения весовых коэффициентов для каждого блока текста в соответствии с такими характеристиками, как расположение этого блока в оригинале, частота появления в тексте, частота использования в ключевых предложениях, а также показатели статистической значимости (ключевые слова). Однако создание итогового документа в данном случае – просто соединение выбранных фрагментов.

Для автореферирования необходимо определять семантическую близость, связность между предложениями текста [Заболеева-Зотова А.В. и др., 2009; Орлова Ю.А., 2007]. Поэтому были предложены следующие методы: выделение фрагментов «объект-действие-субъект» (с использованием POS-таггера AOT для определения частей речи), реализация которого основана на использовании модификации алгоритма LexRank (более подробно описан в [Dragomir R., etc., 2004]), а также ранжирование связанных тематических структур с использованием алгоритма Manifold Ranking, где связная структура текста описывается при помощи матриц [Тарасов С.Д., 2008].

Итак, структурная модель предлагаемого решения, включающая графематический, морфологический модули, модуль выделения ключевых фраз и составления связного реферата приведена на рисунке 5.

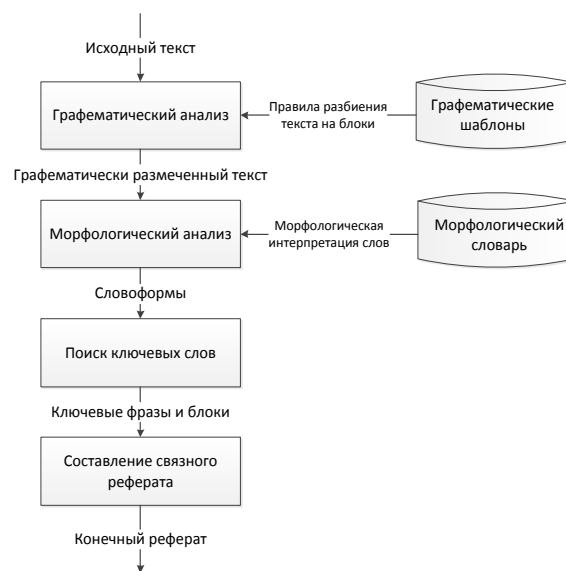


Рисунок 5 - Модель реферирования текста

4. Пример работы алгоритма

Рассмотрим работу алгоритма составления реферата текста на примере небольшого новостного сообщения, взятого из Интернет, заглавием которого является строка «Медики выяснили, в какое время суток человек чувствует себя счастливым»:

«Ученые из США потратили два года на то, чтобы узнать - в какое время суток человек чувствует себя наиболее счастливым? Ученые анализировали сообщения Твиттера. Всего изучались 600 млн. сообщений от 2,5 млн пользователей из более 80 государств. В ходе исследования выяснилось, что пользователи оказались более счастливы утром. К вечеру эмоциональное настроение людей постепенно портится.

Кроме того, выяснилось, что самое хорошее настроение у людей бывает в зимние месяцы: с декабря по январь, сообщает epidemiolog.ru. Однако, надо уточнить: счастливый период заканчивается, скорее всего, не в конце января, а 16-го. Это третий понедельник января, который считается самым депрессивным днем в году.

К такому выводу пришел британский психолог Клифф Арнэлл из Университета Кардиффа. Он вывел сложную формулу, которая учитывает отвратительную, как правило, погоду, то, что праздники прошли и надо браться за работу, надо как-то планировать свое будущее. Не радует и то, что до следующих праздников далеко.

Путем деления и умножения специальных коэффициентов, отражающих вышеозначенное состояние человеческой души в середине января, ученый якобы и получил дату, соответствующую третьему понедельнику» [Электронный ресурс 1]

В результате применения алгоритма предполагается получить следующие результаты:

Ключевые фразы: надо, ученые, выяснилось, настроение, людей, человек, Твиттер, узнать, время суток, чувствует, наиболее счастливым, третий понедельник.

Автореферат текста:

«Ученые из США потратили два года на то, чтобы узнать - в какое время суток человек чувствует себя наиболее счастливым? В ходе исследования выяснилось, что пользователи оказались более счастливы утром. Однако, надо уточнить: счастливый период заканчивается, скорее всего, не в конце января, а 16-го. Это третий понедельник января, который считается самым депрессивным днем в году. К такому выводу пришел британский психолог Клифф Арнэлл из Университета Кардиффа. Он вывел сложную формулу, которая учитывает отвратительную, как правило, погоду, то, что праздники прошли и надо браться за работу, надо как-то планировать свое будущее».

Заключение

Таким образом, совмещение нескольких алгоритмов составления реферата позволит улучшить качество реферирования. В условиях нынешнего информационного века, с огромным количеством новостных сообщений в сети интернет, применение таких технологий является необходимостью, так как значительно ускоряет обработку повседневной информации.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 12-07-00266, 12-07-00270).

Библиографический список

- [Электронный ресурс 1] Главные события политической жизни/ KP.RU [Электронный ресурс]. – Режим доступа: <http://www.kp.md/daily/25982.5/2915545/>.
- [Браславский П. и др., 2008] Браславский П., Соколов Е. Сравнение пяти методов извлечения терминов произвольной длины// Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог». Вып. 7 (14). – М.: РГТУ, 2008. - С. 67-74.
- [Гридина Е.А., 2011] Гридина Е.А. Анализ алгоритмов автоматического реферирования текстов/ Восточно-Европейский журнал передовых технологий. – 2011.
- [Заболеева-Зотова А.В. и др., 2008] Заболеева-Зотова, А.В. Автоматизация семантического анализа текста технического задания / Заболеева-Зотова А.В., Орлова Ю.А. // Системные проблемы надёжности, качества, мат. моделирования, информ. и электронных технологий в инновационных проектах: (Инноватика-2007): матер. междунар. конф. и Рос. науч. школы / Рос. акад. надёжности [и др.]. - М., 2007. - Ч.1. - С. 78-79
- [Заболеева-Зотова А.В. и др., 2009] Заболеева-Зотова, А.В. Formalization of text analysis of a technical specification / Заболеева-Зотова А.В., Орлова Ю.А. // Congress on intelligent systems and information technologies (AIS-IT'09), Divnomorskoe, Russia, September, 3-10 : proc. / Южный федеральный ун-т [и др.]. - М., 2009. - Vol. 4. - С. 62.

[Орлова, Ю.А. и др., 2011] Орлова, Ю.А. Обзор современных автоматизированных систем распознавания эмоциональных реакций человека / Орлова Ю.А., Розалиев В.Л. // Изд. ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах". Вып. 10 : межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2011. - № 3. - С. 68-72.

[Орлова Ю.А., 2007] Орлова, Ю.А. Подсистема предварительной обработки текста / Орлова Ю.А., Заболеева-Зотова А.В. // Технологии Microsoft в теории и практике программирования: тр. IV всерос. конф. студ., аспирант. и мол. уч., 2-3 апр. 2007: Центральный регион, Москва: [тез. докл.] / Моск. авиац. ин-т (гос. техн. ун-т) [и др.]. - М., 2007. - С. 187-188.

[Сокирко А.В., 2004] Сокирко А.В. Морфологические модули на сайте www.aot.ru/ Диалог, 2004 г.

[Тарасов С.Д., 2008] Тарасов С. Д. Алгоритм ранжирования связанных структур в задачах автоматического составления обзорных рефератов новостных сюжетов.// RuSSIR'2008, труды Второй Российской конференции молодых ученых по информационному поиску. – Таганрог: Изд-во ТТИ ЮФУ, 2008. – С. 90-100.

[Электронный ресурс 2] Шардаков Д. Структура текста: как ее создать и сделать текст удобным для восприятия/ Shard-Copywriting.Ru [Электронный ресурс]. – Режим доступа: <http://shard-copywriting.ru/copywriting-basics/glavnyiy-printsip-kopiraytinga-perevernutaya-piramida>.

[Dragomir R. et al., 2004] Dragomir R. Radev, Gunes Erkan. LexRank: Graph-based Lexical Centrality as Saliency in Text Summarization// Journal of Artificial Intelligence Research, 2004.

AUTOMATIZATION OF INTERNET NEWS TEXT ABSTRACTING

Soloshenko A.N., Orlova Y.A., Dmitriev A.S.

*Volgograd State Technical University,
Volgograd, Russia*

nastyasolan@gmail.com

yulia.orlova@gmail.com

dmitrialeksan@yandex.ru

Text abstracting received in recent years a considerable urgency in connection with development of Internet and catalogs of information resources. This article is devoted to a problem of an automated abstracting – drawing up reviews in texts of Internet news. Construction and structure of the news text, stages of its analysis, principles and methods of drawing up of author's abstracts under news articles are considered.

Keywords: automated abstracting, information resources, Internet news texts, methods of abstracting.

This work was partially supported by the Russian Foundation for Basic Research (projects 12-07-00266, 12-07-00270).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.932

АВТОМАТИЗИРОВАННАЯ СИСТЕМА СОЗДАНИЯ РУКОПИСНЫХ ДОКУМЕНТОВ

Константинов В.М., Розалиев В.Л., Дианов И.А.

*Волгоградский Государственный Технический Университет,
г.Волгоград, Россия*

konstantinovr1@gmail.com

vladimir.rozaliiev@gmail.com

vladimir.rozaliiev@gmail.com

В статье описывается система, предназначенная для создания из машинописного написания рукописного аналога. Сферы применения такой системы очень разнообразны. Целью разработки является создание методов и средств распознавания рукописного текста и его синтез для любого подчерка.

Ключевые слова: рукописный текст; автоматизированная система.

ВВЕДЕНИЕ

Существует достаточно ограниченное число программ для синтеза рукописного текста пользователя. Большинство таких средств требует специальных средств для своей работы. Однако направление синтеза подчерка очень перспективно, а сферы применения не ограничиваются только полиграфией и могут быть применены там где требуется рукописное заполнение (документы, анкеты, дневники и т.д.) Проектируемая система может использоваться в области образования для рукописного написания рефератов, лекций и т.п. Пользователями системы могут являться студенты, магистры, аспиранты, школьники.

Второй возможной областью применения данной системы является полиграфия: для создания плакатов, визиток, объявлений, приглашений, открыток с персональным рукописным шрифтом. Пользователями такой системы могут быть дизайнеры.

Объектом проектирования является процесс перевода текста из печатного в рукописный вид.

Основными целями разрабатываемой системы служат: создание персональных рукописных шрифтов; упрощение и минимизация затрат времени на создание рукописных документов.

1. Рассмотрение аналогов

Рассмотрим существующие аналоги разрабатываемой системы. В аналогичных системах [Алексеев и др., 2012] можно выделить следующие критерии сравнения:

- Цена системы
- Функциональные возможности
- Область применения и цели создания ПП
- Особенности способов реализации систем
- Требования к ПО рабочей станции
- Прочие особенности

Рассмотрим программу DMTHandWriter по выделенным критериям.

Цена лицензионной версии DMTHandWriter составляет 1.3 долларов.

Возможности:

- Преобразование из печатного текста любого текстового формата в изображение формата .gif
- Программа может использовать почерки из собственной базы или почерк пользователя
- Возможность задания различных вариантов написания символов
- Эмуляция рукописного текста шариковой ручкой, карандашом или фломастером
- Применение к выходному тексту различных эффектов

- Возможность изменения цвета пасты, обвода и т.п.

DMTHandWriter может применяться широким классом пользователей ввиду своей невысокой стоимости и простоты использования.

Имеет удобный и практичный интерфейс пользователя. Также в набор лицензии включена программа для сканирования почерка пользователя.

Для нормального функционирования необходим установленный пакет Macromedia Flash Player 6.

Исходный текстовый формат преобразуется в изображение с рукописным текстом за среднее время в 11 секунд.

Сравнивая DMTHandWriter с проектируемой системой, планируется создать систему, которая не уступала бы в функциональных возможностях. Также планируется попытаться разработать более реалистичную модель соединения рукописных букв.

Также в области разработки существует другой класс систем - системы, которые генерируют персональный шрифт пользователя.

Такие системы различаются в основном способами ввода образов рукописных букв и соединений между ними, т.е. ввод через специальные шаблоны, планшетные сканеры и с помощью электронного пера.

Полученное таким образом изображение эти системы преобразуют в векторный вид, разбивают его на элементы и создают шрифт, с графическими особенностями пользователя.

Весь класс описанных выше систем не решает главной проблемы в этой области – реалистичного слитного написания букв, т.е. соединения между буквами остаются не правдоподобными. Однако их рассмотрение полезно для понимания методов и алгоритмов создания персональных шрифтов.

В проектируемой системе планируется повторить достижения аналогичных систем.

2. Краткая постановка задачи

Разрабатываемая система должна относиться к классу программ-конвертеров. Такие системы обычно состоят из нескольких независимых модулей, каждый из которых имеет свою задачу и использует в качестве входных данных выходные данные предыдущего модуля. [Заболеева-Зотова и др., 2010]

Проектируемая система должна состоять из модуля-сканера печатного текста из текстового документа различных текстовых форматов, модуля-интерпретатора, который должен генерировать рукописный аналог печатного текста, модуля для применения различных эффектов к скенеризованному рукописному тексту.

Также в системе должна существовать база данных различных образцов подчерков.

В состав проектируемой системы также должна быть включена в качестве подсистемы система для создания персональных рукописных шрифтов.

В исследованном аналоге программе DMTHandWriter процесс генерации рукописного текста выглядит следующим образом: пользователь в интерфейсе системы вводит печатный текст или указывает системе документ с печатным текстом. После чего система, используя встроенную базу рукописных шрифтов преобразует текст в рукописный вид. При создании системы должен быть использован аналогичный подход.

Проектируемая система должна состоять из нескольких блоков:

1) Блок-сканер должен сканировать текстовый документ и разбивать его на слова и отдельные символы. Входные данные: текстовый документ различных текстовых форматов. Выходные данные: таблица слов, символов, знаков препинания.

2) Блок-интерпретатор должен преобразовывать полученный набор слов и символов в печатный аналог. Входные данные: таблица слов, символов, знаков препинания, встроенная база данных подчерков. Выходные данные: графическое представление рукописного текста без соединений.

3) Модуль для применения к полученному рукописному тексту различных эффектов. Входные данные: графическое представление рукописного текста без соединений. Выходное: изображение формата .gif с рукописным текстом.

4) База образцов почерков.

5) Модуль для учёта почерка пользователя. Входные данные, полученные с графического планшета или сканера. Выходные: контейнер соответствий печатных символов изображениям рукописных букв со сканера или массиву координат точек, полученных с графического планшета.

3. Описание решения

Для нормального функционирования системы в штатном режиме необходимо следующее техническое обеспечение: рабочая станция (или планшетный ПК) с процессором Intel 1.5 ГГц с поддержкой инструкций SSE3 (с процессором AMD 1ГГц с поддержкой инструкций SSE3), ОЗУ объемом не менее 3072 Мб, дисковая система объемом не менее 30 Гб, клавиатура, мышь. (Разрешение экрана рабочей станции или планшетного ПК не менее 1280 на 1024 точек).

Входом системы являются:

- Текст, введенный пользователем в область ввода на главной форме программы, или файл (*.txt), выбранный пользователем в диалоге для выбора файла в меню главного окна.

- Настройки графических эффектов, а также настройки, установленные при создании нового проекта.

- Файлы с образцами шрифтов пользователя (*.sim).

Файл *.sim со шрифтом хранит информацию о рукописном представлении 255 символов из таблицы ASCII (цифры, символы, латинский алфавит и русский алфавит). Информация записывается последовательно, начиная с символа с кодом 0. Для каждого символа в файл пишется следующие данные: по 4 байта для координат точек соединения символов, по 2 байта на ширину и высоту битового изображения символа, (4 * N) байт для сохранения пар чисел i, j – координат точек принадлежащих символу.

Выходными данными являются:

- Изображение рукописного аналога входного печатного текста, отображаемое в области вывода в главном окне программы.

- Файл формата *.jpg, содержащий вышеуказанное изображение.

Система состоит из следующих подсистем:

- Подсистема создания и редактирования персональных шрифтов пользователей. Подсистема обеспечивает создание, редактирование, сохранение, удаление файлов с образцами рукописных букв пользователя.

- Подсистема интерпретации печатного текста в рукописный аналог. Подсистема обеспечивает разбор текста, введенного пользователем или разбор текстового файла формата *.txt, указанного пользователем и создание внутренней структуры соответствия печатных символов рукописным аналогам, а также создание файла формата jpg., содержащего выходной рукописный аналог печатного входного текста.

- Блок графических эффектов. Подсистема обеспечивает наложение специальных графических эффектов на выходной рукописный текст (функции задания размера рукописных букв, расстояния между буквами и словами, эффект прыгающих букв, эффект переменного расстояния между буквами, эффект сглаживания).

В процессе выполнения настоящей работы созданы сами подсистемы, а также разработаны методы их связи и обмена данными между ними.

Работы по созданию системы проведены с учетом следующих особенностей [Розалиев и др., 2010]:

- Создания максимально правдоподобного почерка человека.

- Создания максимально возможного количества эффектов для выходного рукописного текста.

- Учет возможности развития системы.

В состав математического обеспечения системы входят следующие положения:

- Для программной реализация эффекта «прыгающих букв» для расчёта высоты буквы (h) будет использована следующая модель $H=h+\delta(W)$, где H – новая высота буквы, h – исходная высота буквы, $\delta(W)$ – случайная величина, сгенерированная по равномерному закону в диапазоне $[-h/2;h/2]$.

- Для программной реализации эффекта увеличения расстояния между буквами будет использована модель $D=d+\text{const}(R)$, где D – новое расстояние между двумя соседними буквами, d – исходное расстояние, $\text{const}(R)$ – некая положительная константа.

- Для программной реализации эффекта сглаживания будет использован следующий метод. Запускается три цикла. Первый цикл обрабатывает пиксели по горизонтали битовой карты буквы, второй - по вертикали, выбирая каждый пиксель в строке пикселей по горизонтали, третий - сглаживает рядом стоящие пиксели по вертикали и горизонтали, начиная от пикселя, установленными первыми двумя циклами. Все цвета разлагаются на составляющие (красный, зеленый, синий), и каждая составляющая пикселей складывается с составляющей соседних пикселей после чего находится среднее значение цвета. В результате должно получиться немного «размытое» изображение.

Рассмотрим примеры экранных форм системы.

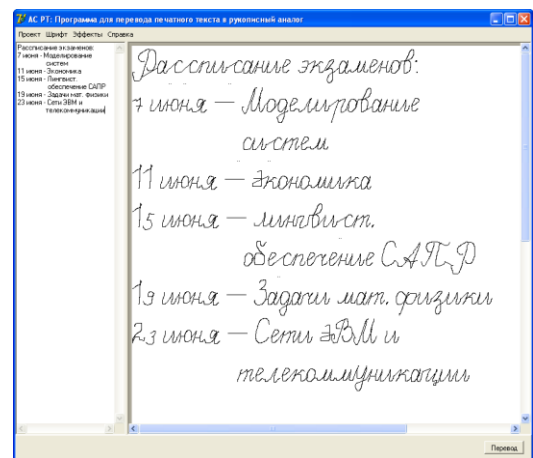


Рисунок 1 – Основное окно перевода текста

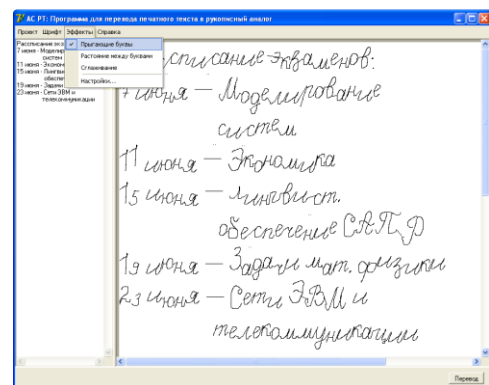


Рисунок 2 – Эффект «прыгающие буквы»

На рисунке 1 показан пример перевода печатного текста в рукописный аналог. Далее показаны примеры применения эффектов для рукописного текста.

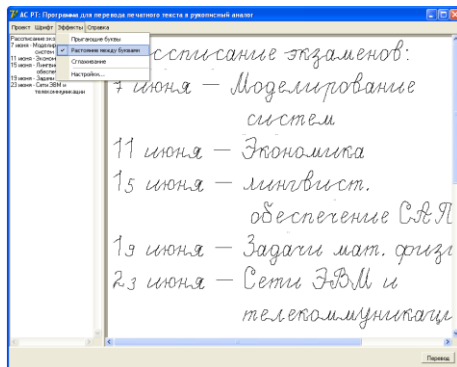


Рисунок 3 – Эффект «растояние между буквами»

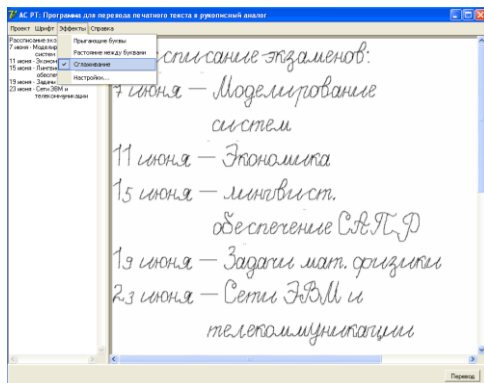


Рисунок 4 – Эффект «сглаживание»

Для редактирования шрифта пользователя создан специальный редактор, показан на рисунке 5.

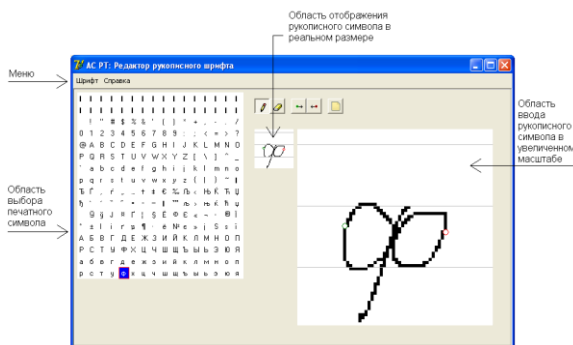


Рисунок 5 – Редактор шрифтов пользователя

ЗАКЛЮЧЕНИЕ

В качестве дальнейшего развития системы рассматриваются следующие направления: улучшение механизма соединения букв в рукописном тексте; увеличение базы данных рукописных шрифтов; эмуляция почерка левши.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 12-07-00266, 12-07-00270).

Библиографический список

[Алексеев и др., 2012] Алексеев, А.В. Автоматизация определения шрифтов по изображению / Алексеев А.В., Розалиев В.Л. // Труды конгресса по интеллектуальным системам и информационным технологиям "IS&IT'12" (Дивноморское, Краснодарский край, 2-9 сент. 2012 г.). В 4 т. Т. 1 : тр. конф. "Интеллектуальные системы '12" и "Интеллектуальные САПР – 2012" / ЮФУ [и др.]. - М., 2012. - С. 292-293.

[Розалиев и др., 2010] В.Л. Розалиев, А.С. Бобков, О.С. Федоров Применение нейронных сетей и грануляции при построении автоматизированной системы определения эмоциональных реакций человека / Изв. ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах»: межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2010. - Вып. 9, № 11. - С. 63-68.

[Заболеева-Зотова и др., 2010] Заболеева-Зотова А.В., Орлова Ю.А. Автоматизация начальных этапов проектирования программного обеспечения // Изв. ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах". Вып. 8 : межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2010. - № 6. - С. 121-124.

RECOGNITION OF SIGN RUSSIAN LANGUAGE GESTURES

Konstantinov V.M., Rozaliev V.L., Dianov I.A.

*Volgograd State Technical University,
Volgograd, Russia*

konstantinovr1@gmail.com

vladimir.rozaliev@gmail.com

This paper describes a system designed for the creation of a typewritten manuscript writing counterpart. Scope of application of such a system is very flexible. The purpose of development is to create methods and tools for handwriting recognition and synthesis for any underscores.

INTRODUCTION

The object of the design is the process of translating the text from the image in handwritten form. The main goals of the system under development are: the creation of personal handwritten fonts simplify and minimize the time required to create written documents.

MAIN PART

The existing system developed counterparts. Identified advantages and disadvantages. The task is to design the system. The realization and shows the screen.

CONCLUSION

In conclusion shows the ways of improving the system through the mechanisms of improvements and increasing database.

This work was partially supported by the Russian Foundation for Basic Research (projects 12-07-00266, 12-07-00270).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПРОТОТИП ПЛАТФОРМЫ АНАЛИЗА РЕЧИ НА ТАТАРСКОМ ЯЗЫКЕ

Хусаинов А.Ф., Сулейманов Д.Ш.

*НИИ «Прикладная семиотика» Академии наук Республики Татарстан,
Казанский (Приволжский) федеральный университет,
г.Казань, Россия*

khusainov.aidar@gmail.com

dvdt.slt@gmail.com

В работе приводится подход к созданию платформы анализа речи на татарском языке. В контексте данной платформы описываются основные модули, такие как модуль распознавания речевых команд, идентификации языка и распознавания фонем. Основная идея работы заключается в исследовании потенциала улучшения качества работы отдельных систем анализа речи за счет применения кросс-модульного взаимодействия. Предлагаемая модель платформы может быть эффективно использована при создании комплексных систем анализа речи.

Ключевые слова: платформа анализа речи, кросс-модульное взаимодействие систем, татарский язык

ВВЕДЕНИЕ

Стремительное развитие современных электронных устройств привело к осознанию ограниченности существующих инструментов человеко-машинного взаимодействия, таких как клавиатура и мышь. Одним из путей преодоления этого ограничения является использование речи при взаимодействии с компьютером и другими устройствами. Данный подход включает в себя работу по решению множества задач, относящихся к таким направлениям анализа речи, как распознавание, синтез речи, идентификация диктора, языка диктора и т.д. Однако большинство исследований ученых сильно специализировано и направлено на разработки отдельных частей речевых технологий. Данный факт может быть объяснен как экономическими причинами, так и технологической сложностью стоящих задач. С одной стороны, такие исследования позволяют глубже изучать предметную область, а с другой, в таком случае зачастую уделяется недостаточно внимания вопросам разработки комплексных архитектурных решений, способных решать весь комплекс задач речевых технологий.

Основываясь на указанных рассуждениях, было решено исследовать возможную архитектуру комплексной системы анализа речи и её использование в контексте татарского языка. Создание такой архитектуры позволит, во-первых, увеличить эффективность работы всех модулей

системы за счет синергетического эффекта, а во-вторых, предоставит готовый инструмент разработки речевых технологий для языков, для которых на данный момент не создано качественных систем анализа речи.

Структура данной работы следующая: в разделе 1 работы обсуждаются основные вопросы, касающиеся структуры модулей распознавания речи, идентификации диктора и языка, а также отмечены основные принципы построения комплексной системы анализа речи. В разделе 2 описывается возможность применения предложенной архитектуры системы к анализу татарского языка.

1. Архитектура системы

1.1. Общие положения

Область речевых технологий включает в себя множество направлений, основные из которых могут быть определены следующим образом:

- распознавание, синтез речи;
- идентификация языка;
- идентификация диктора;
- диаризация;
- определение характеристик диктора;
- идентификация характеристик канала связи.

Синергетический эффект, которого можно достичь при совместной работе различных модулей

анализа речи, можно представить состоящим из 3 уровней:

1. использование общих компонент (вычисление признаков речи, математические преобразования);
2. использование в качестве общих блоков целых модулей анализа речи, таких как модуль определения характеристик шума, распознавания фонем;
3. обмен информацией между различными модулями.

Основываясь на приведенной классификации, отметим, что использование первых двух уровней синергетического эффекта может позволить ускорить процесс разработки программных средств за счет использования общих сущностей в различных ситуациях. В то же время 3 уровень даёт возможность улучшить качество работы отдельных модулей предлагаемой платформы.

Рассмотрим данную точку зрения на примере системы анализа речи. Для простоты будем учитывать только три составных модуля системы: распознавание речи, идентификация языка и диктора. Приведём описание структур модулей в подразделах 1.2-1.4 и описание архитектуры системы в подразделе 1.5.

1.2. Распознавание речи

В качестве структуры системы распознавания речи будем использовать общераспространённую схему работы подобных систем, представленную на рисунке 1. Поступающий на вход системы речевой фрагмент подаётся в блок выделения значимых характеристик речи «Feature extraction», вычисленные на первом этапе характеристики используются декодером для получения результата распознавания.

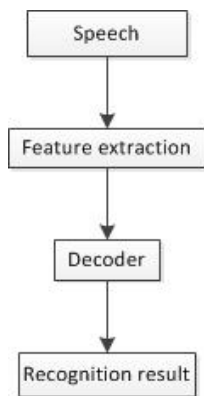


Рисунок 1 – Структура модуля распознавания речи

Для проведения дальнейшего анализа добавим в данную схему основные зависимости системы. Как видно из представленной на рисунке 2 измененной схемы, основными зависимостями блока «Feature extraction» являются подсистема предобработки, включающие в себя различные фильтры, а также подсистема вычисления векторов признаков речи. Декодер использует в своей работе такие подсистемы как модуль обнаружения речевой

активности, акустические и языковые особенности.

Здесь и далее используются следующие аббревиатуры:

- MFCC (mel-frequency cepstral coefficients);
- LPC (linear predictive coding);
- PLP (perceptual linear predictive);
- ANN (artificial neural networks);
- HMM (hidden Markov models);
- DTW (dynamic time warping).

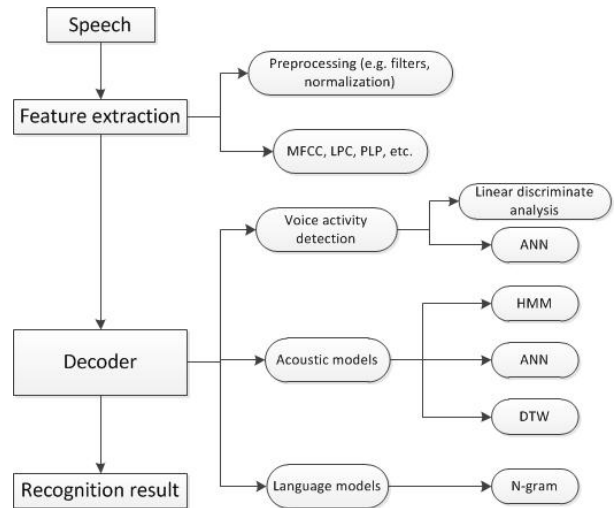


Рисунок 2 –Модуль распознавания речи с зависимостями

1.3. Идентификация языка

Исследования в области идентификации языка насчитывают более 30 лет, за которые учеными было разработано множество подходов к решению данной задачи. Однако, для простоты, остановимся лишь на трёх основных подходах: PRLM (phone recognition followed by language modeling), parallel PRLM и PPR (parallel phone recognition).

Как следует из названия подхода, метод PRLM состоит из двух основных этапов: распознавание фонем и моделирования языков (как правило, с помощью моделей n-gram). Структура работы метода PRLM представлена на рисунке 3.

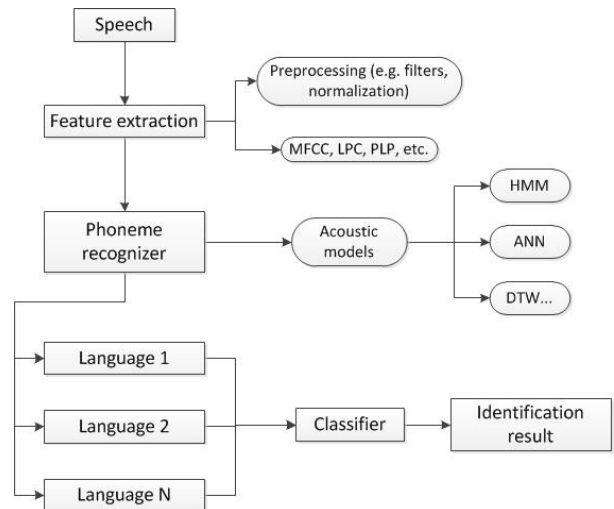


Рисунок 3 – PRLM-подход идентификации языка

При таком подходе сообщения из обучающей части речевой базы подаются на вход системы распознавания фонем, построенной для одного языка (блок «Phoneme recognizer»). На основе полученных последовательностей фонем строятся модели языков, которые в дальнейшем используются в качестве языковых моделей (блоки «Language 1», «Language N»). В процессе распознавания речевой фрагмент также разбивается на последовательность фонем, после чего вычисляются вероятности принадлежности данной последовательности каждой из языковых моделей. С помощью блока классификации «Classifier» определяются вероятности принадлежности произнесённой фразы каждой из доступных языковых моделей. Языковая модель с максимальной вероятностью выбирается в качестве результата идентификации.

Главным и единственным отличием описанного PLRM от parallel PRLM-подхода является использование одновременно нескольких систем распознавания фонем, настроенных на разные языки. Основная идея данной модификации состоит в том, чтобы повысить вероятность того, что максимальное количество произнесенных звуков было корректно распознано на основе языков, использованных для построения систем распознавания фонем.

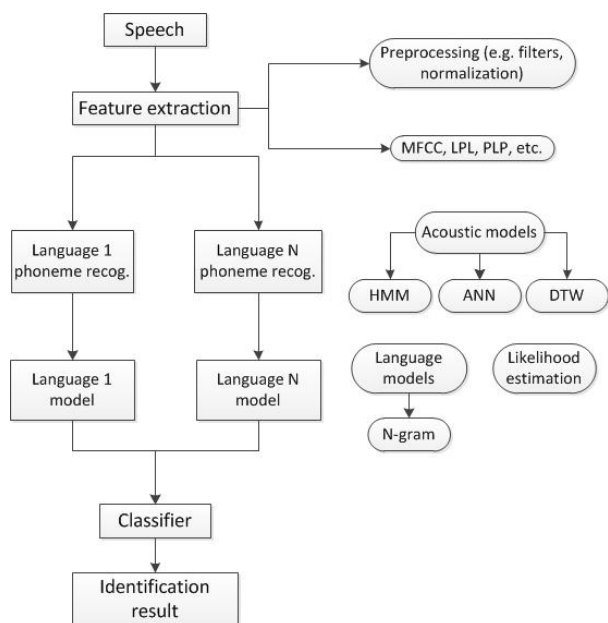


Рисунок 4 – PPR-подход идентификации языка

Если предположить наличие фонетически размеченных речевых корпусов для всех языков, для которых требуется производить идентификацию, то становится возможным использование PPR-подхода. Схема работы данного подхода с учётом имеющихся зависимостей изображена на рисунке 4. Основной особенностью этого метода является использование информации о закономерностях следования фонем в каждом языке не после этапа распознавания, а во время него. Концептуально это позволит отфильтровать не встречающиеся в языке

последовательности фонем, что позволит без искажений рассчитывать результаты идентификации. В рамках приведённой схемы работы алгоритма это выражается в использовании элементов «Language models»-«n-grams» на этапе работы блоков «Language phoneme recognition».

1.4. Идентификация диктора

Основная цель данного модуля - определить человека, произносящего фразу, на основе некоторых значимых особенностей речи. Условно задачу можно разбить на два подкласса: текстозависимую (с наличием ключевой фразы) и текстонезависимую. Структура модуля идентификации диктора представлена на рисунке 5.

Здесь и далее используются следующие аббревиатуры: GMM (Gaussian mixture models), VQ (vector quantization).

На этапе обучения моделей «Speaker model 1», «Speaker model N» используются, в зависимости от наличия ключевой фразы, такие методы, как скрытые Марковские модели (HMM), методы динамического программирования (например, DTW), гауссовские смеси (GMM) и т.д. Полученные модели дикторов используются на этапе распознавания для расчета вероятностей произнесения фразы каждым конкретным диктором, после чего блок «Classifier» определяет модель, которой соответствует максимальная вероятность. и которая Диктор, соответствующий данной модели, и будет являться результатом идентификации.

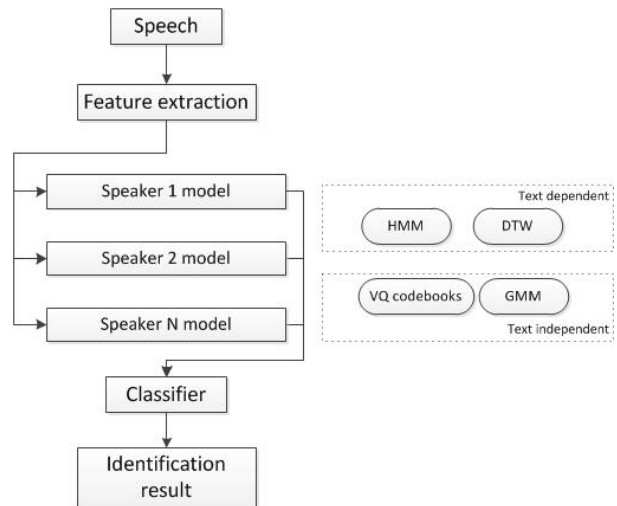


Рисунок 5 – Структура модуля идентификации диктора

1.5. Комплексная система анализа речи

Основываясь на предложенной идее о 3 уровнях синергетического эффекта и описанных схемах работы модулей анализа речи, перейдем к построению архитектуры комплексной системы анализа речи. Основными характеристиками разрабатываемой архитектуры должны стать:

- модульный дизайн – возможность независимой разработки различных модулей;

- гибкость – архитектура должна позволять использовать и сравнивать различные варианты реализации отдельных модулей;
- расширяемость – возможность легкого добавления новых модулей, расширяющих функционал системы;
- настраиваемость системы на разные языки – каркас системы должен состоять из языко-независимых блоков, однако добавление специфических для конкретного языка элементов должно увеличивать качество работы комплекса.

1.5.1. 1 уровень синергетического эффекта

Как видно из структур модулей, представленных на рисунках 2-5, большинство методов используется больше чем в одном модуле (рисунок 6). Следовательно, выделение данных элементов в отдельный, вспомогательный, модуль («Assistant tools») может позволить ускорить процесс разработки системы.

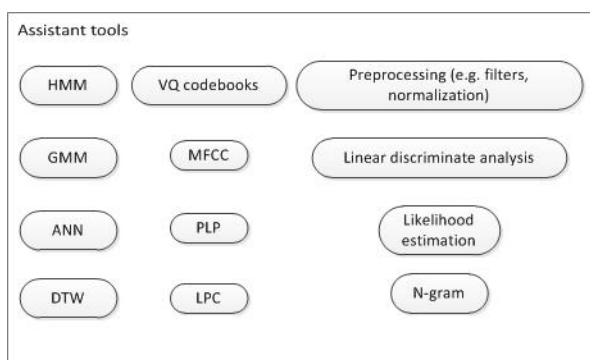


Рисунок 6 – 1 уровень синергетического эффекта

1.5.2. 2 уровень синергетического эффекта

Как было отмечено выше, подход использования общих компонент может быть реализован не только на уровне реализации отдельных инструментов, но и целых модулей, например, модуля распознавания фонем. Стоит заметить, что в случае речевых технологий речь может идти о совместном использовании, в том числе и «неречевых» блоков: модуля анализа движения губ, дополнительной фидбэка шумовых компонент.

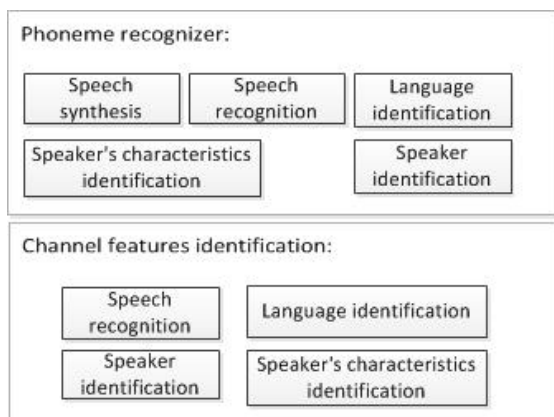


Рисунок 7 – 2 уровень синергетического эффекта в речевых технологиях

Как видно из примеров использования 2 уровня синергетического эффекта, приведенных на рисунке

7, система распознавания фонем «Phoneme recognizer» может быть использована в таких модулях, как синтез речи, распознавания речи, идентификация языка, диктора, а система определения характеристик канала связи – в модулях распознавания речи, определения языка, диктора и характеристик говорящего.

1.5.3. 3 уровень синергетического эффекта

В то время как за счет первых двух уровней синергетического эффекта можно достичь создания системы с лучшим дизайном архитектуры, систему, которую было бы легко расширять и дорабатывать, напрямую на качество работы систем анализа речи они повлиять не в состоянии. Улучшение качества работы может быть достигнуто за счет использования взаимодействий между различными модулями. Возможные варианты взаимодействия и преимущества, к которым может привести их использование, представлены в виде таблицы 1.

Таблица 1 – Примеры взаимодействий между различными модулями анализа речи

Модуль	Возможные варианты использования	Комментарии
Идентификация языка	Распознавание фонем, речи, идентификация диктора, синтез речи	Знание языка, на котором произнесена фраза, позволяет использовать специфичные для данного языка алгоритмы и информацию (например, при решении задачи идентификации диктора может быть использован список дикторов, владеющих данным языком)
Идентификация диктора	Распознавание фонем, речи, идентификация языка, синтез речи	Наличие информации о дикторе может позволить использовать предопределенные для данного диктора акустические и языковые модели
Идентификация пола, возраста диктора	Распознавание фонем, речи, синтез речи	Информация о поле диктора, его возрасте и эмоциональном состоянии может быть использована для корректировки фильтров и других

Модуль	Возможные варианты использования	Комментарии
		параметров настройки модулей системы
Идентификация темы разговора	Распознавание фонем, речи	Тема разговора может быть использована в качестве дополнительного ограничения, накладываемого на используемый при распознавании лексикон

1.5.4. Аспекты программной реализации

Основываясь на свойствах платформы, определенных в п.1.5, можно сделать следующие предположения:

- исходная информация, инструменты и базовые алгоритмы должны быть сгруппированы в отдельные модули по их функциональности;
- базовые структуры данных, такие как вектор MFCC, LPC и другие, должны быть предопределены в системе и использоваться едино всеми модулями;
- модули, такие как модуль распознавания фонем, должны быть предопределены, и их функциональность должна быть описана в стандартных для данной платформы интерфейсах;
- должны быть заданы базовые взаимодействия между модулями;
- платформа должна позволять программисту изменять существующие функции, добавлять новые модули, расширять существующие интерфейсы и переопределять взаимодействия между различными модулями.

Предполагается добиться перечисленных характеристик платформы за счет использования следующих сущностей:

- solution (решение) – сущность, содержащая информацию о том, каким образом платформа должна решать конкретную задачу. Данная сущность задается с помощью вспомогательных сущностей solution flow (см. ниже) и необходимых параметров;
- solution flow (цепочка) – сущность, инкапсулирующая отдельные части алгоритма; задаётся набором действий (activity) и правилами перехода между этими действиями;
- solution flow activity (активити) - сущность, представляющая собой реализацию конкретной функции. В данном контексте, справедливо, что каждая функция каждого модуля реализована в виде отдельного элемента активити. Для сущности solution flow данное действие представляется в качестве черного ящика с известными входными

параметрами и ожидаемым результатом работы. Сущности solution flow activity могут быть соединены друг с другом, представляя совместно алгоритм решения задачи. При этом каждое activity может иметь несколько исходящих связей, использующееся соединение определяется на основе результатов работы функции, заложенной в активити;

- task (task) – сущность, хранящая всю текущую информацию в процессе выполнения платформой конкретного задания. Task позволяет различным элементам решения получать актуальную информацию о работе другого элемента. Фактически, сущность task представляет собой временный объект, строящийся на основе схемы, заданной соответствующим ему решением.

Пример реализации данного подхода на примере работы модуля идентификации языка (PRLM-подход) представлен на рисунке 8.

Предположим, что у нас есть модули под названием «Corpora», «Phone recognizer» и «Language model», и что их интерфейсы предоставляют все необходимые функции. При данных предположениях, мы можем описать решение задачи идентификации языка с помощью решения (в соответствии с введенной терминологией), состоящего из трех цепочек, по одной для каждого из описанных выше модулей. Каждая цепочка состоит из последовательности элементов-активити, реализующих ту или иную функциональность модуля. Элемент task в данном случае ответственен за решение задачи согласно описанному алгоритму (создание потоков для каждой из цепочек, вычисление параметров, переход между элементами-активити).

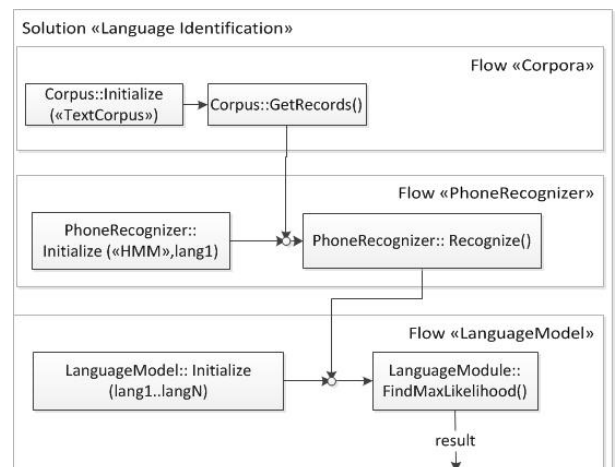


Рисунок 8 – 1 уровень синергетического эффекта в речевых технологиях

Такой тип архитектуры приложения несёт в себе следующие преимущества:

- предоставление инструмента создания сценариев анализа языка, состоящих из простых блоков;
- предоставление простого задаче-ориентированного способа решения конкретных задач. Например, лингвист может изменить

параметры элемента-активности для модуля Corroga и посмотреть, каким образом изменились результаты

распознавания, при этом от него не требуется быть экспертом во всех задействованных областях

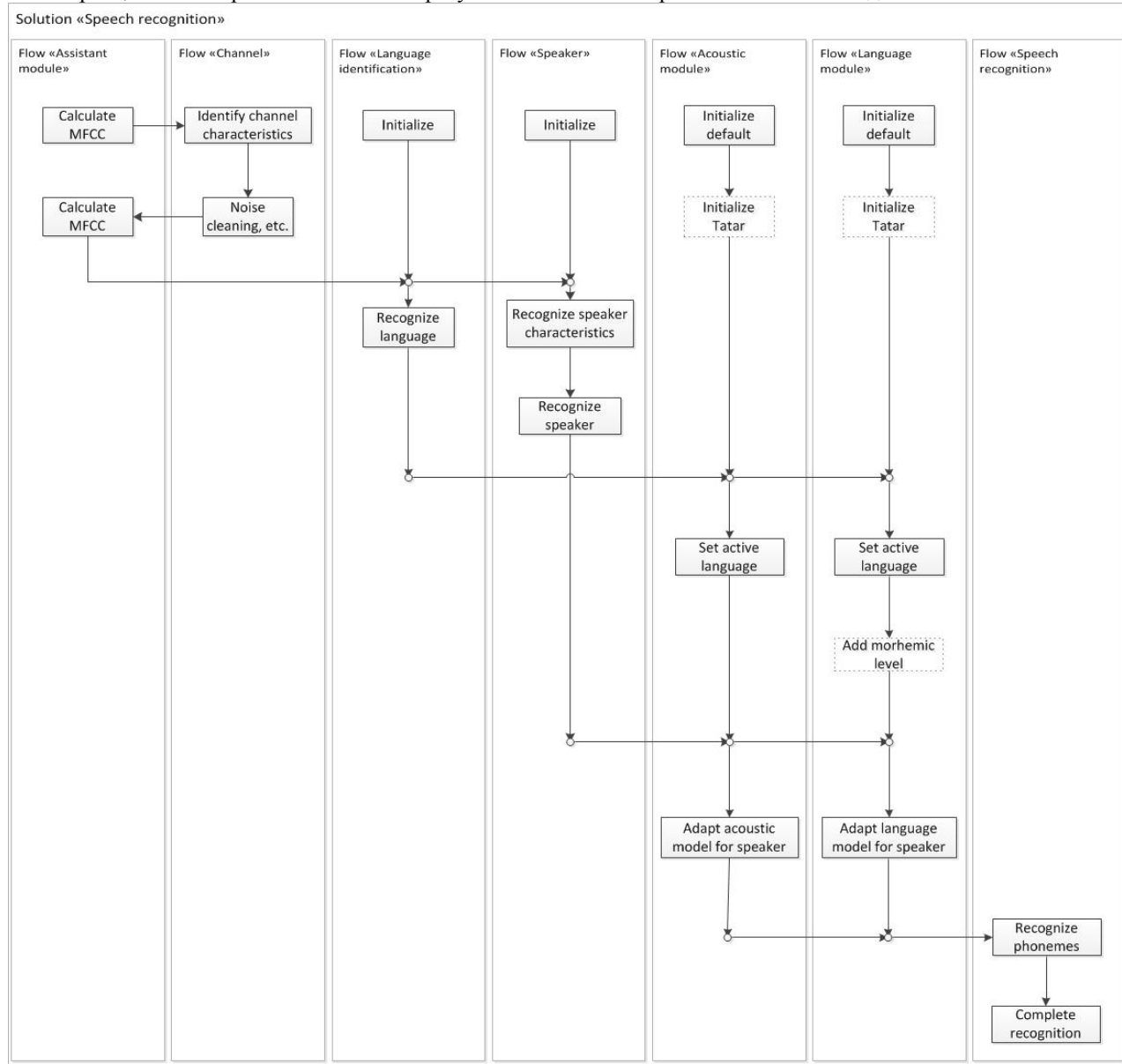


Рисунок 9 – Схема работы системы распознавания речи

анализа речи;

- «параллельная» структура исполнения решений предоставляет возможность уменьшения времени обработки задач.

Кроме того, одним из важнейших преимуществ данной платформы является то, что при таком подходе становится возможным учитывать текущий уровень развития речевых технологий для различных языков. В то время как существует множество ресурсов и алгоритмов, разработанных для английского, французского, испанского языков, существует достаточно большой класс языков с низким уровнем разработок в данной области.

Разработка предлагаемой архитектуры с предопределенными стандартными методами, алгоритмами и речевыми корпусами можно упростить и ускорить процесс создания систем анализа речи для многих языков: разработки

подобных систем смогут сконцентрировать свои усилия на значимых особенностях конкретного языка, используя при этом заранее зарекомендовавшие себя алгоритмы. Возможность использования данной платформы для разработки системы анализа речи на татарском языке будет рассмотрена в следующем разделе.

2. Платформа анализа речи для татарского языка

Существует несколько основных причин заниматься разработкой речевых технологий для татарского языка. Во-первых, существует социокультурный аспект: язык выступает как инструмент отражения быта и сохранения исторического наследия народа, говорящего на нём. И единственным способом сохранения накопленной информации является помощь языку в

адаптации к современным особенностям жизни, которые включают в себе широкое использование информационных технологий. Во-вторых, создание платформы для анализа татарского языка позволит создавать практические приложения для людей, говорящих на татарском языке, например:

- распознавание и синтез речи – для использования людей с нарушениями зрения;
- распознавание речи – использование для поиска информации в сети Интернет, написание электронных писем, смс, анализа аудио-архивов на татарском языке, для навигации по сайтам, в системах голосового самообслуживания;
- верификация диктора – предоставление безопасного доступа к различной информации;
- распознавание чисел – использование в IVR-меню (interactive voice response menu) в телефонии.

Для демонстрации возможностей использования предложенной архитектуры при проектировании систем анализа речи, приведём схему решения с её помощью задачи распознавания речи (рисунок 9). Кроме того, данный пример демонстрирует возможность адаптации языконезависимого решения к татарскому языку за счет добавления в него специфических для татарского языка элементов. Для наглядности такие элементы выделены в схеме пунктирной линией.

Как видно из рисунка 9, решение «Speech recognition» состоит из 7 отдельных потоков, по одному для каждого из следующих модулей:

- Assistant module – вспомогательный модуль, содержащий инструменты для вычисления стандартных параметров речи (реализующий идею о 1м уровне синергетического эффекта);
- Channel – модуль, определяющий характеристики канала связи и производящий на основе полученных данных фильтрацию возможных искажений исходного сигнала;
- Language identification – модуль идентификации языка диктора;
- Speaker – модуль идентификации диктора;
- Acoustic module – модуль, занимающийся настройкой акустических моделей, используемых при распознавании;
- Language module – модуль, занимающийся настройкой языковых моделей, используемых при распознавании;
- Speech recognition – модуль распознавания речи.

Каждый поток решения состоит из набора элементов-активити, выполняющих определенную процедуру и взаимодействующих согласно установленным связям. Прямые переходы между активити изображены на схеме с помощью стрелок (например, переход между элементом «Calculate MFCC» и элементом «Identify channel characteristics» другого потока). Однако существует ситуации, в которых для выполнения действия

необходимо выполнение одновременно нескольких подготовительных процедур. Так, например, до начала процесса распознавания фонем (поток «Speech recognition», активити «Recognize phonemes»), необходимо корректным образом настроить текущие языковую и акустическую модели (активити «Adapt acoustic/language model for speaker»). Для этих целей архитектурой системы предусмотрена возможность синхронизации различных потоков; точка синхронизации в предложенной на рисунке 9 схеме изображается с помощью окружности. Как видно из схемы, результаты работы каждого модуля (потока) используются для корректировки работы других модулей. Так, например, информация о дикторе используется в акустическом и языковом модуле.

ЗАКЛЮЧЕНИЕ

В работе представлен подход к созданию платформы анализа речи и её применения для татарского языка. Были определены три уровня синергетического эффекта, которые могут быть использованы в рамках создания систем анализа речи, и рассмотрены основные преимущества предлагаемой архитектуры.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Huang, 2009] Huang, X., An overview of modern speech recognition / X. Haung, L. Deng // Handbook of Natural Processing, C. 339-366
- [Hitrov, 2011] Hitrov, M. Synergetic effect in speech technologies / M. Hitrov // In The 14th International Conference «Speech and Computer», C. 27-32
- [Jurafsky, 2009] Jurafsky, D. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition / D. Jurafsky, H. James // Prentice Hall, 2009. – 1024 с.
- [Zissman, 1996] Zissman, A. Comparison of Four Approaches to Automatic Language Identification of telephone Speech / A. Zissman // IEEE transactions on speech and audio processing. 4(1)
- [Rabiner, 1993] Rabiner, L. Fundamentals of Speech Recognition / L. Rabiner, B.H.Juang // Prentice Hall Signal Processing Series
- [Niesler, 2006] Niesler, T. Language identification and multilingual speech recognition using discriminatively trained acoustic models / T. Niesler, D. Willett // In ISCA Workshop on Multilingual Speech and Language Processing (MULTILING 2006), Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa

SPEECH ANALYSIS PLATFORM PROTOTYPE FOR TATAR LANGUAGE

Khusainov A.F., Suleymanov D.S.

«Applied semiotics» Institute, Tatarstan Academy of Sciences,

Kazan (Volga region) federal university, Russia, Kazan

khusainov.aidar@gmail.com

dvd.t.slt@gmail.com

In this paper, we present a complex speech analysis platform for Tatar language. We explore main modules of this system including speech command recognition,

language identification and phoneme recognition modules. The major idea was to investigate potential of cross-module interactions to increase effectiveness of each module work.

INTRODUCTION

Nowadays the rapid development of all types of electronic devices led to a problem of limited speed and quality of existing instruments of computer-human communication (keyboard, etc.). One way of solving this problem is to use speech to communicate with computers. This way includes developing speech recognition, synthesis, speaker, language identification systems. But the greater part of scientists' research activities are focused on separate areas of speech technologies. This fact could be explained with economic reasons and complexity of tasks. Such kind of approach makes easier to go deeper in theoretical and practical aspects of specific area of speech technologies. In other hand, it gives too little attention to explore and develop ways of building complex architecture which could provide functionality of all tasks of speech analysis.

Based on the above observation we have focused our research on studying the possible structure of complex platform for speech analysis and its application to Tatar language. Creating such kind of architecture will make possible, first of all, to increase effectiveness of speech analysis systems via synergetic effect and, secondly, will provide tool for developing speech technologies for languages which don't yet have well-designed speech analysis systems.

MAIN PART

Speech analysis technologies include various areas of tasks; for today following main areas could be distinguish as:

- speech recognition;
- speech synthesis;
- language identification;
- speaker identification;
- speaker verification;
- diarisation;
- speaker's characteristics identification (sex, emotional status, prosodic features);
- signal channel's characteristics identification (channel type, existence and nature of noise, etc.).

Synergetic effect in speech analysis systems could be described consisting of three levels:

- lower level – using common tools;.
- middle level – using whole modules (like phoneme recognition) as common blocks;
- top level – exchanging information between different modules.

According to this classification we can see that lower and middle levels give researchers possibility to accelerate algorithms developing speed and increase simplicity of that process by using same entities in

different situations. At the same time top level gives us opportunity to increase effectiveness of separate modules' work.

Based on idea of three-level synergetic effect and speech analysis modules' structure, we now can enumerate main features of proposed system:

- modular design – possibility to separate developing of different modules;
- flexibility - system must provide possibility of using and comparing different algorithms;
- expandability – easy to expand system by adding new modules or new type of realizations;
- language-adjustable – skeleton of the system consists of language-independent blocks, but adding language-specific items and algorithm realizations could increase speed and quality of the system.

Found on given features of platform we suggest that it can be realized by using following internal entities:

- solution – entity which contains information about how concrete task will be solved in system. It consists of algorithm parts represented by solution flows (see below) and required parameters;
- solution flow – entity which encapsulates separate part of algorithm; it's represented by actions, called activities (see below), and transition rules;
- solution flow activity – entity which contains realization of specific function. It can be assumed that all functions of all speech analysis modules are wrapped into this entity;
- task – consolidates all information during execution program.

One of the biggest advantages of this approach is the fact that proposed platform takes into account current development level of speech technologies for different languages. While there are various corpora and algorithms for such languages as English, French, etc., there is still a wide range of less developed languages. And creating proposed platform with pre-defined standard methods, algorithms and existing corpora can simplify process of developing comprehensive speech technologies for another language; developers can concentrate on distinctive features of this language and use already well-designed tools.

CONCLUSION

We have presented a speech platform and its application for Tatar language. We have also discussed possible benefits of creating such kind of complex system and demonstrated three levels of synergetic effect that can be used. We conclude that proposed programming model is well suited to speech technologies due to their computational complexity, etc.

In the future we intend to develop proposed speech platform for Tatar language, implement existing language tools (e.g. Tatar speech synthesis, command recognition system); and to create application-oriented systems based to this platform.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИЧЕСКАЯ СЕТЬ ПРЕДМЕТНОЙ ОБЛАСТИ КАК ОСНОВА ДЛЯ ФОРМИРОВАНИЯ СЕТИ ПЕРЕХОДОВ ПРИ АВТОМАТИЧЕСКОМ РАСПОЗНАВАНИИ СЛИТНОЙ РЕЧИ

Харламов А.А. *, Ермоленко Т.В.**

* *Институт высшей нервной деятельности и нейрофизиологии РАН, г. Москва*

kharlamov@analyst.ru

** *Институт проблем искусственного интеллекта, г. Донецк, Украина*

etv@iai.dn.ua

Задача распознавания слитной речи в настоящий момент не считается успешно решенной. Тем не менее, известны лицензионные продукты таких фирм как Nuance, Autonomy, которые используются в приложениях, в том числе, русскоязычных. Однако сеть допустимых переходов между словами в заданной предметной области строится в этих системах вручную, а это трудоемкий процесс. В статье описывается подход к автоматическому формированию сети переходов для заданной предметной области на основе использования ассоциативной сети корпуса текстов этой предметной области.

Ключевые слова: распознавание ключевых слов в потоке слитной речи, сеть переходов, автоматическое формирование ассоциативной сети текста.

ВВЕДЕНИЕ

Базовая модель распознавания слитной речи может быть реализована на основе разных подходов. Одним из наиболее известных и хорошо зарекомендовавших себя является подход на основе скрытых Марковских моделей, имеющий общие корни с алгоритмом динамического программирования, разработанным Т.К. Винцоком [Винцюк, 1987]. При этом контекст различных уровней учитывается сетью переходов: фонемы соединяются в слова, слова – в предложения, предложения принадлежат текстам, описывающим предметную область. Предполагается, что на вход системы поступает параметризованный речевой сигнал в виде последовательности векторов параметров в пространстве первичного описания. Параметризация сигнала производится препроцессором, который моделирует физическое восприятие речевого сигнала с учетом свойств акустического канала и фоновых помех. В ответ на произнесенное речевое сообщение на выходе системы получается результат распознавания в виде одной или нескольких гипотез о произнесенной последовательности слов. Эти последовательности, дополненные оценкой сопутствующих речевых параметров, передаются в блок смысловой интерпретации, где происходит принятие

окончательного решения в пользу одной из гипотез. Знания о предметной области, в которой используется система распознавания, позволяют существенно сузить и ускорить поиск.

В одном из подходов к распознаванию слитной речи при преобразовании речи в текст используется словарь, в котором каждое слово задано своим графемным представлением. По этому представлению слова формируются одна или несколькими фонемными транскрипциями слова. Далее, исходя из фонемного состава слова, формируется акустический прототип слова путем объединения различных прототипов акустических образов фонемного уровня (например, трифонов). При распознавании предъявленная реализация слова сравнивается с преобразованными сформированными акустическими прототипами слова. При преобразованиях прототипов слова сохраняется порядок следования акустических форм фонемного уровня прототипа и варьируется в разрешенных пределах длительность акустических форм. Процесс сравнения и поиск наилучшей меры сходства осуществляется методом динамического программирования. Распознаваемая реализация относится к тому слову, преобразованный прототип которого дал наибольшее интегральное сходство с распознаваемым сигналом.

Также как прототипы слов из прототипов графем, создаются прототипы словосочетаний и предложений из прототипов слов и рассматриваются их допустимые преобразования. И в этом случае процесс перебора допустимых фраз и возможных границ между словами достигается методами динамического программирования. Ответом на слитную речевую последовательность является та допустимая фраза или предложение, преобразованный прототип которой оказался самым похожим на предъявленный для распознавания речевой сигнал. Наиболее просто, но не с наилучшим качеством, решается задача распознавание слитной речи в случае свободного порядка следования слов. Если же учитывать не только лексику, но и синтаксис и семантику речи, то на порядок слов накладываются дополнительные тематические ограничения (ограничения на сочетаемость слов).

Фонемные эквиваленты графемных представлений фраз и предложений конкретной предметной области, также как и в случае формирования фонемных представлений слов, формируются вручную. При этом строится сеть переходов, в которой учитываются все возможные комбинации слов всех предложений всех текстов, описывающих предметную область. Так сеть просьбы «разбудить в определенное время» из предметной области «Гостиница» будет выглядеть следующим образом (см. рисунок 1).



Рисунок 1 – Сеть переходов для просьбы «разбудить в определенное время»

Здесь в круглых скобках указаны подсловари, которые можно менять местами, а в квадратных – которые нельзя переставлять. Переставлять подсловари можно только внутри старших скобок. Символ (*) означает пустое слово. Не все слова в этой фразе являются ключевыми. Только «разбудите», «часов» и «минут».

В отличие от трудоемкого метода построения моделей слов с учетом правил комбинаторики элементов, не представляет большой сложности автоматически построить Марковскую модель слова. Для этого берется речевой корпус, содержащий определенное количество произнесений слов словаря (достаточное для формирования представительной модели), и в автоматическом режиме формируются модели слов. Объединение этих моделей в более крупные единицы представляет собой иногда непреодолимые трудности. Модель языка таким образом построить трудно (а для русского языка просто невозможно), так как требуемого для обучения модели количества произнесенных текстов просто не существует физически. Для модели предметной области это ограничение менее критично, но тоже достаточно серьезно. Именно поэтому известные системы

распознавания слитной речи работают более или менее прилично на нескольких предметных областях (для которых сформированы модели), и плохо для других предметных областей.

Менее сложна, по сравнению с распознаванием слитной речи, задача распознавания ключевых слов в потоке слитной речи. В этом случае трудоемкий процесс ручного формирования сети допустимых переходов между ключевыми словами в заданной предметной области может быть автоматизирован с использованием технологии автоматического построения семантической сети текста на основе корпуса текстов этой предметной области. Семантическая сеть предметной области (текста) является одним из способов представления в сетевом виде семантики предметной области (текста). Особенностью этого автоматического процесса формирования сети переходов является необходимость выявления ключевых, в заданной предметной области, слов, а также существенно важных для описания предметной области предложений. Предложения, не несущие информации о предметной области, в этом случае отбрасываются. Чем тщательнее будет произведен отбор, тем более точно будет работать система распознавания речи. После выбора ключевых слов, а также значимых предложений текста, на основе этих предложений строится однородная семантическая (ассоциативная) сеть. Далее, эта сеть используется для преимущественного выбора гипотез ключевых слов, которые входят в эту семантическую сеть и находятся на наименьших расстояниях от предыдущего распознанного слова.

1. Автоматическое формирование сети переходов

Идея автоматического формирования сети переходов заключается в следующем. В тексте выявляются ключевые понятия в их взаимосвязях. Сеть из ключевых понятий текста, учитывающая их связи (ассоциативная – однородная семантическая – сеть), может быть основой для построения сети переходов. Ключевые понятия выявляются на фоне других, второстепенных слов. Достигается это [Харламов, 2006] вычислением частоты встречаемости слов, а также попарной встречаемости слов в осмысленных фрагментах текста – предложениях. Далее, частотные характеристики слов – вершин сети – перенормируются в их смысловые веса, в результате чего появляется возможность ранжировать и ключевые слова, и предложения текста с точки зрения их важности в этой сети, и, следовательно, в тексте (или в корпусе текстов, описывающих предметную область). Другими словами, такой анализ позволяет выявить важные ключевые слова и существенные предложения текста на фоне второстепенных.

Построенная на основе этого множества предложений ассоциативная сеть является сетью переходов между ключевыми словами для этой

группы предложений. Сеть, построенная на всех предложениях корпуса текстов, описывающего предметную область, ранга выше порогового, является сетью переходов между ключевыми словами для всей предметной области.

2. Автоматическое формирование ассоциативной сети

Текст представляет собой внутренне структурированную последовательность символов, такую, что различные ее элементы имеют различную сложность и различную частоту встречаемости. В процессе автоматического анализа текста формируются словари этих элементов $\{B_i\}$. Это, например, словари флективных морфем, корневых основ. Выявленные лингвистические единицы в дальнейшем можно использовать для обработки текстовой информации. Словарь флективных морфем можно использовать для морфологического анализа, словарь корневых основ – для выявления ключевых понятий в тексте и формирования однородной (ассоциативной) семантической сети.

Ранее одним из авторов была реализована технология обработки текстовой информации TextAnalyst, позволяющая автоматически выявлять ключевые понятия в тексте на основе только информации о структуре самого текста (независимо от предметной области). Для этого формируется частотный портрет текста, содержащий информацию о частоте встречаемости понятий текста, представленных как корневые основы соответствующих слов, или их устойчивых сочетаний, встречающихся в тексте, а также об их совместной (попарной) встречаемости в смысловых фрагментах текста (например, в предложениях). Частотный портрет, таким образом, содержит информацию о частоте встречаемости понятий и их попарной (в терминах их ассоциативной связи) встречаемости в тексте. Использование хопфиллоподобного алгоритма [Hopfield, 1982] позволяет перейти от частоты встречаемости к смысловому весу (вес связей при этом остается неизменным).

Эта обработка включает несколько этапов. На первом этапе осуществляется первичная обработка: из текста удаляется нетекстовая информация, текст сегментируется на слова и предложения, из текста удаляются стоп-слова, рабочие и общеупотребимые слова, а оставшиеся слова подвергаются морфологической обработке. Для простоты анализа морфологическая обработка производится с использованием традиционного морфологического словаря – словаря первого уровня – $\{B_i\}_1$. Далее формируется словарь второго уровня – $\{B_i\}_2$ – словарь корневых основ (и устойчивых словосочетаний). На следующем этапе строится частотный портрет текста, то есть выявляются частоты p_i встречаемости корневых основ понятий

(полученных в результате морфологического анализа) и их устойчивых сочетаний, и частоты p_{ij} их попарной встречаемости в предложениях текста (то есть формируется словарь третьего уровня $\{B_i\}_3$). Эта сеть является исходной для формирования сети переходов.

3. Автоматическое выявление ключевых понятий текста

И, наконец, на третьем этапе обработки, частоты встречаемости перенормируются в смысловые веса с использованием итеративной процедуры, похожей на алгоритм искусственной нейронной сети, предложенной Хопфилдом [9]:

$$w_i(t+1) = \left(\sum_{i \neq j} w_i(t) w_{ij} \right) \sigma(\bar{E}), \quad (1)$$

здесь $w_i(0) = \ln p_i$; $w_{ij} = \ln p_{ij} / \ln p_j$ и $\sigma(\bar{E}) = 1/(1 + e^{-k\bar{E}})$ функция, нормирующая на среднее значение энергии всех вершин сети \bar{E} . В результате итеративной процедуры перенормировки наибольшие веса получают понятия, связанные с наибольшим числом других понятий с большим весом, то есть те понятия, которые стягивают на себя смысловую структуру текста.

Полученные таким образом смысловые веса ключевых понятий показывают значимость этих понятий в тексте. В дальнейшем эта информация используется для выявления предложений текста, содержащих наиболее важную информацию в тексте.

В результате получается так называемая ассоциативная (однородная) семантическая сеть N как совокупность несимметричных пар понятий $\langle c_i, c_j \rangle$, где c_i и c_j – понятия, связанные между собой отношением ассоциативности (совместной встречаемости в некотором фрагменте текста):

$$N \cong \{ \langle c_i, c_j \rangle \}. \quad (2)$$

В данном случае отношение ассоциативности несимметрично: $\langle c_i, c_j \rangle \neq \langle c_j, c_i \rangle$.

Семантическая сеть, описанная таким образом, может быть переописана как множество так называемых звездочек $z_i = \langle c_i < c_j \rangle$:

$$N \cong \{ z_i \} = \{ \langle c_i < c_j \rangle \}. \quad (3)$$

Под звездочкой $z_i = \langle c_i < c_j \rangle$ понимается конструкция, включающая главное понятие c_i , связанное с множеством понятий-ассоциантов $\langle c_j \rangle$, которые являются семантическими признаками главного понятия, отстоящими от

главного понятия в ассоциативной сети на одну связь. Ассоциативные связи направлены от главного понятия к понятиям-ассоциантам.

4. Сеть переходов в задаче распознавания ключевых слов в потоке слитной речи

Задача распознавания ключевых слов в потоке слитной речи проще, чем задача точного распознавания слитной речи: в этом случае распознаются не все произносимые слова, но только некоторые – ключевые – имеющие большое значение (заранее оговоренные) в тексте. Если в случае распознавания слитной речи для успешного распознавания требуется построение полной сети переходов (см. рисунок 1), которая позволяет существенно сократить оперативный словарь на каждом шаге распознавания от слова к слову, в случае распознавания ключевых слов сеть переходов упрощается. Из нее уходит второстепенная информация. В ней остаются только ключевые слова и переходы между ними.

Ассоциативная сеть, построенная так, как это описано в разделах 2 и 3 может быть использована для уменьшения числа гипотез при распознавании текущего ключевого слова. В этой сети имеются только ключевые слова. Причем после распознавания очередного слова оперативный словарь ограничивается только ближайшими ассоциантами распознанного ключевого слова, что существенно упрощает саму задачу распознавания, так как словарь системы при этом существенно редуцируется.

5. Пример построения ассоциативной сети

Для наглядности представления описанного подхода рассмотрим пример построения сети переходов, на основе текстов разговорников для раздела «Бронирование гостиницы». Сеть формируется на проходе с помощью персонального продукта, реализующего статистический нейросетевой подход к анализу неструктурированных текстов TextAnalyst. Исходные текст из единственного разговорника представлен ниже

Пример. «Я делал заказ. Места были зарезервированы для меня и моей семьи. Заказ был подтвержден в Париже. У вас есть свободные места? Мне нужна комната. Я хотел бы одноместный номер. Я хотел бы номер с ванной. Я хотел бы номер с двумя кроватями. Нам нужен двухместный номер с дополнительной кроватью. Есть что-нибудь подешевле? Не могли бы Вы показать мне комнату получше? Не могли бы Вы показать мне комнату побольше (поменьше)? Комнату, с видом на море. Какова плата за обслуживание и налог? Надбавка за обслуживание учтена? Сколько стоит номер, включая завтрак?»

Завтрак включен? Сколько это стоит в день? Вам нужен залог? Когда я должен освободить номер? Вам нужен мой паспорт? Можете порекомендовать другую гостиницу? Я сниму этот номер на неделю (месяц). Я пробуду два дня. Меня зовут Вася. Где мне расписаться? Можно одолжить Вашу ручку?»

Сформированная ассоциативная сеть этого текста представлена на таблице 1.

Таблица 1 – Ассоциативная сеть текста Примера 1

Родитель	Подчиненный
включая	
	завтрак
	включая завтрак
	завтрак включен
нужен	
	нужен номер
	нужен залог
	нужен мой паспорт
завтрак	
	включая
	включая завтрак
	завтрак включен
нужен залог	
	нужен
нужен номер	
	нужен
нужен мой паспорт	
	нужен
показать мне комнату	
	комнату
плата	
	обслуживание
делал заказ	
	заказ
включая завтрак	
	включая
	завтрак
надбавка за обслуживание учтена	
	обслуживание
завтрак включен	
	включен
	завтрак

зарезервированы	
	семьи
заказ	
	делал заказ
	подтвержден в Париже
подтвержден в Париже	
	заказ
комнату	
	показать мне комнату
	нужна комната
семьи	
	зарезервированы
обслуживание	
	плата за обслуживание и налог
	надбавка за обслуживание учтена
нужна комната	
	комната

Обработка большого по объему текста (собранный из разных разговорников) даст более совершенную сеть переходов.

Конечно, сеть выглядит несколько примитивно, кроме того, в ней есть информационный шум. Сеть построена на маленьких текстах, описывающих предметную область. Для более полного учета переходов, необходимо собрать более представительный корпус текстов. Необходима также чистка сети экспертом, и, возможно, ее дополнение. Однако если учесть, что ручное формирование сетей переходов, описывающих предметную область в системах распознавания речи – очень трудоемкое занятие, подспорье в построении сети имеет несомненные преимущества.

6. Дальнейшие исследования

В рамках данной работы сеть переходов строилась с использованием статистического подхода и для решения задачи распознавания ключевых слов в потоке слитной речи. Наряду со статистическими методами для автоматического анализа текстов используются лингвистические методы. Лингвистический подход к анализу текстов позволяет решить задачу семантического анализа более точно, но только для отдельных предложений текста. Объединение семантических представлений отдельных предложений в этом случае – до сих пор не решенная задача. Поэтому, совмещение двух подходов – статистического и лингвистического – позволит улучшить качество анализа при

возможности построения семантической сети всего текста. В этом случае семантическая сеть всего текста строится статистическим методом, но вместо пар слов в ней учитываются тройки, включающие пару слов и связь между ними. Эти тройки выявляются в процессе синтаксического анализа. Из предложений текста выявляются содержащиеся в них предикатные структуры, а также отношения сочинения, и атрибутивные отношения.

Под предикатной структурой понимается ядерная подлежащно-сказуемая конструкция, включающая в свой состав n актантов. В общем случае ядро – это глагольная конструкция, а актанты объединяются с ним системой отношений, заполняя его валентные гнезда [Харламов и др., 2012]. Таким образом, предикатная структура имеет вид (1):

$$Subj - R_0 - Pred - R_i - Obj_i, i = 1 \dots n, \quad (4)$$

где *Subj* – активный субъект (грамматическое подлежащее), который инициирует использование предиката *Pred* (грамматического сказуемого); R_0 – отношение «быть субъектом», R_i – предикативные отношения, Obj_i – актанты или узлы предикатной структуры (имя существительное, местоимение, числительное) в их атрибутивной форме, n – количество актантов.

Общую схему действий можно представить в виде последовательности шагов.

- Членение предложения по знакам пунктуации и сочинительным союзам на начальные сегменты (фрагментация). Определение вершин и типов начальных сегментов.
- На декартовом произведении омонимов внутри начальных сегментов построение множества однозначных морфологических интерпретаций каждого сегмента.
- Построение синтаксических групп для каждой интерпретации сегмента с помощью синтаксических правил, выявляющих синтаксические связи между словами. Оценка синтаксического покрытия каждой интерпретации.
- Установление иерархии между сегментами с помощью синтаксических правил: вложения контактно расположенных сегментов (причастных, деепричастных оборотов, обособленного определения); определение однородности между контактно расположенными сегментами; определение отношения импликации между сегментами по подчинительным союзам, в них входящим.

Предикатная структура простого предложения выявляется на основе анализа главных членов предложения путем выявления шаблона, соответствующего одной из минимальных структурных схем предложений (МСС), описывающих предикативный минимум предложения. Использование МСС в качестве формального образца позволяет выявить

предикативную основу (структурную схему) простого предложения, и в дальнейшем – его предикатную структуру. Смысловая связь между понятиями предложения (объектом/субъектом) в общем случае может быть описана предикатом, актантами которого выступают данные понятия. Установление таких синтактико-семантических связей позволяет сформировать схему ситуации, описываемой во фразе.

Обусловленный валентностью предиката семантико-синтаксический уровень анализа конструкций, не соответствующий узкому собственно формально-синтаксическому подходу, дает возможность даже из набора неправильных форм (посредством приведения их к начальным формам) с помощью заполнения валентных гнезд определить схему предложения.

Выявление ключевых предикатных структур в тексте возможно с использованием того же подхода, который был использован для выявления ключевых понятий и ключевых предложений в тексте в технологии TextAnalyst.

После выявления ключевых предикатных структур текста, в ассоциативной сети текста отношения между парами ключевых понятий размечаются их типами из соответствующей предикатной структуры таким образом, что звездочка ассоциативной сети заменяется предикатной структурой, причем, главное понятие ассоциативной сети совмещается с субъектом, а ближайшие ассоцианты – главным и второстепенными объектами и их атрибутами предикатной структуры. Ассоциативные связи заменяются соответствующими типами отношений предикатной структуры – собственно предикатным отношением между субъектом и главным объектом, а также другими отношениями – валентными связями актантов предиката – субъекта с, соответственно, второстепенными объектами и их атрибутами.

Заключение

Автоматический смысловой анализ текста с использованием технологии TextAnalyst позволяет выявить ключевые понятия в их взаимосвязях в тексте, а также ранжировать их по степени их смысловой значимости в данном тексте. Ранжирование ключевых понятий, в свою очередь, позволяет ранжировать предложения текста, и выбирать наиболее существенные из них для множества текстов, описывающих предметную область. Такая сеть может быть исходной для построения сети переходов между ключевыми словами в задаче распознавания ключевых слов в потоке слитной речи. В дальнейшем предполагается исследовать возможность объединения статистического и лингвистического подходов к автоматическому смысловому анализу текстов, что позволит строить более точную сеть переходов за счет разметки ассоциативных связей между

ключевыми словами типами их отношений в предикатных структурах соответствующих предложений.

Работа выполнена при поддержке гранта Минобрнауки 2012-1.4-07-514-0018 «Исследование и разработка программного обеспечения понимания неструктурированной текстовой информации на русском и английском языках на базе создания методов компьютерного полного лингвистического анализа».

Библиографический список

[Винцюк, 1987] Винцюк Т.К. Анализ, распознавание и смысловая интерпретация речевых сигналов. – Киев. Наукова думка, 1987. – 267 с.

[Харламов, 2006] Харламов А.А. Нейросетевая технология представления и обработки информации (естественное представление знаний). - М.: «Радиотехника», 2006. - 89 с.

[Hopfield, 1982] Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. 79, 1982. Pp. 2554 – 2558

[Харламов и др., 2012] Харламов А.А., Ермоленко Т.В., Дорохина Г.В., Гнидько Д.С. Метод выделения главных членов предложения в виде предикатных структур, использующий минимальные структурные схемы. Речевые технологии, № 2, 2012.

SUBJECT AREA SEMANTIC NETWORK AS A BASIS FOR COMPILING TRANSITION NETWORK IN AUTOMATIC SPEECH RECOGNITION

Kharlamov A.A. *, Yermolenko T.V. **

**Institute of Higher Nervous Activity of the Russian Academy of Science, Moscow, Russia*

kharlamov@analyst.ru

***Institute of Artificial Intelligence Problems, Donetsk, Ukraine*

etv@iai.dn.ua

The problem of speech recognition is not satisfactory fulfilled at the moment. But there are some products of the famous companies like Nuance and Autonomy which are used in applications. But the transition network between words of the subject area there compiled manually. Approach to automatical compilation of such a network is represented in the paper.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.934.5

КАМПАНЕНТЫ ДЛЯ РОЗНЫХ ПЛАТФОРМАЎ СІНТЭЗАТАРА МАЎЛЕННЯ ПА ТЭКСЦЕ ДЛЯ ІНТЭЛЕКТУАЛЬНЫХ СІСТЭМ

Гецэвіч Ю.С.* , Пакладок Д.А.* , Брэк Д.В.**

* *Аб'яднаны інстытут праблем інфарматыкі НАН Беларусі, г. Мінск, Рэспубліка Беларусь*

yuras.hetsevich@newman.bas-net.by

dima.pokladok@gmail.com

** *Беларускі дзяржаўны ўніверсітэт інфарматыкі і радыёэлектронікі,*

г. Мінск, Рэспубліка Беларусь

bartwader@gmail.com

Разгледжаны пытанні актуальнасці і практычнай пабудовы кампанентаў двухмоўнага сінтэзатара беларускага і рускага маўлення па тэксце для мабільных і інтэрнэт-платформаў. Вынікі тэставання паказваюць, што сінтэзатар маўлення па тэксце для Інтэрнэта можа цалкам задавальняць запыты па агучванні старонак сайтаў, а таксама ён можа быць выкарыстаны ў якасці завершанага кампанента маўленчага інтэрфейсу любым знешнім распрацоўшчыкам праз Інтэрнэт для пабудовы ўласных інтэлектуальных сістэм.

Ключавыя словы: кампанент; сінтэз маўлення па тэксце; мабільны сінтэзатар маўлення па тэксце; пабудова Інтэрнэт-версіі сінтэзатара маўлення па тэксце.

Уводзіны

Сінтэзатары маўлення па тэксце (СМТ) на сённяшні момант дасягнулі высокай якасці і сталі складанымі шматкампанентнымі сістэмамі. Кожны з кампанентаў СМТ паэтапна працуе над вынікам папярэдняга і сам з'яўляецца завершаным рашэннем. Гэтая асаблівасць СМТ дазваляе выкарыстоўваць розныя яе кампаненты у розных інтэлектуальных сістэмах.

Сістэмы сінтэзу маўлення па тэксце на сённяшні момант дасягнулі высокай якасці як па крытэрыях выразнасці і натуральнасці сінтэзаванага голасу, так і па тэхнічных характарыстыках, што спрыяе іх шырокаму ўжыванню ў цэнтрах апрацоўкі запытаў, пры агучванні станаў працы складаных аб'ектаў, пры стварэнні аўдыёкніг і г.д. [Лобанов и др., 2008]. У той жа час застаецца важнай задачай адаптацыя сістэм сінтэзу маўлення па тэксце (іх алгарытмаў і рэсурсаў) для розных вылічальных платформаў, якія маюць розныя сістэмныя абмежаванні дыскавай і/ці апэратыўнай памяці, што далей накладвае абмежаванні на выкарыстанне ўсіх распрацаваных алгарытмаў і лінгвістычных рэсурсаў СМТ. Гэтая задача ўзнікае з-за таго, што звычайна распрацоўшчыкі прымаюць за аксіому, што рэсурсы вылічальнай сістэмы *неабмежаваныя: тэарэтычна,*

калі прыняць вылічальную машыну за машыну Т'юрынга [Barker-Plummer, 1995], ці *практычна,* калі яны нашмат перавышаюць максімальныя паказчыкі выкарыстання памяці і тактавага працэсара алгарытмамі і рэсурсамі СМТ.

Аўтаматычныя маўленчыя тэхналогіі для кожнай канкрэтнай мовы без адаптацыі СМТ для мабільных і інтэрнэт-платформаў застаюцца недаступнымі для шырокага кола карыстальнікаў ва ўсім свеце [Internet World Stats, 2012a], у тым ліку ў Беларусі, дзе кожны другі чалавек карыстаецца Інтэрнэтам [Internet World Stats, 2012b], а колькасць абанентаў сотавай сувязі на 2012 г. перакročыла 10 млн [IT.TUT.BY, 2012].

Шырокае распаўсюджанне можа атрымаць выкарыстанне сістэм сінтэзу маўлення па тэксце на мабільных прыладах, такіх як кішэнныя персанальныя камп'ютары, смартфоны, мабільныя тэлефоны. Узорами прыстасаванняў з СМТ могуць быць такія, якія чытаюць электронныя кнігі, указанні аўтамабільнай навігацыйнай сістэмы; агучваюць SMS-паведамленні ці лісты электроннай пошты.

Пабудова сінтэзатара маўлення па тэксце для Інтэрнэта абумоўліваецца наступнымі перспектывамі:

- сляпыя ці слабабачыя атрымаюць добрую мажлівасць, каб азнаямлівацца з сайтамі ў Інтэрнэце, у той час як уладальнікі сайтаў змогуць павысіць камфортнасць для карыстальнікаў: камусьці можа быць зручна не чытаць сайты, а слухаць;

- Інтэрнэт-контэнт утрымлівае самыя дынамічныя і разнастайныя па напісанню тэксты. У іх ёсць шмат новых слоў, асаблівых тэкставых выказаў, абдруковак, што з’яўляецца добрым матэрыялам для тэставання і паляпшэння працы СМТ. У выніку СМТ змога сканаваць вялікую колькасць тэкстаў, аналізаваць і апрацоўваць іх, паказваць складаныя (ці неадназначныя) месцы экспертам;

- распрацоўшчыкі СМТ змогуць дэманстраваць сінтэзатар у розных рэжымах выкарыстання (поўны ці абмежаваны доступ) вялікай колькасці карыстальнікаў і мэставых груп (для навуковых даследаванняў, для школьнікаў і студэнтаў, для прадпрыемстваў і патэнцыяльных замоўцаў, якія хочуць пабудаваць камерцыйныя сэрвісы агучкі тэкстаў).

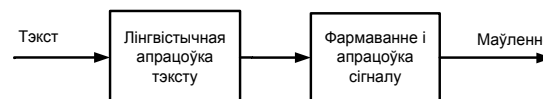
Большасць сучасных сінтэзатараў рускага маўлення па тэксце для мабільных прыладаў, напрыклад, Acapela TTS for Windows Mobile, Nuance TALKS, Mobile Speak і інш., працуюць на смартфонах пад кіраваннем аперацыйных сістэм Windows Mobile ці Symbian. Аднак смартфоны, на якіх магчыма праца гэтых СМТ, складаюць, па розных падліках, толькі 10–15 % рынку мабільных тэлефонаў. Для беларускай мовы такія сістэмы толькі пачынаюць стварацца ў выглядзе асобных блокаў СМТ [Гецэвіч і інш., 2011]. Пры гэтым асноўная маса мабільных апаратаў, якія выкарыстоўваюцца ў цяперашні час, характарызуецца нізкай хуткадзейнасцю і адносна невялікім легальна дазволена для выкарыстання аб’ёмам апэратыўнай памяці (да 24 Мб) [Mihai Preda, 2010]. У той жа час сучасныя СМТ патрабуюць вялікага аб’ёма памяці для захоўвання і выкарыстання лінгвістычных і акустычных рэсурсаў, што не дазваляе наўпрост «перанесці» існуючыя сістэмы на мабільныя платформы [Цирульник и др., 2012].

Інтэрнэт-платформы звычайна характарызуюцца аперацыйнай сістэмай (UNIX, Windows), архітэктурай выканання (C, ASP.NET, Java, Python, Pike, Ruby, Erlang), мовамі праграмавання (ASP, C, ColdFusion Markup Language, Java, JavaScript, Lua, Perl CGI, PHP, Python, Ruby, SMX, Lasso, WebDNA, Progress WebSpeed), фізічнай памяццю і выдзеленай апэратыўнай памяццю для скрыпта (звычайна каля 128 Мб), які выконваецца. На дадзены момант існуе некалькі інтэрнэт-версій СМТ для рускай мовы. Камерцыйныя кампаніі Acapela Group [Acapela Group, 2012] і Nuance на сваіх сайтах дазваляюць выкарыстоўваць дэма-вокны для сінтэзавання любога тэксту [Acapela Group, 2012], кампанія ЦРТ дэманструе інтэрнэт-сэрвісы на аснове СМТ для стварэння аўдыёпапаштовак і агучкі rss-

каналаў [Центр Речевых Технологий, 2001]. Гэтыя СМТ патрабуюць асобнага сервера, дзе могуць працаваць скампіляваныя коды, якія выконваюцца, пад канкрэтную аперацыйную сістэму. Таму яны з’яўляюцца стацыянарнымі СМТ з інтэрнэт-доступам. Яны складаны пераносіцца на іншыя серверы, бо патрэбна пастаянна дамаўляцца з правайдэрам хостынга аб асаблівых умовах па размяшчэнні СМТ у якасці аднаго з галоўных працэсаў сервера. Гэта каштуе нашмат больш, чым выкананне скрыптоў на мовах праграмавання, які ўстаноўлены на гэтым серверы па змоўчванні. У літаратуры таксама не сустракаецца апісання СМТ для крос-платформавых моваў праграмавання, напрыклад, для Perl CGI, PHP, Python і інш. Таму пабудова СМТ на інтэрнэт-платформе на скрыптавай мове праграмавання з’яўляецца новай навуковай задачай.

1. Падыход да распрацоўкі шматкампанентнага сінтэзатара маўлення па тэксце для мабільных і інтэрнэт-платформаў

СМТ – гэта сістэма, здольная генераваць маўленне па тэксце. На працягу апошніх дзесяцігоддзяў яна замацавалася як сістэма, якая складаецца з двух агульных кампанентаў (малюнак 1). Гэта кампанент лінгвістычнай апрацоўкі тэксту да фанемнага выгляду з пазнакамі націску слоў, інтанацыі (прасодыі) і рытма, а таксама кампанент апрацоўкі маўленчага сігналу, які пераўтварае раней атрыманы фанемны выгляд у гукавы сігнал маўлення [Dutoit, 1997].



Малюнак 1 – Агульная функцыянальная схема сінтэзу маўлення па тэксце

Такая архітэктурна дазваляе распрацоўшчыкам СМТ будаваць іх «з канца»: спачатку распрацаваць кампанент атрымання маўлення па фанетычным кодзе, уласцівым для абранай мовы, а пасля распрацоўваць алгарытмы лінгвістычнай апрацоўкі тэкстаў для абранай мовы ў фанемны выгляд. Такі падыход з’яўляецца зручным, бо падчас стварэння спачатку другога кампаненту распрацоўшчык атрымлівае адказ на пытанне, ці магчыма ўвогуле пабудаваць сінтэзатар маўлення па тэксце для дадзенай платформы. Калі акустычны кампанент пабудаваць удаецца, то і астатнія кампаненты можна будзе пабудаваць у будучым, бо звычайна большасць платформаў дазваляе апрацоўваць тэкставыя дадзеныя.

Такім чынам, паводле вышэй прадстаўленай архітэктурны ў наступных раздзелах артыкула будзе апісаны агульны падыход пераносу атрыманых алгарытмаў і лінгвістычных рэсурсаў пры распрацоўцы акустычнага працэсара (АП) мабільнага СМТ [Цирульник и др., 2010][Гецэвіч і

інш., 2011] для пабудовы АП СМТ для Інтэрнэт-платформы Linux (Apache) пад мову праграмавання РНР [Гецэвіч і інш., 2012]. А далей апісваюцца новыя кампаненты фанетычнага, прасадычнага і тэкставага працэсараў СМТ для Інтэрнэт-платформы. У выніку прадстаўляецца эксперыментальны абразец шматкампанентнага Інтэрнэт СМТ, які можа быць выкарыстаны ў якасці завершанага кампанента маўленчага інтэрфейсу любым знешнім распрацоўшчыкам праз Інтэрнэт для пабудовы ўласных інтэлектуальных сістэм.

2. Алгарытм працы і рэсурсы шматмоўнага акустычнага працэсара

Падабенства беларускай (БМ) і рускай (РМ) моў дазваляе стварыць універсальную мадэль акустычнага працэсара, прыдатную для абедзвюх моў адразу. Гэтаму спрыяе падабенства фанетыкі, утварэння гукаў [Lobanov et al., 2006]. Алгарытмы мадыфікацыі сігнала працуюць з алафонамі па агульнай схеме, абстрагуючыся ад пэўнай мовы. Уваходнымі дадзенымі для акустычнага працэсара з'яўляюцца алафонныя паслядоўнасці кожнай сінтагмы тэксту, значэнні даўжыні кожнага алафона і кожнага перыяду асноўнага тону. Выходныя дадзеныя – файл ў фармаце WAV. У якасці рэсурсаў выступаюць адпаведныя для кожнай мовы фанетычныя групы і галасавыя базы, якія створаныя экспертамі. Заўважым, што для дзвюх моў выкарыстоўваюцца розныя па напаўненні, але агульныя па сэнсе фанетычныя групы (паняцці) – галосныя, звонкія і глухія выбухныя і фрыкатыўныя зычныя, афрыкаты, санорныя зычныя (табліца 1).

Табліца 1 – Фанетычныя групы для беларускай і рускай моў

Фанетычныя групы	Склад груп	
	Беларуская мова	Руская мова
Галосныя	А,Е,І,О,У,У	А,Е,І,О,У,У
Выбухныя звонкія зычныя	В,В',Г,Г',Д	В,В',Г,Г',Д,Д'
Выбухныя глухія зычныя	К,К',Р,Р',Т	К,К',Р,Р',Т,Т'
Афрыкаты	ДЗ,ДЗ',ДЗН,С,С',СН	С,СН',СН'
Фрыкатыўныя звонкія зычныя	В,В',ГН,ГН',ЗН,З,З'	В,В',ЗН,З,З'
Фрыкатыўныя глухія зычныя	С,С',Ф,Ф',Н,Н',Ш	С,С',Ф,Ф',Н,Н',Ш
Санорныя зычныя	Л,Л',М,М',Н,Н',Р,Р',J',W	Л,Л',М,М',Н,Н',Р,Р',J'

Алгарытм працы акустычнага працэсара для канкрэтнай мовы наступны [Цирульник и др., 2010]:

- 1) Атрыманне чарговага алафона з сінтагмы.
- 2) Вызначэнне групы алафона з дапамогай лінгвістычных рэсурсаў.
- 3) Змена даўжыні і, пры неабходнасці, кожнага перыяду асноўнага тону алафона для дадзенай фанетычнай групы з дапамогай алгарытму плаўнай сшыўкі.

4) Запіс чарговага алафона ў выніковы гукавы паток.

5) Выхад: гукавы паток – WAV-файл.

Калі разгледзець склад галасавых баз БМ і РМ для мабільных і інтэрнэт-платформаў (табліца 2), то колькасць алафонаў для беларускай і рускай моў адрозніваецца нязначна. Гэта абумоўлена адной групай моў – усходнеславянскай. З іншага боку, ёсць істотнае адрозненне па колькасці (і, адпаведна, па агульным памеры) для розных платформаў – галасавая база для мабільнай платформы менш на 56 % (92,6 % па памеры) для беларускай мовы і на 56 % (92,8 % па памеры) для рускай.

Табліца 2 – Параўнальная характарыстыка галасавых баз беларускай (БМ) і рускай (РМ) моў для мабільных і інтэрнэт-платформаў

Мова	Інтэрнэт-платформа		Мабільная платформа	
	Колькасць алафонаў	Памер, КБ	Колькасць алафонаў	Памер, КБ
БМ	1889	11822	832	869
РМ	1779	10699	784	773

3. Акустычны працэсар для мабільнай платформы

Памяншэнне галасавых баз для мабільнай платформы дасягалася шляхам памяншэння частаты дыскрэтызацыі і разраднасці зыходнай базы (22 кГц 16 біт у 8 кГц 8 біт) і выдаленнем дыалафонаў. Дадзеная аперацыя дазволіла запрацаваць акустычнаму працэсару на большасці сучасных мабільных тэлефонах. Так, распрацаваная праграма была паспяхова пратэставана на тэлефонах Motorola, Sony-Ericsson, LG, якія характарызуюцца тактавай частатой ARM-працэсара ад 68 да 115 МГц, аб'ёмам памяці ад 3 500 да 4 200 Кб, падтрымкай канфігурацыі CLDC 1.0 і профіля MIDP 2.0 [Гецэвіч і інш., 2011]. Сістэма дазваляе сінтэзаваць маўленчы сігнал у рэальным часе на мабільных тэлефонах з АРМ-працэсарамі сёмага пакалення.

Галоўнае меню сістэмы (малюнак 2а) складаецца з пунктаў-выклікаў:

- тэкставага файла для прайгравання;
- наладак;
- прайгравання маўленчага сігналу;
- справачнай інфармацыі па сістэме.

Для сінтэзу маўлення карыстальнік павінен указаць тэкставы файл з алафонным тэкстам, а затым выбраць пункт меню «Воспроизведение». Меню наладак складаецца з выбара галасавой базы і выбара прасадычнага стылю для сінтэзу маўлення па тэксце (малюнак 2б). Для кіравання агучваннем тэксту ў сістэме рэалізаваны функцыі паўзы/працягу, а таксама поўнага спынення сінтэзу маўлення ў рэальным часе (малюнак 2в).



а) б) в)

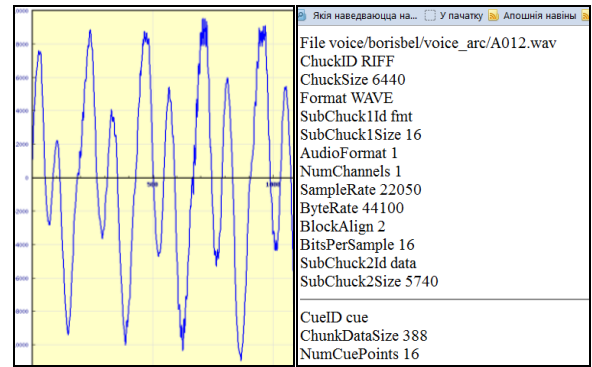
Малюнак 2 – Інтэрфейсы сінтэзатара маўлення па тэксце для мабільных платформаў: а – галоўнае меню; б – меню наладак; в – прайграванне.

4. Кампаненты сінтэзатара маўлення па тэксце для Інтэрнэт-платформы

У адрозненне ад мабільнай платформы інтэрнэт-платформа дазваляе выкарыстоўваць амаль неабмежаваныя аб'ёмы дыскавай памяці, таму галасавая база для яе не мадыфікавалася. Асноўная ўвага была звернута на аптымальнае выкарыстанне аператыўнай памяці, бо на яе абмежаванні застаюцца ў выглядзе наладак ад адміністратара сервера для кожнага скрыпта-працэса, які выконваецца ў канкрэтную адзінку часу.

Акустычны працэсар (АП) для Інтэрнэта будзеца на скрыптавай мове праграмавання PHP. Гэта робіцца для таго, каб любыя напісаны скрыпт мог лёгка пераносіцца на любую іншую платформу (ці хостынг), дзе ёсць PHP. У літаратуры не сустракаецца апісанняў працы з гукавымі файламі на PHP, тым больш няма апісанняў рэалізацыі СМТ для гэтай мовы праграмавання, акрамя ранейшай публікацыі аўтараў гэтага артыкула [Гецэвіч і інш., 2012].

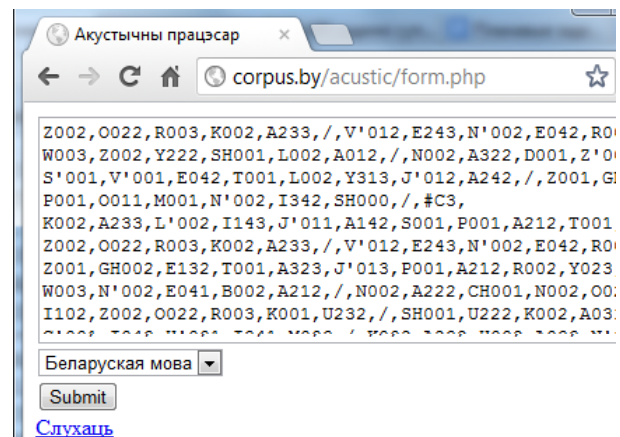
Спачатку для прагляду асцыляграм гукавых файлаў галасавой базы быў пабудаваны спецыяльны інтэрфейс, які даступны па спасылцы ў рэальным часе – <http://corpus.by/acoustic/wave.php>. У выніку зараз любы алафон магчыма «правізулізаваць» і прагледзець яго амплітудна-часавую хвалю ў Інтэрнэце. Напрыклад, была пабудаваная асцыляграма алафона А212 (малюнак 3а). Далей з дапамогай стандартных функцый чытання файлаў на PHP была пабудаваная эксперыментальная праграма для прагляду зборнай інфармацыі (асноўныя характарыстыкі запісанага сігналу, размечаныя рэгіёны і маркеры) любога абранага WAV-файла з галасавой базы (www.corpus.by/acoustic), напрыклад, для алафона А012 з беларускай галасавой базы (малюнак 3б).



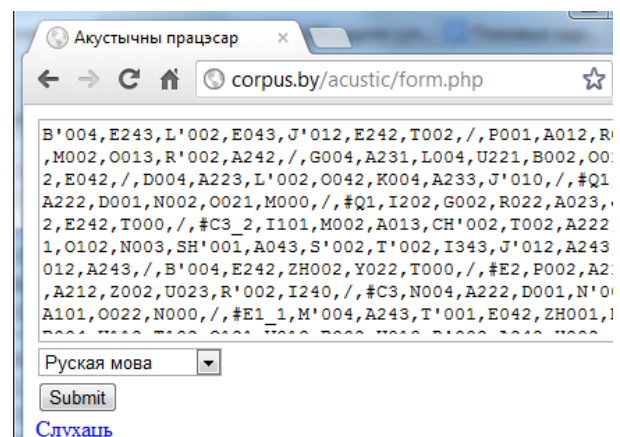
а) б)

Малюнак 3 – Вынікі працы дапаможных эксперыментальных мадэляў, якія рэалізаваныя на PHP для працы з любым файлам гукавой базы: а) вывад асцыляграмы; б) вывад асноўных характарыстык

Пабудаваныя эксперыментальныя мадэлі пацвердзілі магчымасць працы з гукавымі файламі ў WAV-фармаце праз інтэрпрэтуемую мову праграмавання PHP. Таму далей была рэалізаваная эксперыментальная праграма мадэль базавых алгарытмаў акустычнага працэсара на PHP. На малюнках 4 і 5 прадстаўлены інтэрфейсы для праверкі працы распрацаванага акустычнага працэсара СМТ у Інтэрнэце для беларускай ці рускай гукавой базы.



Малюнак 4 – Эксперыментальны акустычны кампанент сінтэзатара беларускага маўлення па тэксце ў выглядзе інтэрнэт-сайта



Малюнак 5 – Эксперыментальны акустычны кампанент сінтэзатара рускага маўлення па тэксце ў выглядзе інтэрнэт-сайта

Двухмоўны АП даступны па спасылцы www.corpus.by/acoustic/form.php. У яго інтэрфейсе карыстальнік можа напісаць фанетычны тэкст ці выкарыстаць уведзены, пазначыць мову і падаць запыт праз Submit на сервер. Праграмны скрыпт акустычнага працэсара прымае запыт, апрацоўвае яго па абазначаным вышэй алгарытме ў п. 2. У выніку будзе згенераваны гукавы WAV-файл з пазнакай выдаткаванага часу на выкананне скрыпта. Па спасылцы з надпісам «Слухаць» (www.corpus.by/acoustic/test.wav) файл можна праслухаць ці захаваць на лакальным камп'ютары. Заўважым, што для генерацыі маўлення абедзвюх моў выкарыстоўваецца адна і тая ж рэалізацыя алгарытмаў акустычнага працэсара на RНР, змяняюцца толькі галасавыя базы.

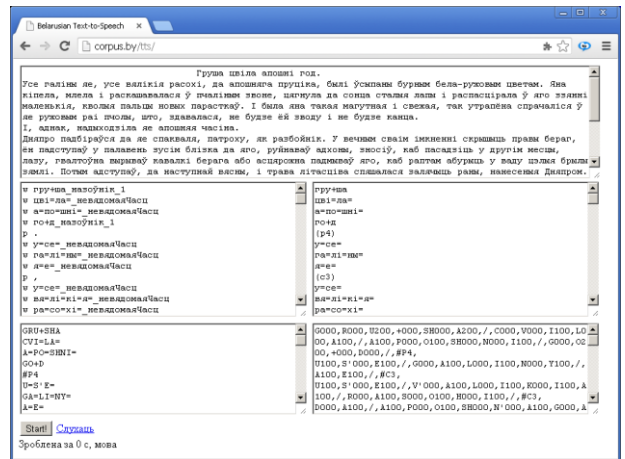
У табліцы 3 прадстаўлены вынікі экспрэс-тэставання акустычнага працэсара на абранай інтэрнэт-платформе (UNIX, Apache, RНР, 64 Мб аператыўнай памяці) для абедзвюх моў. Для кожнай мовы тэсты праводзіліся на адным і тым жа адпаведным тэксце, які быў шмат разоў раскапіраваны для павелічэння тэставай колькасці сінтагмаў. Вынікі паказваюць, што сістэма на такой платформе за адзін запыт здольная выканаць сінтэз маўлення алафоннага тэкста, аб'ём якога не большы за 780 Кб для БМ і не большы за 420 Кб для РМ. Для параўнання, сярэдняя велічыня старонкі ў Інтэрнэце не перавышае 320 Кб па дадзеных Google на 2010 г. [Ramachandran, 2010].

Табліца 3 – Экспрэс-тэставанне акустычнага працэсара ў Інтэрнэце

БМ			РМ		
Кольк асць сінтагмаў	Аб'ём тэксту, Кб	Час адказу сервера, с	Кольк асць сінтагмаў	Аб'ём тэксту, Кб	Час адказу сервера, с
5 400	520	6	3800	280	3
8 100	780	12	5700	420	7
10 800	1040	няма адказу	7600	560	няма адказу

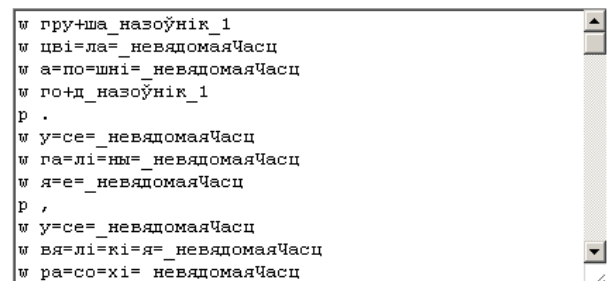
Такім чынам, пабудаваныя эксперыментальныя праграмныя мадэлі і экспрэс-тэсты пацвярджаюць, што магчыма рэалізаваць сінтэзатар маўлення па тэксце для беларускай і рускай моў для інтэрнэт-платформаў з дапамогай скрыптавай мовы праграмавання RНР.

На малюнку 6 паказаны вэб-інтэрфейс сінтэзатара беларускага маўлення па тэксце, які даступны па спасылцы www.corpus.by/tts. Дыялогавае акно складаецца з пяці тэкставых акон і кнопкі «Start!» для запуску сінтэзатара. У першае акно ўводзіцца тэкст для сінтэзу. Выніковы файл з сінтэзаваным маўленчым сігналам можна праслухаць ці спампаваць па згенераванай спасылцы «Слухаць».



Малюнак 6 – Вэб-інтэрфейс сінтэзатара маўлення па тэксце (спасылка www.corpus.by/tts)

Вынік працы тэкставага кампанента сінтэзатара можна прагледзець на малюнку 7. На дадзеным этапе ўваходны тэкст быў тыпізаваны на словы (w) і знакі прыпынку (p). У кожным слове быў прастаўлены націск праз знакі «+» і «<» (ex. гру+ша), прыпісаны маркер катэгорыі мовы (ex. назоўнік) і прыкрытэту слова (ex. 1) паводле базы дадзеных (унутранага слоўніка S). У тым выпадку, калі слова не было знойдзена ў слоўніку S, тады выстаўляўся частковы націск на кожны склад (ex. цві=ла=) і ставіўся маркер, што часціна мовы невядомая (ex. цві=ла=невядомаяЧасц). Дадаваны кампанент можа выкарыстоўвацца незалежна ад Інтэрнэт СМТ інтэлектуальнымі сістэмамі для вызначэння тыпу паслядоўнасці сімвалаў сказу, для знаходжання націску і лексіка-граматычнай інфармацыі любога слова сказаў.



Малюнак 7 – Вынік працы тэкставага кампанента Інтэрнэт СМТ

Вынік працы тэкставага працэсара быў прааналізаваны прасадычным працэсарам і ў выніку былі вылучаны сінтагмы і пазначаны іх інтанацыйныя тыпы маўлення (С, Р, Q, E). На малюнку 8 прадстаўлены вынік працы прасадычнага кампанента сінтэзатара маўлення. Напрыклад, С3 – інтанацыя незавершанасці, Р4 – інтанацыя завершанасці.

```

гру+ша
цві=ла=
а=по=шні=
го+д
{р4}
у=се=
га=лі=ны=
я=е=
{с3}
у=се=
вя=лі=кі=я=
ра=со=хі=

```

Малюнак 8 – Вынік працы прасадычнага кампанента Інтэрнэт СМТ

Наступны крок – фанетычная апрацоўка сінтагмаў. Усе літары кожнай сінтагмы па правілах пераўтварэння літара-фанема замяняюцца на адпаведныя фанемы (малюнак 9).

```

GRU+SHA
CVI=LA=
A=PO=SHNI=
GO+D
#P4
U=S'E=
GA=LI=NY=
A=E=

```

Малюнак 9 – Вынік працы фанетычнага кампанента Інтэрнэт СМТ

Правілы пераўтварэння фанетычнага кампаненту запісваюцца ў выглядзе фармальных замен кожнай літары на адпаведную фанему. У гэтым кампаненце правілы амаль без ніякіх дадатковых апрацовак выкарыстоўваюцца ў якасці аргументаў для рэгулярных выразаў пошуку-замены адных сімвалаў на іншыя:

```

# Агульныя правілы
А-А
Б-В
В-В
Г-Г
Д-Д
Е-Е
...
# Выключэнні з агульных правілаў
(O)[+=]-O
[ЖШЦ](И)-Y
[^АЕЁИОУЫЎЮЯ][_#](И)-Y
[Б](O)[+=]-J',O
[Б](И)-J',I
[СШЖ](Ч)-SH'
(T)[Б][C]-C
[EO](Г)[O][_#]-V
(Б)$-P
(В)$-F
(З)$-S
(Д)$-T
(Г)$-K
(З)[КПСТФХЦШЩ]-S
(З)[_#][КПСТФШ]-S
(Л)[Н][Ц]-O
...
# Спіс фанем змягчэння
Е

```

Ё
Ю
Я
И
Б
...

```

#Агульныя правілы змягчэння
([ХМНЛРБПВФДТЗСГК])[ЕЁЮЯИЬ]
(Н)[НТСЗДЛЧ]
(Л)[Л]
(Р)[Р]
([МБПВФ])[МБПВФ]
...

```

Алафонны кампанент Інтэрнэт СМТ вызначае алафон фанемы X_i паводле яе натуральнага фанетычнага асяроддзя – фанемаў X_{i-1} і X_{i+1} (малюнак 10). Гэта ажыццяўляецца з дапамогай правілаў пераўтварэння фанема-алафон і знаходжаннем пазіцыі ў слове націскай галоснай фанемы.

```

G000,R000,U200,+000,SH000,A200,/,C000,V000,I100,LO
00,A100,/,A100,P000,O100,SH000,N000,I100,/,G000,O2
00,+000,D000,/,#P4,
U100,S'000,E100,/,G000,A100,L000,I100,N000,Y100,/,
A100,E100,/,#C3,
U100,S'000,E100,/,V'000,A100,L000,I100,K000,I100,A
100,/,R000,A100,S000,O100,H000,I100,/,#C3,
D000,A100,/,A100,P000,O100,SH000,N'000,A100,G000,A

```

Малюнак 10 – Вынік працы алафоннага кампанента Інтэрнэт СМТ

Правілы фанема-алафон уяўляюць сабой вынік эксперыментальнага даследавання розных фанетычных груп па іх фанетычным асяроддзі [Лобанов и др., 2008]. Адметна, што дадзеныя правілы могуць выкарыстоўвацца як для беларускай, так і для рускай моў. Яны маюць наступны выгляд:

```

# Алфавіт фанем
A0,A1,A2,A3,E0,E1,E2,E3,I0,I1,I2,I3,O0,O1,Y0,Y1,
Y2,Y3,U0,U1,U2,U3,B,B',D,D',G,G',Z,Z',ZH,L,L',M,M',
N,N',P,P',T,T',K,K',C,CH',F,F',S,S',SH,SH',H,H',V,V',J',
R,R',#,
# Цэнтральныя групы фанем
C0:A0,A1,A2,A3,E0,E1,E2,E3,I0,I1,I2,I3,Y0,Y1,Y2,
Y3,
C1:U0,U1,U2,U3,O0,O1,
C2:P,P',T,T',K',C,CH',F,F',S,S',SH,SH',H',CH*,C'*,
C3:K,G,H,GH*,
...
# Групы фанем левага кантэксту
L0:#,
L1:P,B,F,V,M,U0,U1,U2,U3,O0,O1,W*,
L2:SH,ZH,R,T,C,S,D,Z,N,L,A0,A1,A2,A3,E0,E1,E2,
E3,Y0,Y1,Y2,Y3,DZ*,DZH*,CH*,
L3:K,G,H,GH*,
...
# Групы фанем правага кантэксту
R0:#,
R1:P,B,F,V,M,O0,O1,U0,U1,U2,U3,W*,
R2:SH,ZH,R,T,C,S,D,Z,N,L,K,G,H,A0,A1,A2,A3,E0,

```

E1,E2,E3,Y0,Y1,Y2,Y3,DZ*,DZH*,CH*,

...

Правілы пераўтварэння для фанем з кожнай цэнтральнай групы

C0

L0R0-00

L1R0-10

L2R0-20

...

C2

L10R0-00

L10R4-01

L10R5-02

...

C3

L10R0-00

L10R6-01

L10R7-02

...

...

Такім чынам, былі распрацаваны чатыры кампаненты апрацоўкі дадзеных для сінтэзу маўлення па тэксце для Інтэрнэт-платформаў на аснове сумесных алгарытмаў для стацыянарных і мабільных платформаў. У спалучэнні з пятым кампанентам – АП, яны ўтвараюць паўнаватасны СМТ для Інтэрнэта.

5. Заключэнне

Пастаўлена і вырашана задача пераносу атрыманых алгарытмаў і лінгвістычных рэсурсаў мабільнага і стацыянарнага СМТ для распрацоўкі тэкставага, прасадыхнага, фанетычнага, алафоннага і акустычнага працэсараў СМТ для Інтэрнэт-платформы.

Зараз даступны карыстальнікам для тэставання акустычны кампанент у Інтэрнэце па спасылцы www.corpus.by/acoustic/form.php.

Выніковы эксперыментальны абразец шматкампанентнага Інтэрнэт СМТ, можа быць выкарыстаны ў якасці завершанага кампанента маўленчага інтэрфейсу любым знешнім распрацоўшчыкам праз Інтэрнэт для пабудовы ўласных інтэлектуальных сістэм. Ён даступны па спасылцы www.corpus.by/tts.

Значым таксама, што рэалізацыя акустычнага кампаненты сінтэзатара маўлення па тэксце на мабільных платформах дазваляе стварыць гібридную архітэктuru сінтэзу маўлення, пры якой кампанент апрацоўкі тэксту размешчаны на серверы ў Інтэрнэце, у той час як кампанент апрацоўкі маўленчага сігналу знаходзіцца на мабільным тэлефоне [Цирульник и др., 2010]. Дадатковай добрай якасцю такой архітэктury з'яўляецца магчымасць сінтэзу маўлення па адным і тым жа размечаным тэксце, які паступае з сервера, з выкарыстаннем розных галасоў і розных

прасадыхных стыляў, змешчаных на мабільным тэлефоне.

Такім чынам, на дадзены момант эксперыментальна распрацаваны ўсе кампаненты сінтэзатара маўлення па тэксце для інтэрнэт-платформ: тэкставы, прасадыхны, фанетычны, алафонны і акустычны. У хуткім часе стане магчымым, каб дадзеныя кампаненты маглі выкарыстоўвацца як паасобку, так і быць інтэграванымі ў любой камбінацыі ў любую інтэлектуальную сістэму. Таксама важна зазначыць, што распрацаваны сінтэзатар маўлення па тэксце для Інтэрнэту на мове праграмавання PHP характарызуецца крос-платформавасцю. Менавіта таму шматлікія сайты-будаўнікі змогуць ужыць гэты сінтэзатар без выкарыстання дадатковых тэхналогій, адрозных ад мовы напісання іх сайтаў.

6. Падзякі

Аўтары выказваюць падзяку Алене Скопінай (skelena777@gmail.com) за філалагічныя праўкі тэкста даклада.

БІБЛІЯГРАФІЧНЫ СПІС

[Лобанов и др., 2008] Лобанов, Б.М. Компьютерный синтез и клонирование речи / Б.М. Лобанов, Л.И. Цирульник // Минск : Белорусская наука, 2008. – 344 с.

[Barker-Plummer, 1995] Barker-Plummer, D Turing Machines [Electronic resource]. – 1995. – Mode of access : <http://plato.stanford.edu/entries/turing-machine/>. – Date of access : 01.07.2012.

[Internet World Stats, 2012a] Top Internet Usage [Electronic resource]. – 2012. – Mode of access : <http://www.internetworldstats.com/top20.htm>. – Date of access : 01.07.2012.

[Internet World Stats, 2012b] Europe Internet Usage Stats Facebook and Population Statistics [Electronic resource]. – 2012. – Mode of access : <http://www.internetworldstats.com/stats4.htm#europe>. – Date of access : 01.07.2012.

[IT.TUT.BY, 2012] В Беларуси более 10 млн абонентов GSM-сетей [Электронный ресурс]. – 2010. – Режим доступа : <http://it.tut.by/202331>. – Дата доступа : 01.07.2012.

[Acapela Group, 2012] Acapela Group [Electronic resource]. – 2012. – Mode of access : <http://www.acapela-group.com>. – Date of access : 16.03.2012.

[Nuance Communications, 2002] Nuance company [Electronic resource]. – 2002. – Mode of access : <http://www.nuance.com>. – Date of access : 12.03.2012.

[Mihai Preda, 2010] How to work around Android's 24 MB memory limit [Electronic resource]. – 2010. – Mode of access : <http://blog.java.org/how-to-work-around-androids-24-mb-memory-limit/>. – Date of access : 01.12.2012.

[Центр Речевых Технологий, 2001] Технологии [Электронный ресурс]. – 2001. – Режим доступа : <http://www.speechpro.ru/technologies>. – Дата доступа : 01.07.2012.

[Dutoit, 1997] Dutoit, T. An Introduction to text-to-speech synthesis / T. Dutoit. – Kluwer Academic Publishers, 1997. – 286 p.

[Гецэвіч і інш., 2011] Гецэвіч, Ю.С. Сінтэзатар беларускага і рускага маўлення па тэксце для стацыянарных і мабільных платформ / Ю.С. Гецэвіч, Д.А. Пакладок // Сб. матэрыялаў I Молодежного инновационного форума «Наука и бизнес – 2011» (Минск, 14–18 нояб. 2011 г.). – Минск : Беларуская навука, 2011. – С. 14–15.

[Lobanov et al., 2006] Lobanov, B. Development of multi-voice and multi-language TTS synthesizer (languages: Belarussian, Polish, Russian) / B. Lobanov, L. Tsurulnik // Speech and Computer: proceedings of the 11-th International conference SPECOM'2006, St. Petersburg, Russia, 25-29 June, 2006 / Institute of Informatics and

Automation of RAS, Speech Informatics Group. – St.-Petersburg: Anatolia, 2006. – P. 274–283.

[Цирульник и др., 2010] Цирульник, Л.И. Система синтеза речи по тексту для мобильных телефонов / Л.И. Цирульник, Д.А. Покладок // Речевые технологии – 2010 – № 1 – С. 81–90.

[Ramachandran, 2010] Sreeram Ramachandran Web metrics: Size and number of resources / Google developers [Electronic resource]. – 2010. – Mode of access : <https://developers.google.com/speed/articles/web-metrics>. – Date of access : 14.07.2012.

[Цирульник и др., 2012] Цирульник, Л.И. Грамматический словарь и правила определения словесного ударения для синтеза речи по тексту на мобильных устройствах / Л.И. Цирульник, Д.А. Покладок // Информатика. – 2012. – № 2. – С. 61–68.

[Гецэвіч і інш., 2012] Гецэвіч, Ю.С. Распрацоўка сінтэзатара беларускага і рускага маўленняў па тэксце для мабільных і інтэрнэт-платформаў / Ю.С. Гецэвіч, Д.А. Пакладок, Д.В. Брэк // Развитие информатизации и государственной системы научнотехнической информации (РИНТИ-2012) : доклады XI Международной конференции (Минск, 15 ноября 2012 г.). – Минск : ОИПИ НАН Беларуси, 2012. – С. 254–259.

COMPONENTS FOR DIFFERENT PLATFORMS OF TEXT-TO-SPEECH SYNTHESIZERS FOR INTELLECTUAL SYSTEMS

Hetsevich Y.S.*, Pakladok D.A.*, Brek D.V.**

** United Institute of Informatics Problems,
National Academy of Sciences,
Minsk, Republic of Belarus*

yuras.hetsevich@newman.bas-net.by
dima.pokladok@gmail.com

*** The Belarusian State University of Informatics
and Radioelectronics,
Minsk, Republic of Belarus*

bartwader@gmail.com

Questions concerning urgency and practical construction of components for the bilingual (Belarusian and Russian) text-to-speech synthesizer for mobile and internet platforms are discussed here. According to the test results the speech synthesizer for the Internet can completely meet the demands for voicing of website pages. It can also be used as a complete component of a speech interface by any external developer by means of the Internet in order to build their own intelligent systems.

INTRODUCTION

Nowadays there is a problem of absence of cross-platform text-to-speech synthesizers which would be able to voice texts from Internet resources. The solution to this problem will help Internet websites to attain a higher level of interactivity. By means of such a speech synthesizer it is possible to expand audience of Internet sites. Blind or visually impaired people will have a possibility to familiarize themselves with the contents of websites. One can do something they need or like, for example, tide up in house, drive a car, and still listen to private messages, articles, news, etc. from the Internet.

The Internet TTS developers will be able to demonstrate the synthesizer in different modes of use

(full or limited access) to a large number of users and target groups (for research works, pupils and students, entrepreneurs and potential clients who want to construct commercial services of voicing texts).

MAIN PART

In the beginning experimental models were built. They confirmed the ability to work with audio files in WAV-format through an interpreted programming language PHP. After that an experimental programming model of basic algorithms of the acoustic processor for the text-to-speech synthesizer in PHP was obtained. The bilingual acoustic processor is available online at the link www.corpus.by/acoustic/form.php. By means of its interface a user can enter a phonetic text (or use already entered), identify a language and process an query to the server. The software script of the acoustic processor receives the query, processes it, and in the end generates a WAV-file. The resulting file can be listened or saved to one's local computer.

Further, all the other parts of the speech synthesizer were developed for the Internet. As a result, any written text consistently passes all stages of processing through text, prosodic, phonetic, allophonic and acoustic processors. On the output a WAV-file with a synthesized speech is created. The result of each stage is visually presented on the Internet page (www.corpus.by/tts).

CONCLUSION

Thus, at present all the components (text, prosodic, phonetic, allophonic and acoustic) of the text-to-speech synthesizer for online platforms have been experimentally developed. Soon it will become possible to use these components separately and to integrate them in any combinations into any intelligent system.

Another result of the work on development of the text-to-speech synthesizer for the Internet has been the emergence of the Internet service, which shows all the stages of converting a written text into an audio file. By means of this service it is possible to improve the quality of speech synthesis by finding and correcting errors at any stage of the process. Obviously, further work will enable to voice web-pages, articles, news and emails at a high level.



УДК 004.822:514

О СИНТЕЗАТОРЕ ТАДЖИКСКОЙ РЕЧИ ПО ТЕКСТУ

Худойбердиев Х.А.*

* *Худжандский политехнический институт Таджикского технического Университета,
г. Худжанд, Республика Таджикистан*

tajlingvo@gmail.com

В настоящей работе излагается основная идея реализации синтезатора таджикской речи по тексту.

Ключевые слова: синтез речи; обработка текста; структура текстовой информации; синтезатор.

1. Напомним ряд понятий, используемых в дальнейшем.

Текст – это последовательность предложений, построенных согласно правилам данного языка и данной знаковой системы и образующая сообщение, [1].

В свою очередь, *предложение* будем рассматривать как совокупность упорядоченных элементов 7 типов, называемых *словом, числом, символом, пробелом, внутренним знаком препинания* (запятая, двоеточие, точка с запятой, тире), *внешним знаком препинания* (точка, многоточие, знак вопросительный, знак восклицательный) и, наконец, *служебным символом окончания абзаца* (в письменном тексте его нет, но он присутствует в электронном тексте как непечатаемый знак ¶).

Отметим, что смысл, который мы заключаем в названия элементов, следует понимать в общепринятых значениях. Подчеркнем также, что в конкретном предложении некоторые элементы могут отсутствовать (например, числа, символы, внутренние знаки препинания и т.д.) в то время как присутствие других - обязательно (например, внешний знак препинания).

Нам понадобятся 5 типов пауз, используемых в речи:

P_s - пауза между слогами при произношении слова;

P_w - пауза между словами при чтении предложения (соответствует пробелу между словами);

P_i - пауза, отмечающая внутренний знак препинания;

P_e - пауза, отмечающая внешний знак препинания;

P_a - пауза, отмечающая конец абзаца.

2. Теперь опишем идею синтеза речи по тексту в виде принципиальной блок-схемы, (см. рисунок 1).

Работа синтезатора происходит следующим образом. После ввода очередного предложения оно анализируется по составу своих элементов. Если очередной элемент – слово, то в блоке 1 оно разделяется на слоги с указанием ударного слога и затем осуществляется его озвучивание с использованием базы “слог-звук”.

Если очередной элемент – число, то оно в блоке 2 преобразуется в текст и затем его озвучивание происходит через блок 1.

Если очередной элемент – символ, то его озвучивание происходит в блоке 3 путем извлечения соответствующего звучания из базы “символ-звук”.

Если очередной элемент – пробел, внутренний или внешний знак препинания или же знак окончания абзаца, то для них из соответствующего блока извлекается соответствующая пауза.

Синтезатор речи, представленный в виде блок-схемы, подсказывает, что в его основу закладывается принцип конкатенации озвученных слогов.

Поскольку слог выступает в качестве основной звуковой единицы речи, то для реализации синтезатора *требуется описать многообразие все слогов соответствующего естественного языка*, что и сделано в [6].

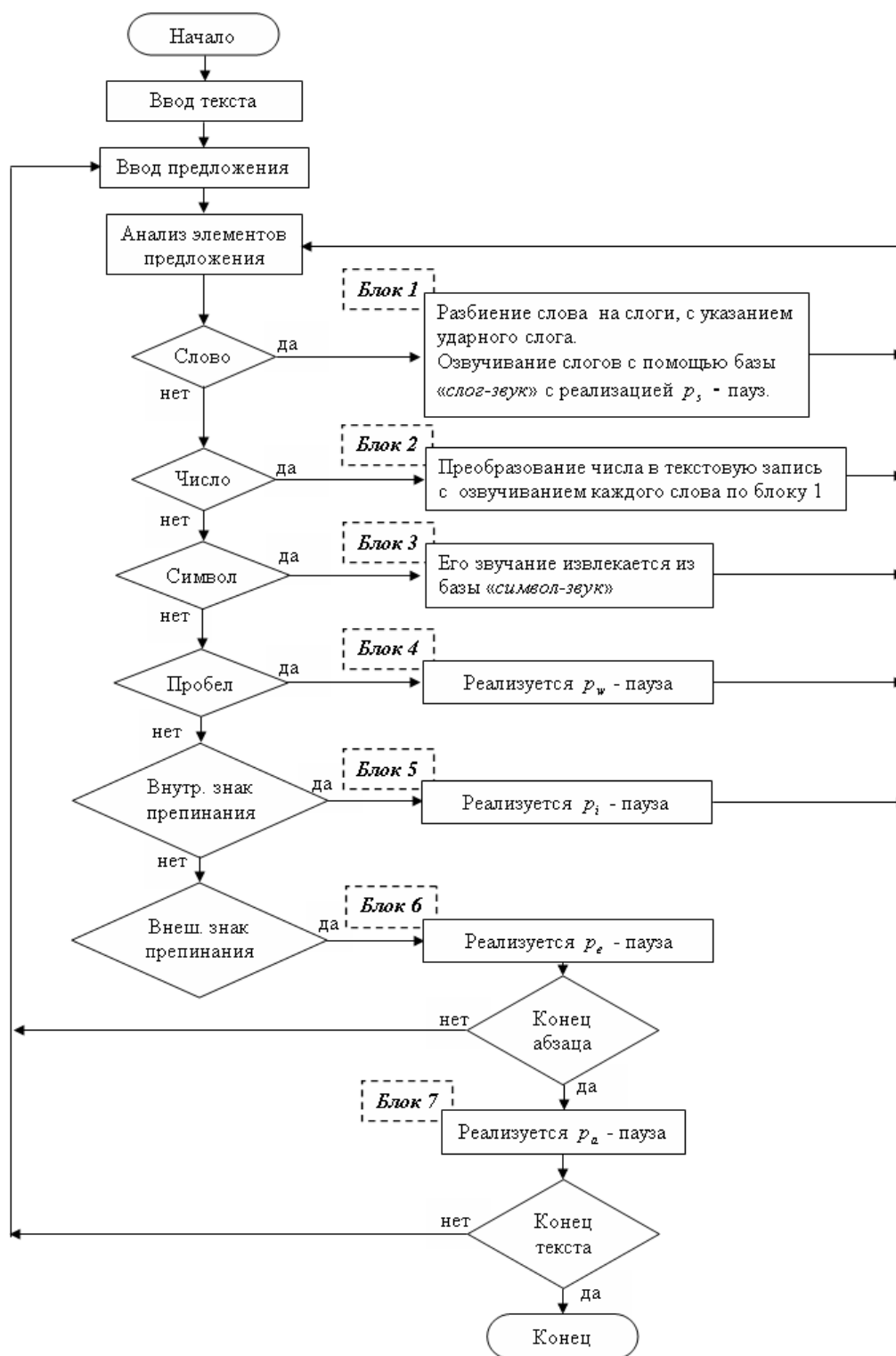


Рисунок 1 - Принципиальная блок-схема синтеза речи по тексту

Поскольку каждый слог, представленный в виде цепочки букв, нуждается в его звуковом образе, то *требуется создание базы “слог-звук”* что и сделано в [3].

Поскольку синтезатор предусматривает озвучивание чисел и символов, то для реализации синтезатора разработаны соответствующие алгоритмы и программы для трансформирования

числа в текст и создания базы “символ-звук”.

И, наконец, требуется настроить длительности пауз p_s , p_w , p_i , p_e и p_a таким образом, чтобы получить, по-возможности, естественную и разборчивую синтетическую речь.

3. На основе детальных проработок

концептуальной схемы был создан синтезатор Tajik Text-to-Speech, т.е. комплекс программ для синтеза таджикской речи по тексту. Настройка параметров синтезатора проводилась путем вычислительных экспериментов. Установлено удовлетворительные значения длительностей пауз:

- для границ абзацев $P_a = 900$ мс;
- для границ предложений $P_e = 600$ мс;
- для запятых $P_i = 400$ мс;
- для межсловной и межслоговой пауз,

соответственно, $p_w = 200$ мс и $p_s = 20$ мс.

Структурная схема программного комплекса представлена на рисунке 2.

Блок 1. Подсистема «Интерфейс пользователя» состоит из двух компонентов - «Ввод текста» и «Озвученная речь», которые имеют одностороннюю связь, т.е. пользователь имеет возможность ввести текстовую информацию и в результате получить речевой вариант вводимого текста. Для получения результатов блок 1 связывается с блоком 2 по двум направлениям - по предоставлению информации для лингвистического анализа и получению результатов озвучивания. Блок 1 взаимодействует также и с блоком 3 непосредственно для использования необходимых данных о настройках системы (выбор мужского или женского голоса, громкость и скорость озвучивания).

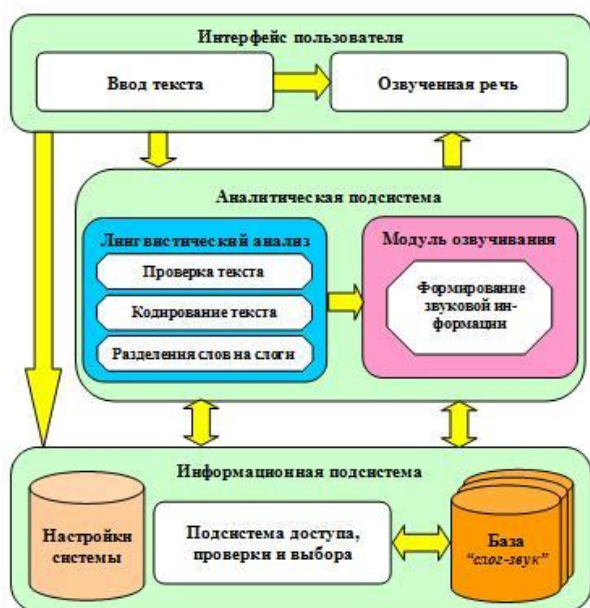


Рисунок 2 - Структурная схема Tajik Text-to-Speech

Блок 2. Аналитическая подсистема состоит из двух частей - «Лингвистического анализа» и «Модуля озвучивания». Первый из них состоит из подмодулей «Проверка текста», «Кодирование текста» и «Разделение слов на слоги». «Проверка текста» используется для проверки вводимой информации, которая включает в себя такие текстовые элементы, как слова, целые числа, символы и знаки препинания. Данный подмодуль

проверяет текстовые элементы, преобразует в тестовый вариант целые числа и символы и затем передает их для кодирования.

Процесс кодирования осуществляет одноименный подмодуль, который каждое слово W входного текста преобразует в упорядоченную совокупность $W_{0,1}^*$ нулей и единиц (напомним, что цифрой 1 замещаются гласные буквы, а цифрой 0 – согласные), т.е. все слова представляются своими слоговыми структурами.

Закодированный текст передается подмодулю «Разделение слов на слоги», который работает на основе собственного алгоритма. Разделенные на слоги слова лингвистически анализируются и передаются «Модулю озвучивания».

В указанном модуле происходит формирование звуковой информации с использованием базы «слог-звук» информационной подсистемы, ударных слогов, межсловных и межсловных пауз, а также пауз, отмечающих такие знаки препинания как запятая и точка.

Модуль озвучивания является заключительной стадией аналитической подсистемы, и звуковой вариант текстовой информации отправляется в интерфейс пользователя.

Блок 3, «Информационная подсистема» содержит базы данных, названные «Настройкой системы» и «Базой слог-звук». Первая из них используется для хранения временных данных настройки системы, вторая база «слог-звук» - для хранения статистических данных о звуковых файлах 3259 слогов таджикского языка. Для работы с этой базой данных используется модуль обеспечения доступа, проверки и выборки необходимых данных.

Для оценки эффективности работы синтезатора были организованы эксперименты по озвучиванию разнообразной текстовой информации (фрагменты из повестей, романов, научных статей, учебников, газет, журналов, Интернет - сайтов). Оценка полноты множества слогов, использованных для формирования синтетической речи, связывалась с процентом озвученных слов по отношению к общему количеству слов в пределах выбранных фрагментов текста. Результаты эксперимента показаны в таблице.

В приведенной таблице во 2-м столбце приводятся названия файлов, в 3-м столбце – источник, из которого взята информация и в 6-ом – доля озвученных слов в процентах по отношению к общему числу слов.

Таблица 1. Коэффициент озвучивания текста для фрагментов текстовой информации

№	Название файла	Источник	%
1	Гимн.rtf	Учебник	100,00
2	С_Айни1.rtf	«Ятим», С.Айни	99,47
3	Вестник32.rtf	Газета «Вароруд»	98,63
4	Сино.rtf	Реферат	98,88
5	Ду дилдода.rtf	Интернет сайт www.gazeta.tj	98,10
6	Президент.rtf	Газета «Чархи гардун»	96,69
Итого			98,45

Полученные результаты показали вполне удовлетворительное качество работы комплекса программ Tajik Text-to-Speech по озвучиванию таджикского текста.

Таким образом, комплекс программ Tajik Text-to-Speech, хотя и не решает полностью поставленной проблемы, *все же в настоящее время оказывается первым программным продуктом, хорошо осуществляющим компьютерное озвучивание таджикских текстов.* На данном уровне разработки комплекс уже сейчас может быть использован людьми с ослабленным зрением.

Другие эксперименты проводились на научных семинарах Института математики АН РТ. Его участники по собственному усмотрению вводили в компьютер таджикские тексты и затем оценивали естественность и разборчивость звучания синтетической речи. Общее мнение семинара - *компьютерный синтезатор, построенный по принципу конкатенации 3259 таджикских слогов, вполне успешно выполняет функцию озвучивания таджикских текстов.*

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- Советский энциклопедический словарь. -М.: Советская энциклопедия, 1980. 1600 с.
- Л.В.Златоустова, С.В.Кодзасов, О.Ф.Кривнова, И.Г.Фролова. Алгоритмы преобразования русских орфографических текстов в фонетическую запись. -М.: МГУ, 1970.
- Лобанов Б.М. Цирульник Л.И. Жадинец Д.В. Сизонов О.Г. Алгоритмы синтеза просодических характеристик речи по тексту в системе «Мультифон». Объединённый институт проблем информатики НАН Беларуси, Минск. 2007.
- Лобанов Б.М., Карневская Е.Б., Левковская Т.В. Синтезатор речи по тексту как компьютерное средство «клонирования» персонального голоса // Тр. Международной конференции Диалог-2001 / -М.:, 2001. С. 265-272.
- Лобанов Б.М., Цирульник М.И. Компьютерный синтез и клонирование речи, Минск, Белорусская наука, 2008, - 316 с.
- Тим Кинтцель. Программирование звука на ПК. Пер. с англ. -М.: ДМК Пресс, 2005, 432 с.
- Баландюк Ю. В. Просодическое устройство слова в корейском языке (на материале двуслога). Автореферат диссертации на соискание ученой степени кандидата филологических наук, Институт стран Азии и Африки при МГУ им. М.В.Ломоносова, -М.: – 2003, 23 с.
- Худойбердиев Х.А., О многообразии слогов таджикского языка // Известия АН РТ, № 2 (127), 2007. - с. 31-34.
- Усманов З.Д., Худойбердиев Х.А. Алгоритм безударного озвучивания таджикского текста // ДАН РТ, Т.50, № 4, 2007. - с. 302-305.
- Усманов З.Д., Худойбердиев Х.А. Компьютерное озвучивание таджикского текста // Патент (интеллектуальный продукт) зарегистрирован 041ТJ 04.09.2007 НПИ центром.

ON A TAJIK SPEECH SYNTHESIZER UNDER A TEXT

Kh. A. Khudoyberdiev*

Khujand Polytechnic Institute
of Tajik Technical University
tajlingvo@gmail.com

In the article a block-scheme of the program complex for realization of a synthesis of Tajik speech under a text is described. Values of parameters providing a satisfactory work of a synthesizer are specified.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

AN ANN-BASED METHOD FOR VOCAL FOLD PATHOLOGY DIAGNOSIS

Vahid Majidnezhad *, Igor Kheidorov *

* *United Institute of Informatics Problems, National Academy of Sciences, Minsk, Belarus*

vahidmn@yahoo.com

ikheidorov@sakrament.com

Abstract: There are different algorithms for vocal fold pathology diagnosis. These algorithms usually have three stages which are Feature Extraction, Feature Reduction and Classification. While the third stage implies a choice of a variety of machine learning methods, the first and second stages play a critical role in performance and accuracy of the classification system. In this paper we present initial study of feature extraction and feature reduction in the task of vocal fold pathology diagnosis. A new type of feature vector, based on wavelet packet decomposition and Mel-Frequency-Cepstral-Coefficients (MFCCs), is proposed. Also Principal-Component Analysis (PCA) is used for feature reduction. An Artificial Neural Network is used as a classifier for evaluating the performance of our proposed method.

Keywords: Wavelet Packet Decomposition, Mel-Frequency-Cepstral-Coefficient (MFCC), Principal-Component Analysis (PCA), Artificial Neural Network (ANN).

INTRODUCTION

Speech signal information often plays an important role for specialists to understand the process of vocal fold pathology formation. In some cases vocal signal analysis can be the only way to analyze the state of vocal folds. Nowadays diverse medical techniques exist for direct examination and detection of pathologies. Laryngoscopy and electromyography are most frequently used by medical specialists. But these methods possess a number of disadvantages. Human vocal tract is hardly-accessible for visual examination during phonation process and that makes it more problematic to identify a pathology. Moreover, these diagnostic means may cause the patients feel much discomfort and distort the actual signal so that it may lead to incorrect diagnosis as well [Alonso et al., 2001] [Ceballos et al., 2005] [Ceballos et al., 1996] [Adnene et al., 2003]. Acoustic analysis as a diagnostic method has no drawbacks, peculiar to the above mentioned methods. It is a non-invasive diagnostic technique that allows pathologists to examine many people in short time period with minimal discomfort. The general scheme of vocal fold pathology diagnostic methods consists of three stages which are feature extraction, feature reduction and classification.

Different parameters for feature extraction are used. Traditionally, one deals with such parameters like pitch, jitter, shimmer, amplitude perturbation, pitch perturbation, signal to noise ratio, normalized noise energy [Manfredi, 2000] and others [Llorente et al., 2004] [Rosa et al., 2000] [Mallat, 1989] [Wallen et al., 1996]. Feature extraction, using the above mentioned

parameters, has shown its efficiency for a number of practical tasks. In the proposed method, we have used the Mel-Frequency-Cepstral-Coefficients (MFCCs), Energy and Shannon Entropy parameters for creating the features vector. Also different approaches for feature reduction are used such as Principal Component Analysis (PCA) [Chen et al., 2007] [Go´mez et al., 2005] [Michaelis et al., 1998] [Marinaki et al., 2004] and Linear Discriminant Analysis (LDA) [Ji-Yeoun et al., 2007]. In the proposed method we have used PCA for feature reduction. Finally, the reduced features are used for speech classification into the healthy and pathological class. Different machine learning methods such as Support Vector Machines [Chen et al., 2007], Artificial Neural Networks [Ritchings et al., 2002], etc can be used as a classifier. In the proposed method we have used the ANN for the classification purpose.

1. Methodology

The wavelet transform, as was shown in [Manfredi, 2000], is a flexible tool for analysis of speech signals. This led us to supposition that feature vectors based on wavelets can show good results. The idea to build feature vector on wavelets for audio classification was previously reported by [Li et al., 2003] and [Tzanetakis et al., 2002]. These authors used the discrete wavelet transform (DWT) coefficients for their method of feature extraction for content-based audio classification. [Kukharchik et al., 2007] used continues wavelet transform (CWT) coefficients for their method of feature extraction. [Cavalcanti et al., 2010] used Wavelet Packet Decomposition (WPD) nodes coefficients for their method for feature extraction.

The block diagram of our proposed method is illustrated in Fig. 1. In the first stage, by the use of MFCC and Wavelet Packet Decomposition, feature vector containing 1035 features is made. In the second stage, by the use of PCA method, the dimension of feature vector is reduced. In the last stage, by the use of Artificial Neural Network (ANN), the speech signal classified into two classes: pathological or healthy.

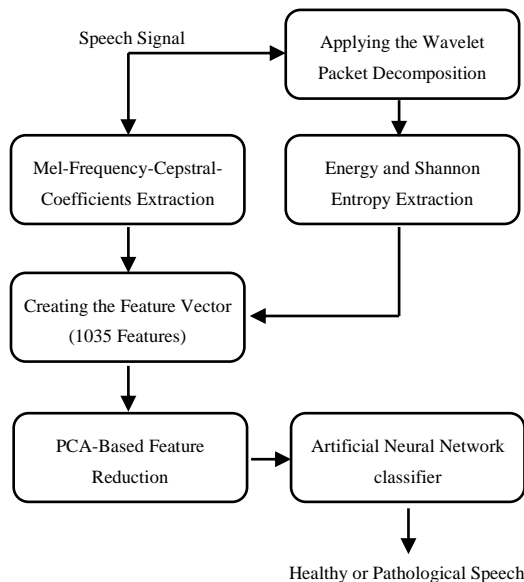


Figure 1- The scheme of the proposed method for detection of vocal fold pathology.

1.1. Feature extraction

As it is shown in Fig. 1, first, by the use of cepstral representation of input signal, 13 Mel-Frequency-Cepstral-Coefficients (MFCC) are extracted. Then the wavelet packet decomposition in 5 levels is applied on the input signal to make the wavelet packet tree. Then, from the nodes of resulting wavelet packet tree, 511 energy features along with 511 Shannon entropy features are extracted. Finally, by the combination of these features, the initial feature vector with the length of 1035 features is created.

1.1.1. Mel-Frequency-Cepstral-Coefficients (MFCCs)

MFCCs are widely used features to characterize a voice signal and can be estimated by using a parametric approach derived from linear prediction coefficients (LPC), or by the non-parametric discrete fast Fourier transform (FFT), which typically encodes more information than the LPC method. The signal is windowed with a hamming window in the time domain and converted into the frequency domain by FFT, which gives the magnitude of the FFT. Then the FFT data is converted into filter bank outputs and the cosine transform is found to reduce dimensionality. The filter bank is constructed using 13 linearly-spaced filters (133.33Hz between center frequencies,) followed by 27 log-spaced filters (separated by a factor of 1.0711703 in frequency.) Each filter is constructed by combining the amplitude of FFT bin. The Matlab code to calculate the MFCC features was adapted from the Auditory Toolbox (Malcolm Slaney). The MFCCs are used as features in

[Ji-Yeoun et al., 2007] to classify the speech into pathological and healthy class.

1.1.2. Wavelet Packet Decomposition

Recently, wavelet packets (WPs) have been widely used by many researchers to analyze voice and speech signals. There are many out-standing properties of wavelet packets which encourage researchers to employ them in widespread fields. The most important, multi resolution property of WPs is helpful in voice signal synthesis [Herisa et al., 2009] [Fonseca et al., 2007].

The hierarchical WP transform uses a family of wavelet functions and their associated scaling functions to decompose the original signal into subsequent sub-bands. The decomposition process is recursively applied to both the low and high frequency sub-bands to generate the next level of the hierarchy. In this study, mother wavelet function of the tenth order Daubechies has been chosen and the signals have been decomposed to 8 levels. The mother wavelet used in this study is reported to be effective in voice signal analysis [Guido et al., 2005] [Umapathy et al., 2005] and is being widely used in many pathological voice analyses [Fonseca et al., 2007].

1.2. Feature Reduction

Using every feature for classification process is not good idea and it may be causes to the increasing the rate of misclassification. Therefore, it is better to choose the proper features from the whole features. This process is called as "Feature Reduction". One way for feature reduction is Principal Component Analysis (PCA) which is used frequently in pervious works such as [Chen et al., 2007] [Go´mez et al., 2005] [Michaelis et al., 1998] [Marinaki et al., 2004].

PCA method searches a mapping to find the best representation for distribution of data. Therefore, it uses a signal-representation criterion to perform dimension reduction while preserving much of the randomness or variance in the high-dimensional space as possible. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. PCA involves the calculation of the eigenvalues decomposition of a data covariance matrix or singular value decomposition of a data matrix, usually after mean centering the data for each attribute. PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate, called the first principal component, the second greatest variance on the second coordinate, and so on.

1.3. Artificial Neural Network

An artificial neural network (ANN) as a computing system is made up of a number of simple, and highly interconnected processing elements, which processes information by its dynamic state response to external inputs. Neural networks are better suited for achieving human-like performance in the fields such as speech

processing, image recognition, etc. Processing elements in an ANN are also known as neurons. These neurons are interconnected by means of information channels called interconnections. Each neuron can have multiple inputs; while there can be only one output. Inputs to a neuron could be from external stimuli or could be from output of the other neurons. Copies of the single output that comes from a neuron could be input to many other neurons in the network. When the weighted sum of the inputs to the neuron exceeds a certain threshold, the neuron is fired and an output signal is produced. The network can recognize input patterns once the weights are adjusted or tuned via some kind of learning process [Lee et al., 1992].

2. Experiments and Results

The database was created by specialists from the Belarusian Republican Center of Speech, Voice and Hearing Pathologies. We have selected 75 pathological speeches and 55 healthy speeches which are related to the vowel “a”. All the records are wave files in PCM format. The whole scheme of our proposed method is illustrated in Fig. 1. We have adopted a 10 fold cross-validation scheme to assess the generalization capabilities of the system in our experiments.

It is necessary to know how many neurons in the hidden layer are required to achieve the optimal results. So, in the first experiment the numbers of neurons in the hidden layer are investigated and the results are shown in Fig. 2. As it is obvious, using 21 neurons in the hidden layer lead to the optimal case which is 74.62% of accuracy. Finally, according to the result of the first experiment, an ANN with 21 neurons is created for using in the second experiment. The goal of the second experiment is to reduce the initial feature vector’s length so that the accuracy of the classification phase increases. For this purpose, the conventional PCA method is used and the results are shown in Fig. 3. As it is obvious, using the 51 features lead to the best result which is 89.23% of accuracy. The 51 selected features, by the means of PCA method, are shown in the table 1.

Table 1- The selected features for the construction of feature vector.

Feature Reduction Method	The selected features	Accuracy
PCA (feature vector length=51)	The 1-3, 5-6 and 10 th coefficients of MFCC. Energy at the 19, 32-33, 35, 65-66, 70-71, 129, 131, 135, 140, 256-258, 260, 262-263, 265-266, 268, 271 and 274 th nodes of WP Tree. Entropy at the 8, 16-17, 32-33, 64, 66, 69, 128-129, 132, 134-135, 256-259, 261, 264, 268, 270 and 280 th nodes of WP Tree.	89.23%
None (feature vector length=1035)	All the 511 entropy & 511 energy & 13 MFCC coefficients.	74.62%

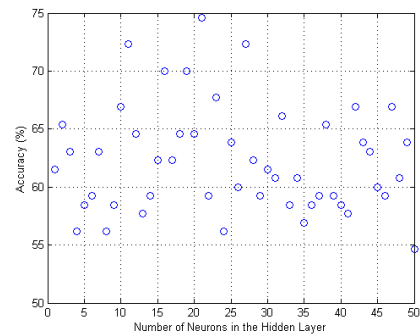


Figure 2 - The Classification Results based on the number of Neurons

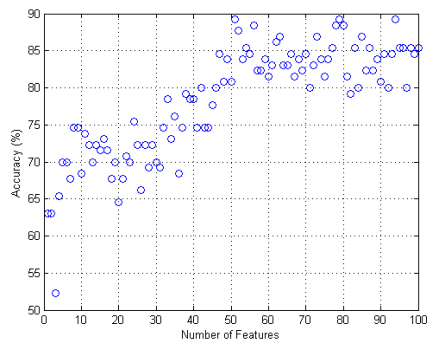


Figure 3 - The Classification Results based on the number of Features

3. Conclusion

Acoustic analysis is a proper method in vocal fold pathology diagnosis so that it can complement and in some cases replace the other invasive, based on direct vocal fold observation, methods. In this article, an ANN-Based method for vocal fold pathology diagnosis is proposed so that in the proposed scheme, Mel-Frequency-Cepstral-Coefficients along with the wavelet packet decomposition are used for feature extraction phase. Also PCA method for the feature reduction phase is used. And finally the ANN is used for the classification phase. Two experiments are designed to investigate the optimal case for the numbers of neurons in the hidden layer of the ANN and also the optimal case for the feature vector length as the input of ANN.

REFERENCES

- [Alonso et al., 2001] Alonso, J.B. Automatic Detection of Pathologies in the Voice by HOS Based Parameters / J.B. Alonso, J.D. Leon, I. Alonso, M.A. Ferrer // EURASIP Journal on Applied Signal Processing, - 2001 - No. 4 – pp. 275-284.
- [Ceballos et al., 2005] Ceballos, L.G. A Non-Linear Based Speech Feature Analysis Method with Application to Vocal Fold Pathology Assessment. / L.G. Ceballos, J. Hansen, J. Kaiser // IEEE Trans. Biomedical Engineering. – 2005 – No. 45(3) – pp. 300-313.
- [Ceballos et al., 1996] Ceballos, L.G. Vocal Fold Pathology Assessment Using AM Autocorrelation Analysis of the Teager Energy Operator/ L.G. Ceballos, J. Hansen, J. Kaiser // ICSLP-1996 Proc. – 1996 - pp: 757-760.
- [Adnene et al., 2003] Adnene, C. Analysis of Pathological Voices by Speech Processing. / C. Adnene, B. Lamia // Signal Processing and Its Applications. - 2003 – No. 1(1) – pp. 365-367.
- [Manfredi, 2000] Manfredi, C. Adaptive Noise Energy Estimation in Pathological Speech Signals. / C. Manfredi // IEEE Trans. Biomedical Engineering. –2000 – No. 47(11) – pp. 1538-1543.
- [Llorente et al., 2004] Llorente, J.I.G. Automatic Detection of Voice Impairments by Means of Short-Term Cepstral Parameters and

Neural Network Based Detectors/ J.I.G. Llorente, P.G. Vilda // IEEE Trans. Biomedical Engineering. – 2004 – No. 51(2) – pp. 380-384.

[Rosa et al., 2000] Rosa, M.D.O. Adaptive Estimation of Residue Signal for Voice Pathology Diagnosis. / M.D.O Rosa, J.C. Pereira, M. Grellet // IEEE Trans. Biomedical Engineering – 2000 – No. 47(1) – pp. 96-104.

[Mallat, 1989] Mallat, S.G. A Theory for Multi-resolution Signal Decomposition: the Wavelet Representation. / S.G. Mallat // IEEE Trans Pattern Analysis and Machine Intelligence. – 1989 – No. 11(7) – pp. 674-693.

[Wallen et al., 1996] Wallen, E.J. A Screening Test for Speech Pathology Assessment Using Objective Quality Measures. / E.J. Wallen, J.H. Hansen // ICSLP Proc. – 1996 – pp. 776-779.

[Chen et al., 2007] Chen, W. SVM-based identification of pathological voices. / W. Chen, C. Peng, X. Zhu, B. Wan, D. Wei // Proceedings of the 29th Annual International Conference of the IEEE EMBS. – 2007 – pp. 3786-3789.

[Go´mez et al., 2005] Go´mez, P. Principal component analysis of spectral perturbation parameters for voice pathology detection. / P. Go´mez, F. Dr´az, A. A´lvarez, K. Murphy, C. Lazaro, R. Martinez, V. Rodellar // Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems. – 2005 - pp: 41-46.

[Michaelis et al., 1998] Michaelis, D. Selection and combination of acoustic features for the description of pathologic voices. / D. Michaelis, M. Frohlich, H.W. Strube // Journal of the Acoustical Society of America. – 1998 – No. 103(3) – pp. 1628-1639.

[Marinaki et al., 2004] Marinaki, M. Automatic detection of vocal fold paralysis and edema. / M. Marinaki, C. Kotropoulos, I. Pitas, N. Maglaveras // Proceedings of Eighth International Conference on Spoken Language Processing-ICSLP. – 2004.

[Ji-Yeoun et al., 2007] Ji-Yeoun, L. Classification of pathological and normal voice based on linear Discriminant analysis. / L. Ji-Yeoun, J. SangBae, H. Minsoo // Proceedings of the Eighth International Conference on Adaptive and Natural Computing Algorithms. – 2007 - pp. 382-390.

[Ritchings et al., 2002] Ritchings, R.T. Pathological voice quality assessment using artificial neural networks. / R.T. Ritchings, M.A. McGillion, C.J. Moore // Medical Engineering & Physics. – 2002 – No. 24(8) – pp. 561-564.

[Li et al., 2003] Li, T. A comparative study on content based music genre classification. / T. Li, M. Oginara, Q. Li // Proc. Of the 26th annual int.ACM SIGIR conf. on Research and development in information retrieval. – 2003 – pp. 282-289.

[Tzanetakis et al., 2002] Tzanetakis, G. Musical genre classification of audio signals. / G. Tzanetakis, P. Cook // IEEE Trans. on Speech and Audio Processing. – 2002 – No. 10(5) – pp. 293-302.

[Kukharchik et al., 2007] Kukharchik, P. Vocal fold pathology detection using modified wavelet-like features and support vector machines. / P. Kukharchik, D. Martynov, I. Kheidorov, O. Kotov // 15th European Signal Processing Conference (EUSIPCO). – 2007 – pp. 2214-2218.

[Cavalcanti et al., 2010] Cavalcanti, N. Comparative Analysis between Wavelets for the Identification of Pathological Voices. / N. Cavalcanti, S. Silva, A. Bresolin, H. Bezerra, A. Guerreiro // Proceedings of the 15th Iberoamerican congress conference on Progress in pattern recognition, image analysis, computer vision, and applications. – 2010 – pp. 236-243.

[Herisa et al., 2009] Herisa, H.K. Optimal feature selection for the assessment of vocal fold disorders. / H.K. Herisa, B.S. Aghazadeh and M.N. Bahrami // Computers in Biology and Medicine. – 2009 – No. 39(10) – pp. 860-868.

[Fonseca et al., 2007] Fonseca, E. Wavelet time frequency analysis and least squares support vector machines for identification of voice disorders. / E. Fonseca, R.C. Guido, J.C. Pereira, P.R. Scalassarsa, C.D. Maciel, J.C. Pereira // Computers in Biology and Medicine. – 2007 – No. 37(4) – pp. 571-578.

[Guido et al., 2005] Guido, R.C. Trying different wavelets on the search for voice disorders sorting. / R.C. Guido, J.C. Pereira, E. Fonseca, F.L. Sanchez, L.S. Vierira // 37th IEEE International Southeastern Symposium on System Theory. – 2005 – pp. 495-499.

[Umapathy et al., 2005] Umapathy, K. Feature analysis of pathological speech signals using local discriminant bases technique. / K. Umapathy, S. Krishnan // Medical and Biological Engineering and Computing. – 2005 – No. 43(4) – pp. 457-464.

[Lee et al., 1992] Lee, K.Y. SHORT-TERM LOAD FORECASTING USING AN ARTIFICIAL NEURAL NETWORK. / K.Y. Lee, Y.T. Cha, J.H. Park // Transactions on Power Systems. – 1992 – No. 7(1) – pp. 124-132.

ДИАГНОСТИКА ПАТОЛОГИИ ГОЛОСОВОГО ТРАКТА НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ

Вахид Маджиднежад *, Игорь Хейдоров *

* *Объединенный институт проблем информатики Национальной академии наук Беларуси (ОИПИ), Минск, Беларусь*

vahidmn@yahoo.com

ikheidorov@sakrament.com

В этой статье представляется метод искусственных нейронных сетей для решения задач диагностики патологии голосового тракта.

ВВЕДЕНИЕ

В настоящее время существуют разнообразные медицинские методы для непосредственного изучения и выявления патологий. Однако эти методы обладают рядом недостатков. Голосовой тракт человека труднодоступен для визуального осмотра, что не позволяет использовать его для выявления патологии. Кроме того, различные приборы для диагностики могут вызвать у пациента значительный дискомфорт, что приводит к искажению реального сигнала и в конечном счете неправильному диагнозу. Предлагаемый в данной статье акустический анализ является неинвазивным методом диагностики и не имеет недостатков, свойственных указанным выше способам.

ОСНОВНАЯ ЧАСТЬ

Для решения задачи диагностики патологии голосовых связок была разработана искусственная нейронная сеть. Для нахождения оптимального числа нейронов в скрытом слое ИНС и длины вектора признаков, подаваемого на вход ИНС, было проведено два эксперимента. В соответствии с результатами первого эксперимента была построена ИНС с 21 нейронами, которая была использована в последующих экспериментах. Цель второго эксперимента заключается в сокращении размерности вектора признаков таким образом, чтобы улучшить точность классификации. Для этого был использован метод главных компонент. Результаты эксперимента показали, что использование для классификации 51 признака приводит к наилучшим результатам (точность 89.23%).

ЗАКЛЮЧЕНИЕ

В данной работе был предложен новый тип вектора признаков на основе вейвлет-разложения пакетов и мел-частотных кепстральных коэффициентов (MFCCs). Для уменьшения размерности вектора признаков был использован метод главных компонент, а в качестве классификатора была использована искусственная нейронная сеть.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОБОБЩЕННАЯ МОДЕЛЬ ЭМОЦИИ И ПРОЦЕССА АНАЛИЗА РЕЧЕВОГО СИГНАЛА В СИСТЕМЕ БЕЗОПАСНОСТИ

Пекарь Д. В. *, Садов В. С. *

**Белорусский Государственный Университет*

Минск, Беларусь

pekar.dima@gmail.com

sadov@bsu.com

Эмоции являются естественным проявлением работы психо-физиологической деятельности нервной системы человека, а речь одним из наиболее естественных способов их проявления, что представляет большой интерес для своего изучения и использования в речевых системах. Эффективная автоматическая обработка речевых сигналов предполагает под собой строгую формализацию предмета анализа, которая может поддаваться не только качественной оценке, но и количественной параметризации.

Ключевые слова: модель, эмоциональная окраска, семантический анализ, обработка речи.

ВВЕДЕНИЕ

Эмоции являются неотъемлемой частью жизнедеятельности человека, которые формируются вследствие психо-физиологической деятельности его нервной системы. Принимая данный факт во внимание, можно положить, что эмоции имеют непосредственную связь с его внутренним состоянием. Согласно существующим теориям, эмоции имеют сложное происхождение, а именно "эмоции – это психофизиологический феномен, поэтому о возникновении переживания человека можно судить как по самоотчету человека о переживаемом им состоянии, так и по характеру изменения вегетативных показателей (частоте сердечных сокращений, артериальному давлению, частоте дыхания и т. д.) и психомоторики: мимике, пантомимике (позе), двигательным реакциям, голосу" [1]. Таким образом, речь является одним из наиболее естественных способов проявления эмоций и отражением эмоционального состояния человека. Анализ эмоциональной окраски речевого сигнала имеет огромное значение для систем распознавания и синтеза речи [2-3], call-центрах [4], диалоговых системах [5-6], системах безопасности, который подразумевает использование некой формализованной модели эмоции.

1. Формализованная модель эмоции в системе анализа эмоциональной окраски речи

Для построения эффективной системы анализа эмоциональной окраски речевого сигнала

необходимо формализовать понятие эмоции – определить ее модель как математический объект, который обладает определенными параметрами и свойствами. Подобный подход позволяет реализовать в дальнейшем процесс анализа эмоционального состояния диктора по его речи в виде последовательности методов обработки и анализа входного речевого образца.

В процессе разработки модели эмоции для анализа эмоциональной окраски речи были выработаны следующие наиболее важные критерии, которым она должна удовлетворять. И так, разрабатываемая формализованная модель должна быть оптимальной в информационном смысле и эффективной с вычислительной точки зрения; должна обладать полнотой, т. е. содержать все необходимые параметры для классификации эмоциональной окраски речи и оценки величины потенциальной опасности; должна быть настраиваемой под те или иные условия использования

Предложенная формализованная модель эмоции, изображенная на рисунке 1.

Основным параметром представленной модели является X – вектор признаков, полученный на основе цифровой обработки речевого сигнала, который содержит информацию о частотном и энергетическом спектре сигнала, значениях таких параметров как частоты формант и основного тона, их статистических параметров, скорости изменения значений характеристик речевого сигнала, а также их модуляции.

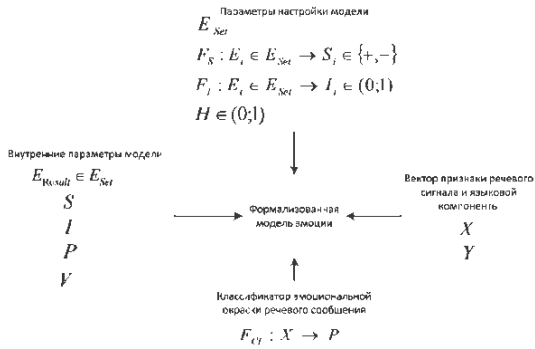


Рис. 1 – Обобщенная формализованная модель эмоции

Параметр Y представляет собой вектор признаков, полученный на основе применения семантического [7] и сентиментного [8,9] анализа языковой информации. С помощью данного параметра возможно извлечение дополнительной информации об эмоциональной окраске сообщения, которая выражена с помощью языко-знаковых систем в общем случае, или тем или иным языком. Таким образом, рассмотренные параметры X и Y предложенной модели эмоции позволяют учитывать при анализе речевого сигнала, как вербальную его составляющую, так и невербальную компоненту, что дает возможность осуществления более полного и комплексного анализа высказывания человека.

Параметр P представляет собой вектор вероятностей такой, что для его элементов выполняется следующее равенство:

$$\sum_{i=1}^{|E_{Set}|} P_i = 1, \quad (1)$$

где элементы вектора P отражают вероятность принадлежности речевого фрагмента к соответствующему типу эмоции из предопределенного множества релевантных эмоций E_{Set} . Подобный подход к описанию эмоции позволяет учесть ее фундаментальную природу – эмоциональная окраска не может быть интерпретирована строго и однозначно. При подобном описании эмоции, элементы вектора P представляют собой координаты данной эмоции в многомерном пространстве, где единичный базисный вектор соответствует отдельной эмоции. Размерность вектора P равна мощности множества возможных эмоций E_{Set} .

Функциональная зависимость $F_{Cl} : X \rightarrow P$ осуществляет отображение значений вектора признаков речевого сигнала X на вектор вероятностей P . В качестве такого функционального отображения применяется классификатор, который строится на основе обучающей выборке данных. Использование построенного классификатора в качестве самостоятельного параметра объясняется тем, что изначально избыточный входной вектор признаков, а так же и параметры используемого

классификатора могут быть оптимизированы под заданного диктора или имеющийся набор речевых образцов для достижения наилучшей точности определения типа эмоции [10]. Подобный подход дает дополнительное преимущество, а именно позволяет придать адаптивность и гибкость предлагаемой модели, т. е. использование различных векторов признаков на стадии определения типа эмоции без изменения структуры входных данных.

Следующим параметром предложенной модели является вес отдельной эмоции I , который отражает степень ее влияния на оценку величины потенциальной опасности. Данный параметр позволяет учитывать тот факт, что различные эмоции могут иметь различный вес и ассоциироваться с различными уровнями потенциальной опасности. Значение параметра I задается априорно для каждой эмоции E_i из допустимого множества E_{Set} , согласно требуемым параметрам работы системы:

$$F_I : E_i \in E_{Set} \rightarrow I_i \in (0;1), \quad (2)$$

Параметр модели S соответствует полярности эмоции. Согласно современной трактовке эмоций, положительный и отрицательный типы эмоций принимаются как противоположные и человек, в отдельный взятый момент, может находиться либо в радостном, либо в грустном эмоциональном состоянии [1]. Данное утверждение можно считать лишь условно истинным, поскольку деление эмоций на положительные и отрицательные условно, а также наличие сложных эмоций согласно некоторым исследователям [11] противоречит данному высказыванию. Для устранения двойственности интерпретации полярности эмоции полагается, что под положительной эмоцией подразумевается та эмоция, которая способствует повышению конструктивного поведения человека. Соответственно, под отрицательной эмоцией будет подразумеваться та эмоция, которая располагает к деструктивному поведению индивида. Соотнесение каждой допустимой эмоции из множества E_{Set} и ее полярности устанавливается априорно, что отражено соотношением в представленной модели:

$$F_S : E_i \in E_{Set} \rightarrow S_i \in \{+, -\}, \quad (3)$$

Полярность эмоции необходима для оценки величины потенциальной опасности.

Поскольку при анализе эмоциональной окраски речи используется два 'информационного канала' поступления данных – результаты цифровой обработки речевого сигнала и семантической / сентиментной обработки, то это дает возможность варьирования степени информативности каждого из каналов. Для реализации данного свойства модель содержит параметр H , который определяет степень информативности результатов цифровой обработки речевого сигнала при принятии решения о величине

потенциальной опасности. Соответственно, результаты семантической обработки языковой информации имеют вес $1-H$. С помощью параметра H можно 'отключать' влияние семантического и сентиментного анализа языковой информации в случае низкого качества или невозможности распознавания произнесенной речи, или же использовать только семантический / сентиментный анализ в случае анализа текстовой информации, что доставляет дополнительную универсальность и гибкость предлагаемой модели.

Параметр $E_{Result} \in E_{Set}$ соответствует типу эмоции, который присваивается речевому фрагменту после его анализа. Возможный тип эмоции ограничен predetermined множеством E_{Set} релевантных для рассмотрения/выявления типов эмоций.

С каждым входным речевым фрагментом ассоциируется характеристическая величина потенциальной опасности, полученная на основе анализа речевого высказывания, что отражено в соответствующем параметре модели V .

В представленной модели инкапсулирован необходимый набор данных, достаточный для классификации эмоций и оценки величины потенциальной опасности, а также настроечные параметры, которые позволяют настраивать модель под учет конкретных условий ее использования.

2. Модель процесса анализа речевого сообщения в системе безопасности

На основе предложенной модели эмоции разработан следующий процесс обработки речевого сообщения в системе безопасности выявления потенциально опасных ситуаций, модель которого изображена на рисунке 2.

Как видно из схемы, модель процесса анализа речевого сообщения имеет многоуровневую структуру, каждый уровень которой имеет свои задачи по анализу и обработке входных данных.



Рис. 2 – Модель процесса анализа речевого сообщения

Первым уровнем представленной модели является уровень предварительной обработки речевого сообщения. На данном уровне осуществляется цифровая обработка

сегментированного сигнала с целью повышения его качества, а также распознавание произнесенной речи и представления ее в текстовом виде для дальнейшей обработки и анализа. Упомянутый уровень находится за рамками рассмотрения представленной работы, поскольку в настоящее время существуют программные средства, которые позволяют решать задачу распознавания речи с приемлемой точностью [12]. Разрабатываемая система анализа эмоциональной окраски речи подразумевает использование исходного речевого сигнала для анализа его эмоциональной окраски и текстового сообщения, которое является распознанной версией произнесенной фразы или выражения для получения языковой компоненты сообщения.

Этап предварительной обработки в предлагаемой модели процесса анализа речевого сообщения позволяет осуществлять 2-х режимный вариант работы. Первым режимом работы является режим сканирования информационного потока путем поиска только априорно заданных слов-индикаторов, которые могут указывать на наличие потенциальной опасности. При обнаружении хотя бы одного predetermined слова-индикатора потенциальной опасности, система генерирует соответствующее сообщение. Подобный режим работы системы целесообразен в случае работы в реальном масштабе времени при анализе высокоинтенсивных потоков речевой информации и наличии ограниченных вычислительных ресурсов для их обработки. Неизбежным следствием снижения вычислительных ресурсов является возможность пропуска целевой информации, если она передается с использованием синонимии языка. Для осуществления детального анализа информационного потока система должна работать во втором режиме с использованием семантического графа для поиска целевого содержания.

Следующим уровнем представленной модели процесса анализа эмоциональной окраски является вычисление вектора признаков. Поскольку разрабатываемая система позволяет анализировать как речевой сигнал, так и языковую составляющую речевого сообщения, то на данном этапе происходит вычисление двух векторов признаков. Первый вектор признаков состоит из акустических признаков – значений различных параметров речевого сигнала, которые вычисляются с помощью методов цифровой обработки сигналов. Второй вектор признаков состоит из семантических признаков речевого сообщения, которые вычисляются с помощью методов семантического анализа. Особенностью данного уровня рассматриваемой модели анализа речевого сообщения является то, что после вычисления вектора акустических признаков происходит удаление некоторых его компонент. Процесс осуществляется на основе данных, которые получены блоком оптимизации входного вектора

признаков, что позволяет исключить малоинформативные элементы вектора признаков и тем самым оптимизировать процесс классификации эмоциональной окраски речевого сообщения путем уменьшения количества используемых признаков.

Третьим уровнем предложенной модели анализа речевого сообщения является уровень классификации эмоциональной окраски. Входными данными для него являются вектора признаков с акустическими и семантическими признаками, которые подаются на вход классификатора эмоциональной окраски. Результатом работы классификатора является вектор вероятностей P . Важным преимуществом предложенной модели сообщения является то, что применяемый классификатор является обучаемым, то есть он проходит настройку с помощью обучающей выборки данных. Применение подобного подхода позволяет реализовать более гибкий процесс анализа речевого сообщения, который оптимизируется под значения входных данных. Дополнительным преимуществом предложенной модели является то, что настроечные параметры классификатора вычислены с учетом входных данных и оптимизированы под использование найденного вектора признаков, при котором используемый классификатор работает наилучшим образом.

Заключительным уровнем рассматриваемой модели является уровень принятия решения об уровне потенциальной опасности, которой включает в себя два этапа. На первом этапе принимается решение о финальном типе эмоции и его знаке, к которому будет отнесен исследуемый речевой фрагмент. Основой выбранного решения является вектор вероятностей P , набор нечетких правил R и конфигурационная настройка, связывающая каждый тип эмоции с соответствующей ей полярностью и заданным весом.

Заключение

Предложенная модель позволяет перейти от качественной характеристики человеческих эмоций к их количественной параметризации и численной оценки.

Подробно рассмотрев модель эмоции и процесса анализа речевого сообщения можно отметить такие преимущества как гибкость и универсальность, т. е. модель может быть настроена под определенные требования; параметры модели могут быть адаптированы под используемые данные для достижения лучших результатов классификации эмоционального состояния.

Предложенный процесс анализа речевого сигнала в системе безопасности позволяет получить количественную оценку уровню потенциальной опасности на основе комплексного анализа, как

речевой компоненты, так и языковой составляющей речевого сообщения.

Библиографический список

- [Ильин, 2001] Ильин Е. П. Эмоции и чувства / Е. П. Ильин. – СПб: Питер, 2001. – 752 с.
- [Schuller, 2007] Towards More Reality In The Recognition of Emotional Speech : proceedings of the 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). – 2007. – P. 941-944.
- [Schröder, 2001] Emotional Speech Synthesis: A Review: proceedings of the 7th European Conference on Speech Communication and Technology. – 2001.
- [Petrushin, 1999] Emotion in speech: Recognition and application to call centers : proceedings of the conference Artificial Neural Networks in Engineering (ANNIE'99). – 1999.
- [Fragopanagos, 2005] Fragopanagos N. Emotion recognition in human-computer interaction / N. Fragopanagos, J.G. Taylor // Neural Networks. – 2005. – No. 18. – P. 389-405
- [Yacoub, 2003] Recognition of Emotions in Interactive Voice Response Systems : proceedings of the 8th European Conference on Speech Communication and Technology. – 2003. – p. 1-4.
- [Yih, 2012] Polarity Inducing Latent Semantic Analysis: proceedings of the conference on empirical methods in natural language processing and computational natural language learning. – 2012.
- [Wilson, 2005] Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis: proceedings of HLT-EMNLP. – 2005. – P. 347-354.
- [Hernández, 2009] Recognizing Polarity and Attitude of Words in Text: proceedings of the 14th Portuguese Conference on Artificial Intelligence. – 2009.
- [Пекарь, 2012] Пекарь Д. В. Алгоритм оптимизации входных признаков и параметров классификатора эмоционального состояния человека по его речи / Д. В. Пекарь, С. Г. Тихоненко, А. Ф. Чернявский, В. С. Садов // Информатика. – 2012. – № 3(35).
- [Розалиев, 2008] Розалиев В. Л. Предпосылки, возможности, перспективы создания автоматизированной системы распознавания эмоциональности речи // Известия Волгоградского государственного технического университета: межвузовский сборник научных статей. – 2008. – №2(40). – С.58-61.
- [Walker, 2004] Walker W. Sphinx-4: a flexible open source framework for speech recognition / W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, J. Woelfel // Sun Microsystems. Inc. – 2004.

GENERALIZED MODEL OF HUMAN EMOTION AND PROCESS FLOW OF SPEECH SIGNAL ANALYSIS IN SECURITY SYSTEM

Pekar D. V. *, Sadov V. S. *

* *Belarusian State University,
Minsk, Republic of Belarus*

pekar.dima@gmail.com

sadov@bsu.com

In presented paper formalized model of human emotion and process flow of speech signal analysis is described. Proposed model can formalize human emotion and parameterize its values. Also it handles all required attributes for further automatic processing. Presented model has advantages such as ability to be customized to fit specific needs, could be adapted to incoming data without changes in input interface. Described process flow allows to estimate level of potential danger based on analysis of voice and lingual component of human speech.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.942; 781.1

ОПТИМИЗАЦИЯ ОКОННОГО ФУРЬЕ И НЕПРЕРЫВНОГО ВЕЙВЛЕТ ПРЕОБРАЗОВАНИЙ ДЛЯ СПЕКТРАЛЬНОГО АНАЛИЗА МУЗЫКАЛЬНЫХ СИГНАЛОВ

Алиев Р.М.

*Белорусский государственный университет,
г. Минск, Республика Беларусь*

RomanAliyev@gmail.com

Предлагается метод для улучшения частотно-временного разрешения при спектральном анализе музыкальных сигналов. Этот метод позволяет найти оптимальные параметры спектральных преобразований и состоит из трех этапов – спектральное моделирование музыкальных звуков, синтез музыкальных спектров и сравнение вычисленного спектра с синтезированным спектром. В ходе экспериментов с базой данных классической музыки были исследованы оптимальный размер окна в оконном преобразовании Фурье и оптимальная компактность материнского вейвлета в непрерывном вейвлет преобразовании.

Ключевые слова: распознавание музыки; спектральный анализ; оконное преобразование Фурье; непрерывное вейвлет преобразование.

ВВЕДЕНИЕ

За последние тридцать лет благодаря развитию компьютерных технологий и цифровой обработки сигнала в решении задач распознавания музыки было достигнуто множество значимых результатов и было предложено множество алгоритмов [Gómez et al., 2003] [Peeters, 2004], позволяющих более качественно выделять спектральные характеристики музыкального звука. В частности, один из них – алгоритм непрерывного вейвлет преобразования получил широкий интерес в распознавании музыки после публикации работы Ж. Морле о приложении вейвлетов к анализу звуковых паттернов [Kronald-Martinet et al., 1987] и позиционируется сейчас как хорошая альтернатива Фурье преобразованию. Он является аналогом многомасштабного преобразования [Brown, 1991], т.е. позволяет выделять низкие и высокие частоты сигнала в разных масштабах времени. Потому, безусловно, в отличие от Фурье преобразования, вейвлет-анализ предоставляет возможности для получения спектра звука с лучшей частотно-временной точностью.

Более наглядно последнее заключение проиллюстрировано в [Mallat, 2002] и [Addison, 2002] в контексте принципа неопределенности Гейзенберга, согласно которому можно сказать, что чем сильнее Фурье или вейвлет базис локализован во времени, тем шире его спектр и наоборот. Поэтому результат любого спектрального

преобразования имеет ограниченное частотно-временное разрешение и спектральные свойства современной музыки не укладываются в эти ограничения. Например, для разделения в спектре двух частот 27.5 Гц и 29.1 Гц, соответствующих двум наименьшим по высоте нотам фортепиано А0 и А#0, потребуется Фурье преобразование с размером окна 0.6 секунд. При применении в этом случае комплексного вейвлета Морле с областью определения, равном 8-ми, анализируемым частотам будут соответствовать масштабы вейвлета с временными длительностями 0.29 и 0.27 секунд и частотными интервалами 3.4 Гц и 3.7 Гц. Таким образом, элементы базиса Фурье, отличающиеся слабой локализацией, приводят к нежелательному размазыванию временного спектра, а компактность носителя вейвлет базиса является причиной недостаточного частотного разрешения для выделения фундаментальных частот музыкального строя. Отсюда можно сделать вывод, что оптимальной спектральной картины можно достичь только подстройкой размера окна в Фурье преобразовании или степени компактности вейвлет базиса в вейвлет преобразовании под спектральные свойства музыки.

Поэтому в данной работе была поставлена цель, разработать методологию для поиска компромиссного частотно-временного разрешения в алгоритмах спектрального анализа музыки. Для достижения этой цели и для синтеза эталонных музыкальных спектров будет использован подход на

основе спектрального моделирования звука, который является достаточно эффективным методом для моделирования и синтеза музыки [Bonada et al., 2011]. В качестве эксперимента согласно разработанной методологии будет проведено исследование оптимального размера окна в оконном преобразовании Фурье и оптимальной компактности материнского вейвлета в вейвлет преобразовании для базы данных с классической музыкой.

Предлагаемый метод

Предлагаемый метод включает в себя три этапа (Рис. 1). Первый – спектральное моделирование музыкальных звуков, главная задача которого – представить необходимые нотные звуки музыкальных инструментов в виде спектральных характеристик, которые затем используются на втором этапе для синтеза музыкального звука и спектра, согласно заданному музыкальному тексту. Далее на последнем этапе производится поиск оптимальных параметров для исследуемого алгоритма спектрального анализа музыки путем минимизации расстояния между огибающими функциями синтезированного спектра и огибающими функциями спектра, полученного в результате спектрального анализа.

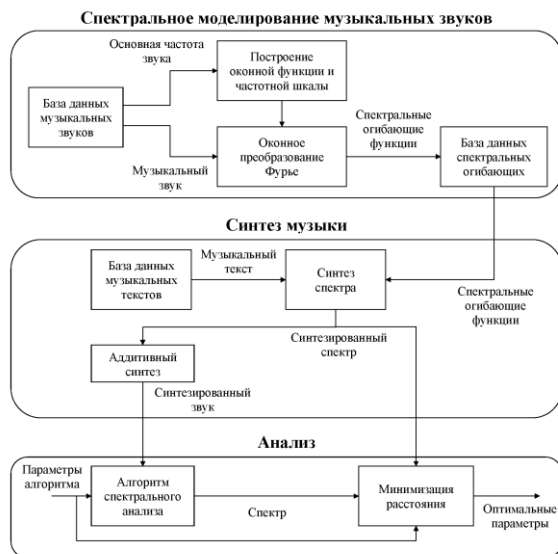


Рисунок 1 – Общая схема метода для поиска компромиссного частотно-временного разрешения в алгоритмах спектрального анализа музыки

Здесь в первую очередь стоит отметить, что в базе данных музыкальных звуков должны содержаться все записи нот музыкальных инструментов, описанные в музыкальных текстах. Кроме этого для возможности синтеза и сравнения спектров необходимо, чтобы на каждом из трех этапов метода использовалась одна и та же частотная шкала.

Для моделирования принципов звукообразования простейших музыкальных звуков был использован метод спектрального

моделирования музыки, суть которого заключается в извлечении из естественных звуков характерных для музыки спектральных огибающих функций с помощью оконного преобразования Фурье:

$$F(\tau, f) = \int_{\tau-T/2}^{\tau+T/2} s(t)w\left(\frac{t-\tau}{T}\right)e^{-2i\pi f(t-\tau)} dt, \quad (1)$$

где $F(\tau, f)$ - функция спектральной огибающей; τ и f - момент времени и частота преобразования соответственно; $s(t)$ - анализируемый сигнал; $w(t)$ - оконная функция; T - ширина оконной функции.

Оконная функция и множество частот преобразования строятся исходя из значения основной частоты анализируемого звука. Согласно музыкальному равномерно темперированному строю возможные значения основной частоты удовлетворяют следующей формуле:

$$f'(n) = 440 \cdot 2^{(n-57)/12}, \quad (2)$$

где $n \geq 0$ – порядковый номер ноты в музыкальном строе.

Значения гармонических частот музыкального звука основываются на целочисленных соотношениях с основной частотой и могут быть вычислены следующим образом:

$$f''(k) = kf_0, \quad (3)$$

где $k \geq 1$ – порядковый номер гармоники; f_0 – основная частота звука.

Однако только первые двадцать гармоник в спектре звука являются наиболее значительными [Алдошина и др., 2006]. Отсюда можно заметить, что разница между любой из первых двадцати гармонических частот и ближайшей к ней частоте из музыкального строя не существенна. Если этим пренебречь, тогда, используя только частоты музыкального строя, допустимо анализировать и музыкальную тональность, и гармоники звука. Таким образом, частоты преобразования (и частотная шкала всего метода) могут быть заданы следующим образом:

$$\begin{cases} f(f_0, k) = 440 \cdot 2^{(n-57)/12} \\ n = \arg \min_{n'} |kf_0 - 440 \cdot 2^{(n'-57)/12}| \\ 1 \leq k \leq 20 \end{cases} \quad (4)$$

В качестве оконной функции была выбрана оконная функция Таки:

$$w(x) = \begin{cases} \sin^2(2\pi x), & 1/4 < |x| \leq 1/2 \\ 1, & |x| \leq 1/4 \end{cases} \quad (5)$$

Благодаря плоской вершине она слабо локализует сигнал во временной шкале, что позволяет получить высокую четкость спектра в частотной шкале. Кроме этого в отличие от остальных оконных функций высокого разрешения

(например, прямоугольного окна) окно Таки имеет незначительное влияние боковых лепестков для извлечения гармонических частот музыкальных звуков.

Для вычисления оптимальной ширины оконной функции оконное преобразование Фурье следует рассмотреть в качестве полосопропускающего фильтра. Тогда ширина окна Таки задается следующим образом:

$$T = \frac{1}{0,75 \cdot \Delta f_{0,5}}, \quad (6)$$

где $\Delta f_{0,5}$ – ширина частотной полосы по амплитуде, равной 0,5.

Т.к. расстояние между гармоническими частотами в спектре музыкального звука равно основной частоте, поэтому для достижения максимально возможной временной четкости спектральных огибающих функций и для минимизации возможных отклонений гармонических частот спектра от ожидаемых частот можно принять $\Delta f_{0,5} = f_0$.

На этапе синтеза осуществляется синтез спектра и звука музыкальных композиций. Ключевыми источниками информации для осуществления этого процесса является база данных музыкальных текстов, содержащая музыкально-лингвистические свойства музыкальных композиций, и база данных спектральных огибающих, отражающая спектральные свойства каждой ноты музыкального текста. Аналитически синтез спектра может быть представлен следующим образом:

$$Spec(t, f) = \sum_{n \in N_t} (F_n(t - \tau_n, f) \cdot E_n(t - \tau_n)), \quad (7)$$

где $Spec(t, f)$ – функция синтезированного спектра; N_t – множество нот, проигрываемое в момент времени t ; F_n – огибающая спектра для ноты n ; τ_n – момент времени в спектре, когда нота n начинает воспроизводиться; $E_n(t)$ – огибающая функция, задающая громкость воспроизведения и затухание ноты n :

$$E_n(t) = \begin{cases} a, & 0 \leq t \leq d \\ a - (t - d) / d', & d < t \leq d + d' \end{cases} \quad (8)$$

где $0 \leq a \leq 1$ – громкость воспроизведения; d и d' – длительности воспроизведения и затухания ноты.

Далее с помощью аддитивного синтеза воссоздается звуковой сигнал музыкального произведения:

$$S(t) = \sum_f (Spec(t, f) \cdot \sin(2\pi ft)). \quad (9)$$

В процессе анализа производится минимизация расстояния между синтезированным спектром и

спектром, извлеченным из синтезированного звука согласно следующей формуле:

$$p = \arg \min_{p'} D(Spec, T(S, p')), \quad (10)$$

где p – оптимальное значение исследуемого параметра преобразования; D – функция расстояния; T – преобразование, которое лежит в основе исследуемого алгоритма спектрального анализа; $Spec$ и S – синтезированный спектр и сигнал музыкальной композиции соответственно.

В качестве функции расстояния D предлагается использовать манхэттенскую метрику. Она не содержит возведений в степень, что снижает влияние больших разностей на точность вычислений с плавающей запятой:

$$D = \sum_{t,f} |Spec'(t, f) - Spec''(t, f)|. \quad (11)$$

Эксперимент

В ходе эксперимента на основе выше описанной методологии были исследованы параметры двух преобразований. Первый из них – размер окна в оконном преобразовании Фурье с окном Гаусса:

$$w(x) = e^{-x^2/0,045} \quad (12)$$

И второй – компактность вейвлета в непрерывном вейвлет преобразовании на основе комплексного вейвлета Морле:

$$\psi(x) = e^{-(x/l_0/l)^2/2} * e^{-i2\pi x}, \quad (13)$$

где $l_0 = 10$ – ширина области определения вейвлета; l – компактность вейвлета (ширина временной локализации).

Таблица 1 – Список промоделированных акустических музыкальных звуков.

Инструмент	Звуки
Саксофон	D#3-F6
Банджо	C4-E7
Фагот	A#1-F5
Виолончель	C2-C6
Кларнет	D3-H6
Контрабас	C1-G4
Флейта	C4-D7
Гитара	E2-D5
Гобой	H3-G#6
Орган	C1-C6
Рояль	A0-C8
Тромбон	E2-E5
Труба	F3-D6
Туба	F1-F4
Альт	C3-C7
Скрипка	G3-G7

В качестве базы данных музыкальных текстов было использовано 610 30-секундных MID отрывков классической музыки. Также были промоделированы необходимые для синтеза звуки

акустических музыкальных инструментов (Табл. 1). Все звуковые записи эксперимента хранились в формате WAV IEEE Float с частотой дискретизации 44100 Гц без сжатия.

Затем, используя каждое из двух исследуемых преобразований, из каждой синтезированной мелодии при заданном параметре преобразования извлекался спектр с временным шагом 10 мс в частотной шкале, согласно формуле (4). Таким образом, каждому музыкальному отрывку соответствовало оптимальное значение параметра преобразования. В результате для оконного преобразования Фурье было вычислено распределение оптимального размера окна в диапазоне от 50 мс до 1000 мс с шагом 50 мс, а для вейвлет преобразования – распределение оптимальной компактности вейвлета в диапазоне от 10 до 150 с шагом 5 (Рис. 2).

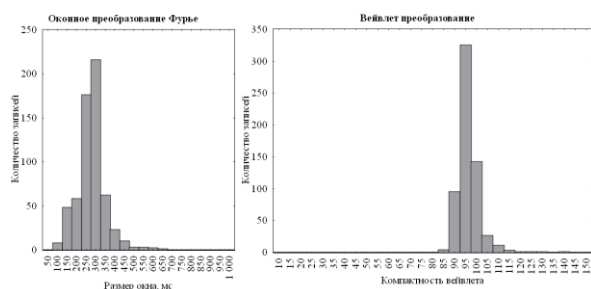


Рисунок 2 – Распределение значений оптимальных параметров преобразований.

На Рис. 2 заметна ярко выраженная концентрация значений размера окна вблизи значения 300 мс и значений компактности вейвлета вблизи значения 95. Следовательно, можно предположить, что при этих значениях в большинстве случаев исследуемые алгоритмы будут разделять фундаментальные частоты классической музыки с оптимальной частотно-временной точностью.

ЗАКЛЮЧЕНИЕ

Предложенный метод позволяет достигать оптимальной частотно-временной точности в алгоритмах спектрального анализа музыки. Этому способствует подход на основе спектрального моделирования музыкальных звуков. Результаты проведенного эксперимента с классической музыкой соответствуют гипотезе исследования об оптимальных значениях параметров преобразований. Следовательно, целесообразно дальнейшее развитие предложенного метода, путем моделирования музыки других жанров, а также путем исследования других типов преобразований.

БЛАГОДАРНОСТИ

Автор выражает особую благодарность музыкальной школе Аризонского университета, а так же организаторам проектов «Sound Exchange», «Composition Today» и «Midiworld» за безвозмездно предоставленные музыкальные тексты и звуковые

записи акустических инструментов. Автор благодарит своего научного руководителя Хейдорова И.Э. за консультацию в проведении эксперимента.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Gómez et al., 2003] E. Gómez. Melody description and extraction in the context of music content processing / E. Gómez, A. Klapuri, and B. Meudic // Journal of New Music Research. – 2003 – V. 32(1). P. 23-40.
- [Peeters, 2004] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project / G. Peeters // Technical report, UIDADO IST Project. – 2004.
- [Kronald-Martinet et al., 1987] R. Kronald-Martinet. Analysis of sound patterns through wavelet transform / R. Kronald-Martinet, J. Morlet, A. Grossman // International Journal of Pattern Recognition and Artificial Intelligence. – 1986. – V. 1. – № 2. – P. 273-302.
- [Brown, 1991] Brown J.C. Calculation of a constant q spectral transform / Brown J.C. // Journal of the Acoustical Society of America. – 1991. – V. 89. – № 1. – P. 425-434.
- [Mallat, 2002] Stéphane G. Mallat. A Wavelet Tour of Signal Processing / Stéphane G. Mallat // Academic Press. – 2009. – P. 23-34.
- [Addison, 2002] Paul S. Addison. The Illustrated Wavelet Transform Handbook / Paul S. Addison // Taylor & Francis Group. – 2002. – P. 35-51.
- [Bonada et al., 2011] Bonada J. Spectral Processing / Bonada J., Serra X., Amatriain X., Loscos A. // Book Chapter in DAFX Digital Audio Effects. – 2011. – P. 393-446.
- [Алдошина и др., 2006] И. Алдошина. Музыкальная акустика / И. Алдошина, Р. Приттс // Учебник. СПб.: Композитор. – 2006.

OPTIMIZATION OF SHORT-TIME FOURIER AND CONTINUOUS WAVELET TRANSFORMS FOR SPECTRAL ANALYSIS OF MUSICAL SIGNALS

R.M. Aliyev

*Belarusian State University,
Minsk, Republic of Belarus*
RomanAliyev@gmail.com

The method to improve time-frequency accuracy during spectral analysis of musical signals is offered. The method optimizes the parameters of spectral transforms and includes three stages – spectral modeling of sound samples, synthesis of music spectrums, and comparison of the synthesized and computed spectrums. Experiments with classical music databases presented the optimum window size in short-time Fourier transform and the optimum wavelet compactness in continuous wavelet transform are explored.

Key words: music recognition, spectral analysis, short-time Fourier transform, continuous wavelet transform.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СИСТЕМА РАСПОЗНАВАНИЯ ЖЕСТОВ РУССКОГО ЯЗЫКА ГЛУХИХ

Дорофеев Н.С., Розалиев В.Л., Заболева-Зотова А.В.

*Волгоградский Государственный Технический Университет,
г. Волгоград, Россия*

nikita.dorofei@gmail.com

vladimir.rozaliiev@gmail.com

zabzot@gmail.com

Данная статья посвящена исследованиям в области автоматизации распознавания жестов русского языка глухих. Авторами предложена информационная модель распознавания дактильных жестов с использованием камеры с сенсором глубины. Разработана удобная модель описания кисти руки, использование которой значительно повышает точность распознавания. Также разработан улучшенный метод детектирования рук на изображении.

Ключевые слова: язык жестов; распознавание жестов; камера с сенсором глубины

ВВЕДЕНИЕ

Адаптация слабослышащих и глухих людей к жизни современного общества является крайне важной задачей. Её достижению могут помочь новые технологии в сфере распознавания жестов. Традиционные визуальные методы распознавания жестов всё ещё недостаточно точные для использования в реальных приложениях. Одной из причин этому являются ограничения оптических сенсоров чувствительных к условиям освещения и шумам фона. Для достижения высокой надежности распознавания жестов также используются информационные перчатки. Такие сенсоры гораздо надежнее, но имеют существенные недостатки: пользователю приходится носить перчатку, которая иногда требует калибровки, что очень неудобно и делает движения не натуральными. Также перчатки зачастую очень дорогие, хотя в последнее время и ведутся работы по уменьшению стоимости таких устройств. В результате, этот метод распознавания жестов не очень популярен. Благодаря разработке компании Microsoft камеры с сенсором глубины Kinect, появились новые возможности для распознавания жестов. Сенсоры глубины существовали уже относительно давно, но Kinect обладает перед ними рядом значительных преимуществ: большое распространение, относительно небольшая стоимость и наличие RGB камеры. Получаемая с Kinect карта глубины инвариантна к условиям освещения и фону,

поскольку она основана на инфракрасном излучении. Помехой для данного типа записывающих устройств может быть только сильный туман и некоторые другие погодные условия.

Несмотря на успешные применения Kinect к распознаванию лица и отслеживанию тела человека, проблема использования этого сенсора к распознаванию небольших жестов рук до сих пор является нерешенной. Основная причина - низкое разрешение карты глубины сенсора [Kean et al., 2011].

В языках жестов передача информации во время общения происходит по нескольким каналам: непосредственно через жесты руками, выражение лица, форму губ, положение тела и головы. Сами жесты руками описываются через положение рук, направление движения, форму и направление кистей рук. Таким образом, возможность определения формы и положения кистей рук является очень важной задачей в контексте распознавания жестового языка, которая еще не была полностью решена.

В данном исследовании была поставлена цель расширения возможностей коммуникации глухонемых с окружающим миром за счёт распознавания дактильных букв русского языка жестов в режиме реального времени. В качестве устройства ввода информации о жесте использовался Microsoft Kinect.

1. Информационная модель распознавания статических жестов

Весь процесс распознавания дактильных жестов можно представить в виде информационной модели:

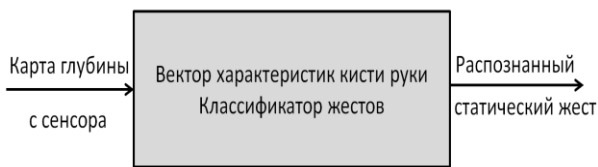


Рисунок 1 - Информационная модель распознавания статических жестов

$$IM = \langle FV, C, I, O \rangle, \quad (1)$$

где IM – информационная модель;

FV – вектор характеристик кисти руки. Первоначальная информация об изображении (цвет и расстояние) анализируется: определяется положение кистей рук и вектор их характерных черт;

C – классификатор жестов. По найденному вектору характерных черт кистей, определяется, какой жест был показан;

I – множество входных значений – карта глубины – расстояние до каждой точки на изображении;

O – множество выходных значений – распознанный на изображении статический жест.

2. Метод детектирования рук на изображении

Первым этапом распознавания является сегментация полученного изображения с сенсора Kinect с целью нахождения на нем кисти руки или обеих рук. Разработка метода для нахождения кисти руки на изображении является одной из самых сложных проблем в процессе создания системы распознавания жестов. Существует несколько признаков, по которым можно детектировать объект на изображении: внешность, форма, цвет, расстояние до объекта и контекст. В таких случаях, как детектирование лица на изображении, хорошим признаком является внешность, так как глаза, нос и рот всегда будут находиться примерно в одинаковых пропорциях. Поэтому основанный на характеристиках внешности объекта метод каскадов Хаара отлично применяется для распознавания лица. В случае распознавания рук дело обстоит сложнее: надежный метод распознавания может быть реализован основываясь, в основном, на цветовых характеристиках. Так как цвет рук может меняться в зависимости от человека и контекста, представляется разумным сначала найти лицо человека на изображении и получать информацию о цвете рук исходя из цвета лица. Введенное ограничение наличия лица человека на изображении

в любом случае является обязательным, поскольку распознавание жестового языка без распознавания лица будет ненадежным.

Имея информацию о цвете объекта, необходимо детектировать его на изображении. Осуществить данную задачу можно с помощью алгоритма Camshift, надежность которого доказана в работах [Hai et al., 2011]. Модель этого алгоритма основана на гистограммах и является обучающейся в процессе распознавания. Естественно, данный алгоритм найдет все объекты данного цвета на изображении. Чтобы этого не допустить, используется информация о расстоянии до объектов на изображении, т.е. карта глубины с сенсора Kinect.

Итак, после нахождения позиции (x,y) и размеров (w,h) лица на изображении с помощью метода каскадов Хаара, можно найти усредненное расстояние до лица, используя карту глубины D:

$$d_f = \frac{1}{wh} \sum_{i=x}^{x+w} \sum_{j=y}^{y+h} D(i, j). \quad (2)$$

Все объекты, находящиеся ближе к камере, чем само лицо человека, могут быть найдены с использованием порогового значения:

$$D(i, j) > d_f + t_h, \quad (3)$$

где t_h – параметр, определяющий, насколько близко должны быть поднесены руки к камере, чтобы показанные жесты распознавались системой.

Таким образом, в системе делается допущение, что руки находятся ближе к камере, чем остальные части тела человека, что вполне обосновано в контексте системы распознавания жестового языка.

3. Исследование контура кисти руки

На втором этапе распознавания на сегментированном изображении находятся контуры оставшихся объектов, т.е. рук человека. Экспериментально было выяснено, что самым эффективным и информативным методом нахождения контуров на изображении в контексте данного исследования, является детектор границ Кенни. После его применения, полученные контуры представляются в виде кривой на графике (для каждой руки своя кривая), как показано на рисунке 2.

График показывает относительное расстояние каждой точки контура к центральной точке. Последняя определяется, как центр вписанной в контур окружности с максимальным радиусом. Начальная точка определяется согласно линии, найденной алгоритмом RANSAC по черному ремешку на запястье человека. В данном представлении, на горизонтальной оси находится угол между каждой точкой контура и начальной точкой относительно центральной точки, нормализованный на 360°. На вертикальной оси

находится относительное Евклидово расстояние между каждой точкой контура и центральной точкой, нормализованное по радиусу максимальной вписанной окружности. Такое представление кисти руки хорошо описывает её топологические свойства – расположение пальцев. Положение последних можно определить, задав некоторое пороговое значение t . Хотя найти оптимальное значение t является непростой задачей. [3]

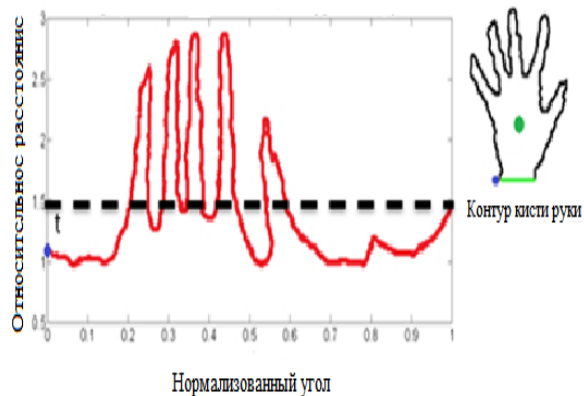


Рисунок 2 - Графическое представление контура кисти руки

4. Классификация полученных данных

На третьем этапе необходимо выбрать метод классификации полученных данных. В ходе исследования было протестировано более 10 различных методов классификации. Самым точным методом показал себя многослойный перцептрон (98% в некоторых случаях). Надежность нейросетевого подхода подтверждается и другими исследованиями [Розалиев и др., 2010], [Заболеева-Зотова и др., 2011a]. Далее на основе обучающей выборки, состоящей из данных, описывающих жест в виде, представленном на рисунке 2, была обучена нейронная сеть с найденными в ходе экспериментов оптимальными параметрами. Для тестирования был разработан прототип системы распознавания дактильных букв. Нейронная сеть была обучена 13 жестам – буквам русского жестового языка. Тесты показали достаточно высокую точность распознавания – 88%.

ЗАКЛЮЧЕНИЕ

В ходе данного исследования была разработана система распознавания дактильных жестов русского жестового языка, которая была обучена 13 жестам.

Как было отмечено, жестовый язык состоит из целого ряда каналов передачи информации, поэтому распознавание небольших жестов рук не решает проблему распознавания русского жестового языка полностью, но является важным элементом будущей полноценной системы.

Предметом дальнейшего исследования является поиск более точного, чем пороговое значение,

метода определения характерных черт кисти руки на графике, поскольку при некоторых положениях пальцев могут происходить коллизии. Из-за этого возникают затруднения в обучении системы полной азбуке дактильных жестов.

Полученные результаты планируется использовать в работах по определению эмоциональных реакций человека [Заболеева-Зотова и др., 2011b], [Розалиев и др., 2010], [Заболеева-Зотова и др., 2012]. Используемые в данном исследовании методы распознавания мелких движений рук человека, включая движения пальцами, безусловно, найдут применение и в этой области.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 12-07-00266, 12-07-00270).

Библиографический список

[Kean et al., 2011] Sean Kean, Jonathan Hall, Phoenix Perry, Meet the Kinect: An Introduction to Programming Natural User Interfaces (Technology in Action). Apress, 2011. – 220 с.

[Hai et al., 2011] H. Hai, L. Bin, H. BenXiong and C. Yi, "Interaction System of Treadmill Games based on depth maps and CAM-Shift", IEEE 3rd International Communication Software and Networks (2011), pp.219-222

[Заболеева-Зотова и др., 2011a] Применение нечётких темпоральных высказываний для описания движений при эмоциональных реакциях / А.В. Заболеева-Зотова, Ю.А. Орлова, В.Л. Розалиев, А.С. Бобков // Изв. ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах". Вып. 10 : межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2011. - № 3. - С. 60-64.

[Розалиев и др., 2010] Розалиев, В.Л. Применение нейронных сетей и грануляции при построении автоматизированной системы определения эмоциональных реакций человека / В.Л. Розалиев, А.С. Бобков, О.С. Федоров // Изв. ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах»: межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2010. - Вып. 9, № 11. - С. 63-68.

[Заболеева-Зотова и др., 2011b] Развитие системы автоматизированного определения эмоций и возможные сферы применения / А.В. Заболеева-Зотова, Ю.А. Орлова, В.Л. Розалиев, А.С. Бобков // Открытое образование. - 2011. - № 2. - С. 59-62.

[Розалиев и др., 2010] Розалиев, В.Л. Моделирование эмоционального состояния человека на основе гибридных методов / В.Л. Розалиев, А.В. Заболеева-Зотова // Программные продукты и системы: международный науч.-практ. журнал. – Тверь, 2010 – Вып.2 (90). – С.141-146.

[Заболеева-Зотова и др., 2012] Заболеева-Зотова, А.В. Automated Identification of Emotional States / Заболеева-Зотова А.В., Розалиев В.Л., Бобков А.С. // Advances in Decision Technology and Intelligent Information Systems. Vol. XIII : 24th Int. Conf. of Systems Research, Informatics and Cybernetics (July 30 – August 3, 2012, Baden-Baden, Germany) / The Int. Institute for Advanced Studies in Systems Research and Cybernetics. – [Tecumseh], Canada, 2012. – P. 21-25. – Англ.

SYSTEM FOR RUSSIAN SIGN LANGUAGE RECOGNITION

Dorofeev N.S., Rozaliev V.L., Zaboleeva-Zotova A.V.

*Volgograd State Technical University,
Volgograd, Russia*

zabzot@gmail.com

vladimir.rozaliev@gmail.com

nikita.dorofei@gmail.com

This article is devoted to research in the field of gesture recognition of the Russian Sign Language. The authors propose an information model of dactyl gesture recognition using sensor Microsoft Kinect. A convenient model for hand description was developed, the use of which greatly improves the accuracy of recognition. Also an improved method for hands detection in the picture was elaborated.

INTRODUCTION

Adaptation of hearing impaired people to the life of the modern society is an extremely important problem. New technologies in the field of gesture recognition can help to solve it. Traditional visual methods of gesture recognition are still not accurate enough for using in real applications. One of the reasons for this is that optical sensors have some limitations and they are very sensitive to lighting conditions and background noises.

To achieve high reliability of gesture recognition data gloves are often used. These sensors are much more reliable, but have significant drawbacks: the user has to wear a glove, which sometimes requires calibration, what is very uncomfortable and makes movements not natural. Also gloves are often very expensive, although in recent years situation has become better. As a result, this method of gesture recognition is not very popular. Due to the development of Kinect – Microsoft's camera with depth sensor, new opportunities for gesture recognition appeared. Despite the successful use of Kinect to face recognition and body tracking, the problem of application of this sensor to the recognition of subtle hand gestures is still not solved. The main reason – sensor has low resolution of depth map.

In sign languages information transmission during the communication is performed using several channels: directly through the hands gestures, face expression, mouth shape, location of body and head. Hand gestures can be described through hands and arms location, direction of the movement, shape of palms and their orientation. Thus capability of hands shape and direction recognition is very important task in the context of sign language recognition, which isn't solved completely yet.

This research aims to enlarge opportunities for deaf people to communicate with the outside world by recognizing dactyl letters of the Russian sign language

in real time. As an input device for gestures Microsoft Kinect was used.

MAIN PART

In the first section of this report information model of static gesture recognition was presented.

In the second part method for hand tracking on the image was suggested. It is based on the skin color. Since hands color can change depending on human and context, it is reasonable to find face on the image as a first step and then use its color to find all skin-colored objects, including hands. To perform that task Haar cascades are used for face detection and Camshift algorithm for further segmentation based on previously found skin color. To find hands on the image depth map from the Kinect is used. Average distance to the face is calculated. All objects that are situated closer to the camera than face are considered to be hands (threshold value also can be specified).

In the next section approach for analysis of found hand is described. To extract main features of shown gesture only contours of hands are studied. Values of relative distance from each point on the contour to the contour center are calculated. Angles between starting point and each point on the contour, normalized by 360°, are taken into account too. Starting point is defined as the first point on the black belt, which is put on the wrist. Those values describe hand well from the topological point of view.

In the fourth part of the report method for classification of obtained features vector is discussed. During the research more than 10 classification methods were tried. The best results were shown by neural networks. For testing purposes the system of Russian sign language dactyl letters was developed. It was taught 13 gestures. Average accuracy of recognition is quite high – 88%.

CONCLUSION

The system for Russian sign language dactyl letters was developed in the course of that research. It has shown quite high level of recognition accuracy on 13 gestures - letters of Russian sign language.

Results of that research are going to be used in works on emotional reactions recognition. Methods of subtle hands movement recognition used in this work undoubtedly will have application in that field too.

This work was partially supported by the Russian Foundation for Basic Research (projects 11-07-00398, 12-07-00266, 12-07-00270).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

УНИФИЦИРОВАННЫЕ СЕМАНТИЧЕСКИЕ МОДЕЛИ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ И ТЕХНОЛОГИЯ ИХ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ

Корончик Д. Н.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

denis.koronchik@gmail.com

В работе описывается пример использования и реализации модели пользовательского интерфейса интеллектуальной системы. Поэтапно описывается процесс получения запросов от пользователя, и вывод ответов на них. Все описанные в статье принципы и положения использованы при разработке сайта ims.ostis.net.

Ключевые слова: модель пользовательского интерфейса, машина обработки знаний, технология проектирования пользовательских интерфейсов, пользовательский интерфейс интеллектуальной системы.

Введение

В настоящее время существует много подходов, библиотек и инструментов для создания пользовательских интерфейсов. Среди них можно выделить следующие: Qt [Qt, 2012], Adobe Flex [Flex, 2012], Microsoft Silverlight [Silverlight, 2012]. Многие из них ориентируются на компонентное проектирование, но, к сожалению, стыковка компонентов между собой требует высокого уровня профессионализма от разработчика.

В большинстве своем пользовательские интерфейсы современных систем являются сложными, что обусловлено сложностью самих систем. Основной проблемой в них является то, что пользователю, имеющему низкий уровень квалификации, сложно разобраться. Это в свою очередь уменьшает количество пользователей и снижает эффективность их эксплуатации. Одним из важных факторов в этом является то, что все такие пользовательские интерфейсы имеют различные принципы организации. При переходе от одной системы к другой пользователю необходимо затратить некоторое время, чтобы освоить новую систему. Кроме того для их освоения необходимо читать большое количество документации.

В настоящее время разработка пользовательских интерфейсов отнимает большую часть времени затрачиваемого на разработку всей системы. Проектирование пользовательских интерфейсов интеллектуальных систем является более сложной

задачей по ряду причин, что делает разработку технологии для их проектирования более актуальной. Сложность разработки таких интерфейсов обусловлена требованиями, которые предъявляются к ним:

- должны отображать различные виды знаний (при прочих равных условиях, чем больше различных видов знаний имеется в базе знаний системы, тем она интеллектуальней);
- должны обеспечивать возможность пользователю ставить перед системой существенно большее количество задач (в том числе и свободно конструируемых);
- должны как можно более четко отражать семантику предметной области в рамках которой происходит общение с пользователем.

Пользовательский интерфейс является единственным способом взаимодействия пользователя с программной системой. Поэтому они должны быть достаточно простыми и легкими в освоении [Поспелов, 1989]. Предлагаемая в данной работе технология направлена на решение описанных выше проблем.

1. Общие положения

В рамках проекта OSTIS ведется разработка семантической технологии проектирования пользовательских интерфейсов интеллектуальных систем. Более подробно с её описанием можно ознакомиться по ссылке [Корончик, 2012]. В данной статье мы рассмотрим важнейшую часть

технологии – унифицированную семантическую модель пользовательского интерфейса. Для этого приведем основные принципы, положенные в эту модель:

- пользовательский интерфейс рассматривается как специализированная интеллектуальная система, которая направлена на получение сообщений от пользователя и вывода ему ответов системы. Другими словами, основной задачей пользовательского интерфейса является перевод сообщения от пользователя, полученного на некотором внешнем языке, на язык понятный системе, а также вывод ответа системы на некотором внешнем языке, понятном пользователю;
- в основе графических интерфейсов лежит SCg-код (Semantic Code graphical – который является одним из возможных способов визуального представления текстов SC-кода) [Голенков и др, 2001]. Объекты, отображаемые на экране с помощью SCg-кода, будем называть sc.g-элементами. Основным принципом, положенным в его основу, является то, что все изображенные на экране объекты (sc.g-элементы), в том числе и элементы управления, являются изображением узлов семантической сети. Другими словами каждому изображенному на экране объекту соответствует узел в семантической сети (база знаний);
- выделение семантики в пользовательских действиях, с последующим анализом, а также их унификация и четкая типология

Так как пользовательский интерфейс представляет собой интеллектуальную систему, построенную с помощью семантической технологии компонентного проектирования интеллектуальных систем, то он точно так же как и любая другая система строится с использованием компонентов.

Выделены следующие типы компонентов:

- трансляторы с текстов различных внешних языков в тексты SC-кода;
- трансляторы с текстов SC-кода в тексты различных внешних языков;
- компоненты вывода информационных конструкций пользователю;
- компоненты ввода информационных конструкций;

Каждый компонент пользовательского интерфейса состоит из некоторой базы знаний необходимой для его работы и набора sc-агентов [Голенков, 2012].

2. Применение на практике

Опишем применение описываемых моделей на примере разработки сайта **ims.ostis.net**. Основное назначение сайта в текущий момент – это навигация по некоторому объему хранимых знаний. Взаимодействие пользователя с сайтом осуществляется через обмен сообщениями. Общение происходит с помощью двух внешних

языков: SCg, SCn [Голенков, 2012]. Эти языки могут быть расширены, так как сам сайт использует описываемый в модели компонентный подход.

Чтобы лучше понять каким образом происходит обмен сообщениями между пользователем и системой, рассмотрим всю цепочку событий, которые происходят от начала формирования запроса пользователя до вывода ответа на экран.

Вопрос пользователя системе формируется следующим образом:

1. пользователь указывает аргументы вопроса. Это делается с помощью левой клавиши мыши с одновременно зажатой клавишей Ctrl (без неё, ссылка происходит инициирование вывода полной семантической окрестности указанного объекта). Пользователь может формировать аргументы в любом порядке, важно, что количество и очередность аргументов зависят от команды, которую пользователь хочет инициировать;
2. пользователь выбирает команду из главного меню, и левым щелчком мыши инициирует её. Если же пользователь удерживает Ctrl при клике по команде в меню, то команда не будет инициирована и попадет в список аргументов. Другими словами пользователь может задавать вопросы и к самим элементам управления.

Предположим, что пользователь указал один аргумент “объектX” и инициировал команду “Поиск всех выходящих дуг из указанного объекта”. После инициирования команды в базе знаний генерируется конструкция, представленная на рисунке 1. Это еще не сам запрос поиска выходящих дуг, а лишь некоторая первичная информация о том, что пользователь указал некоторые аргументы и инициировал команду. На это событие реагирует sc-агент, который формирует уже сам запрос.

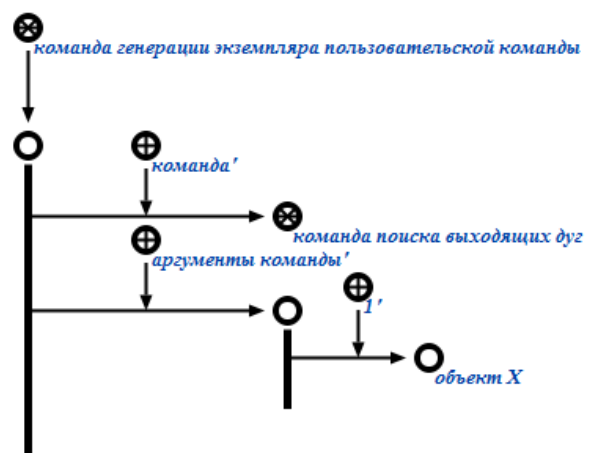


Рисунок 1 – Конструкция, генерируемая пользовательским интерфейсом при инициировании команды

Важно отметить, что команды главного меню – это классы команд, а не конкретные их экземпляры. Они делятся на два типа: атомарные и неатомарные команды. Неатомарные команды – это команды, инициирование которых аналогично запросу декомпозиции самих себя (раскрывающиеся пункты

меню). Атомарные команды в базе знаний описывают некоторую обобщенную формулировку, по которой потом генерируется некоторый запрос. Примером такой команды может быть Команда поиска выходящих дуг, её описание показано на рисунке 2.

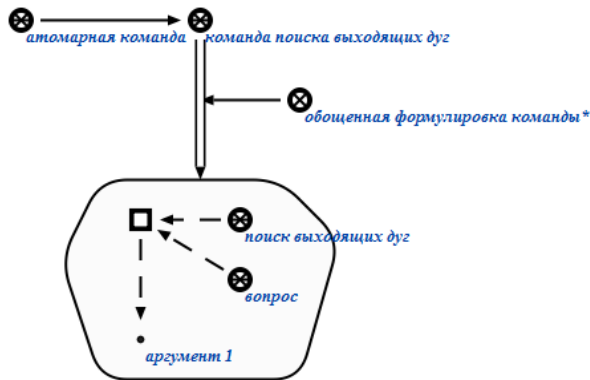


Рисунок 2 – Пример описания атомарной команды

Конкретный запрос пользователя генерируется на основании обобщенной формулировки команды. За этот процесс отвечает sc-агент, условием инициирования которого является появление конструкции представленной на рисунке 1. В результате работы этого sc-агента в базе знаний формируется конструкция, которая представлена на рисунке 3. Важно то, что при переходе на другую платформу или реализацию пользовательского интерфейса, к примеру, не только в web-ориентированном но и в desktop варианте, нам необходимо реализовать лишь генерацию первичной команды (рисунок 1). Так как второй этап уже не работает с внешней средой.

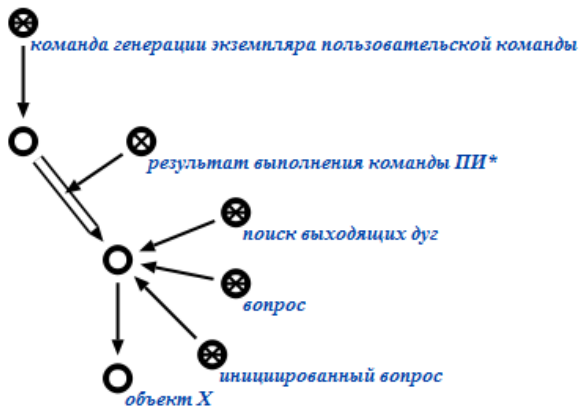


Рисунок 3 – Пример иницированного вопроса

На появление иницированного вопроса в базе знаний (рисунок 3) реагируют агенты машины обработки знаний, которые берутся за поиск ответа на данный вопрос. Параллельно с этим на появление результата генерации экземпляра команды реагирует sc-агент пользовательского интерфейса, который указывает автора вопроса, и внешние языки, на которые должен быть выведен ответ на данный вопрос. В дальнейшем эта информация понадобится, чтобы принять решение о необходимости трансляции ответа на внешний

язык и последующий его вывод пользователю. В результате его работы в базе знаний появляется конструкция, представленная на рисунке 4.



Рисунок 4 – Результат работы sc-агента, который указывает дополнительную информацию о вопросе

Сейчас все что остается пользовательскому интерфейсу, так это дождаться когда ответ вопрос будет обработан. Когда появляется информация о том, что обработка вопроса завершена (вопрос удален из множества иницированных вопросов и помещен во множество завершенных вопросов), снова запускается sc-агент пользовательского интерфейса, который находит автора вопроса. Если автором вопроса является пользователь, то запускается процесс вывода ответа. Сначала в базе знаний генерируется команда, которая иницирует трансляцию ответа на указанные внешние языки (рисунок 5).

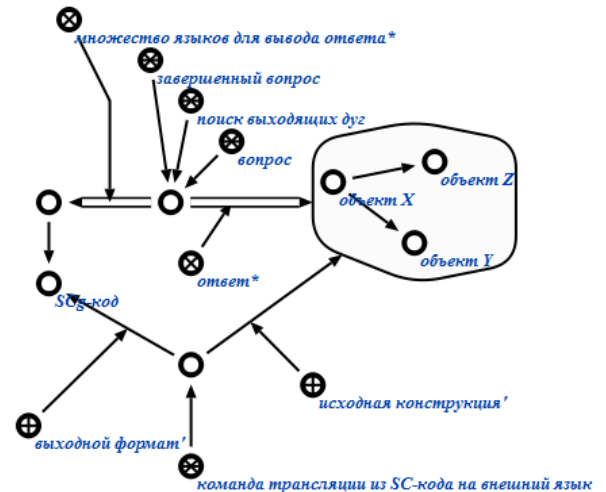


Рисунок 5 – Пример команды трансляции ответа на внешний язык

На появления такой команды реагирует sc-агент компонента трансляции из SC-кода на внешний язык (в нашем случае это компонент трансляции в SCg-код). После завершения его работы в базе знаний появляется конструкция, которая связывает ответ с некоторым файлом, в котором этот ответ записан на внешнем языке (рисунок 6), а команда трансляции помещается во множество завершенных команд. На данном этапе система готова

представить пользователю ответ на его вопрос в том виде в котором того просил пользователь.

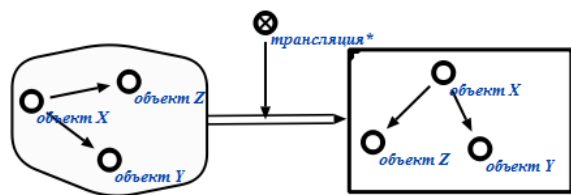


Рисунок 6 – Результат выполнения трансляции

После завершения трансляции запускается “эффекторный” sc-агент. Он выводит содержимое файла, в котором содержится ответ на вопрос, пользователю, который этот вопрос задавал.

В ответе системы, который

ЗАКЛЮЧЕНИЕ

Разработанный прототип работает в среде интернет. Взаимодействие с пользователем осуществляется по описанным выше принципам. Данный подход показал свою жизнеспособность и продемонстрировал некоторые преимущества.

К примеру, из-за слабой зависимости между подсистемами (зависимость лишь по топологии конструкций в базе знаний), они могут разрабатываться параллельно. Это дает существенное сокращение сроков разработки.

Интерфейс легко расширяем путем добавления новых компонентов, причем все компоненты могут быть добавлены динамически во время работы системы.

Возможность задавать вопросы к элементам управления позволяет существенно сократить время на освоение системы, за счет того, что пользователь может узнавать у системы как пользоваться теми или иными командами.

В процессе разработки системы, разработчики могут использовать уже имеющиеся в библиотеке компоненты, что сокращает сроки разработки, также пополнять библиотеку компонентов новыми компонентами, что позволяет технологии развиваться.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Голенков и др, 2001] Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков, [и др]; – Мн. : БГУИР, 2001

[Голенков, 2012] Графодинамические модели параллельной обработки знаний / В. В. Голенков, Н. А. Гулякина // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» - Минск, 2012

[Корончик, 2012] Семантические модели мультимодальных пользовательских интерфейсов и семантическая технология их проектирования / Д. Н. Корончик // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» - Минск, 2012

[Поспелов, 1989] Поспелов Д.А. Интеллектуальные интерфейсы для ЭВМ новых поколений// Электронная

вычислительная техника. Сборник статей. Вып.3. - М.: Радио и связь, 1989. - С.4-20.

[Flex, 2012] Free, open-source framework | Adobe Flex [Электронный ресурс]. – 2012. – Режим доступа: <http://www.adobe.com/products/flex.html>. - Дата доступа: 05.10.2012

[OSTIS, 2012] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2012. - Режим доступа: <http://ostis.net>. – Дата доступа: 11.10.2012

[Qt, 2012] Qt project [Электронный ресурс]. – 2012 – Режим доступа: <http://qt-project.org/>. – Дата доступа: 02.10.2012

[Silverlight, 2012] Microsoft Silverlight [Электронный ресурс]. -2012 – Режим доступа: <http://www.microsoft.com/silverlight/>. – Дата доступа: 03.10.2012

UNIFIED SEMANTIC MODELS OF USER INTERFACE FOR INTELLIGENT SYSTEMS AND TECHNOLOGY FOR THEIR DEVELOP

Koronchik D. N.

*Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus
denis.koronchik@gmail.com*

This article describes usage of semantic user interface model in develop process.

INTRODUCTION

The development of software engineering environments has had a long and close relationship with the development of advanced user interface technologies. There are many technologies for user interface development. The only way to interact with machine is user interface. It would be impossible to interact with systems without it.

MAIN PART

The main purposes of semantic user interface technology are: to decrease time of user interface development by using components; lower level of developer qualification; to decrease requirements to user's qualification; make user interface integration with intelligent system much simpler. The offered technology based on a four principles: unified semantic model lies in a basis of all designed user interfaces; user interfaces designed from components; develop process supports by intelligent tools; technology includes help system, that helps developers in design process.

The offered technology includes: semantic model of user interfaces, components library, design tools, design technique, technique to study design process, help system for technology

CONCLUSION

Principles and techniques, described in this paper, was used to create site [ims.ostis.net](http://ostis.net).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8:004.5

ПРОБЛЕМЫ ПРОМЫШЛЕННОГО ПРОЕКТИРОВАНИЯ ИНТЕРФЕЙСОВ

Ветров Ю.А.

*Компания Mail.Ru Group,
г. Москва, Россия
jvetrau@gmail.com*

Потребность в хорошо продуманных пользовательских интерфейсах появилась вместе со сложной техникой, которая требовала эффективной и безопасной работы оператора с ней. Сейчас дисциплина проектирования пользовательских интерфейсов востребована особенно активно. Ведь на высоко конкурентном рынке физических и цифровых потребительских продуктов хороший дизайн и интерфейс – один из ключевых критериев коммерческого успеха. Компания Mail.Ru Group работает именно на таком рынке и в статье будет рассказано о проблемах и задачах, с которыми сталкивается компания при создании дизайна крупных сервисов.

Ключевые слова: проектирование пользовательских интерфейсов и юзабилити; user experience; процесс; методы и практики

ВВЕДЕНИЕ

Термин «эргономика» ввел в современный лексикон польский ученый и общественный деятель Войцех Ястребовский в 1857 году. Индустриальная эпоха требовала особого внимания к эффективности труда, что отразилось, в том числе, в теории «научной организации труда» Фредерика Тейлора и других подходах к научному управлению. Хотя термин это греческий (Εργον означает «работа», а Νόμος – «естественные законы») и эллины использовали эргономические принципы при создании инструментов, рабочих мест и процессов. Например, Гиппократ описывал правильное рабочее место хирурга. Египетские династии также следовали принципам эргономики [Wikipedia, 2013a].

Первая, и в особенности Вторая Мировая Война, дали новый толчок развитию дисциплины. В этот период появилось много сложной военной техники, обучать работе с которой требовалось все новые и новые группы рекрутов. И даже опытные солдаты, постоянно находясь в стрессовой ситуации, совершали ошибки. Причем их цена стала достаточно высокой – гибель в бою. Это дало толчок к множеству исследований и усовершенствований в механизмах управления боевыми машинами.

Но параллельно с этим появился новый уровень проблем. Образовались сложные социотехнические системы, в которых пользователи разных машин должны были решать комплексные совместные

задачи. Например, оператор радарной станции должен выбрать объекты для атаки и передать эту информацию пилоту, который должен уничтожить цель. Такие задачи потребовали развития компьютерной техники, только-только появившейся к концу войны. А это означало, в том числе, и работу с пользовательскими интерфейсами [Wikipedia, 2013b].

Пользу компьютеров в бизнесе заметили достаточно быстро – в 1951 появился первый коммерческий продукт LEO I, помогавший вести бухгалтерию сети чайных кафе в Великобритании [Wikipedia, 2013c]. Последующие десятилетия показали стремительное проникновение профессиональной, а потом и домашней вычислительной техники. На рынок попадало все больше и больше аппаратных и программных продуктов, боровшихся теперь уже за коммерческий успех. Для этого их создателям важно было учитывать, что массовый пользователь не разбирается в сложной новой технике и работа с ней должна быть максимально простой для него, а обучение – не слишком затратным. К пониманию этого пришли только в 1980-х годах, так что вместе с удешевлением стоимости компьютеров это стало одним из ключевых факторов их массового распространения.

Эти же факторы – более доступный по цене, более простой в использовании, более компактный – стали причиной текущей мобильной революции, когда количество продаваемых смартфонов превысило количество персональных компьютеров.

При том, что по своим возможностям решения повседневных задач они стали вполне сопоставимы. Эти же три фактора – цена, простота, компактность (каждый из которых делает технологии доступнее) станут толчком к появлению новых компьютерных парадигм и в будущем. За второй фактор как раз и отвечает проектирование и дизайн пользовательских интерфейсов.

1. Текущие задачи и проблемы дисциплины

1.1. Пользовательский интерфейс приложений

Основная масса современных задач по проектированию и дизайну интерфейсов (здесь и далее термин «интерфейс» используется в понимании «пользовательский интерфейс») является проектами по созданию приложений на базе одной из программных платформ. Наиболее востребованные платформы – мобильные и настольные ОС (операционные системы), веб (браузер). В последние годы модель сторонних приложений пришла и в другие сферы, которые ранее были закрытыми – «умные» телевизоры, автомобили, бытовая техника. Поэтому пространство для работы современного проектировщика и дизайнера интерфейсов активно расширяется как за счет создания новых продуктов, так и за счет переноса продукта на новые платформы. Эта тенденция приводит к поиску новых наукоёмких и высокотехнологичных подходов для создания интерфейсов.

В работе над приложением проектировщик и дизайнер сталкиваются с такими проблемами как:

- Поиск подходящей ниши для нового продукта. Конкуренция высока и все сложнее найти еще нерешенные проблемы потребителей, которые позволят стать продукту востребованным успешным.
- Определение оптимального набора функций продукта и представление их в интерфейсе таким образом, чтобы работа пользователя с ним была до нужной степени комфортной.
- Понимание, насколько созданный продукт удовлетворяет задачи и потребности пользователей. Регулярное отслеживание проблемы и оперативно корректирование продукта с учетом этих задач и потребностей.
- Обеспечение переноса продукта на новые платформы более дешевым способом, также как и поддержки всего портфеля продуктов. При этом не в ущерб потребительским свойствам продукта.

1.2. Программные платформы

Более сложной задачей является создание самой программной платформы. Еще десять назад эта ниша казалась закрытой – потребительский рынок настольных компьютеров был поделен между Windows и Mac OS, а мобильные ОС имели

достаточно примитивный вид. Но сейчас, во многом благодаря тому, что был найден успешный механизм распространения приложений через официальный унифицированный интерфейс электронного магазина, количество новых платформ активно растет. В работе над платформой проектировщик и дизайнер решают такие задачи как:

- Разработка модели работы платформы и сторонних приложений в ней, которая позволит им взаимодействовать друг друга на достаточно гибком уровне. Модель платформы как простой «прокладки» для запуска сторонних приложений, которые никак не связаны друг с другом, устарела и имеет малую ценность для потребителя.
- Определение принципов построения интерфейса сторонних приложений. Они должны обеспечить общие правила, достаточно простые, чтобы порог вхождения был низким, и жесткие, чтобы сохранить единство внешнего вида и поведения интерфейсов сторонних производителей.
- Обеспечение обновления версий платформы так, чтобы оно минимально сказывалось на работоспособности уже созданных приложений, но при этом не сковывало старыми неэффективными решениями.

1.3. Экосистемы

При успешном развитии платформы она создает экосистему вокруг себя. Такая экосистема включает устройства, приложения и сервисы, которые в связке формируют значительно более ценный для пользователя продукт. При этом сервис – еще одна сфера задач проектировщика, которая активно набирает обороты в последние несколько лет. Ведь в хорошем продукте важны не только проработанные основные сценарии использования, но и весь жизненный цикл владения им. А это означает обращения в службу поддержки, решение проблем с резервированием данных, обновлением платформы и множество других второстепенных сценариев. В работе над экосистемой проектировщик и дизайнер сталкиваются с рядом проблем, таких как:

- Необходимость понимания всего жизненного цикла работы пользователя с экосистемой и постоянно улучшать как всю связку платформа-приложение-сервис, так и отдельные ее части.
- Необходимость иметь влияние на рабочий процесс и продуктивные решения на разных уровнях работы компании. Создание экосистемы предполагает единое видение, которое проводят в жизнь все подразделения и специалисты компании.

Далее отдельно остановимся на некоторых из описанных проблем, с которыми наша компания сталкивается наиболее часто.

2. Подробнее о наиболее важных проблемах

2.1. Унификация интерфейса портфеля продуктов

Для компаний с большим количеством продуктов, связанных друг с другом общей рыночной нишей и пользовательской базой (к примеру, почтовый сервис может предлагать также календарь, сервис обмена мгновенными сообщениями, облачное хранилище документов и инструменты для их редактирования), возникает проблема единого подхода к созданию их интерфейсов. Это важно по нескольким причинам. Во-первых, унификация облегчает переход пользователя к новому продукту, ведь его поведение и внешний вид уже хорошо знакомы, а значит обучение работе с ним и изменение привычек – минимальны. Во-вторых, это облегчает разработку и поддержку новых продуктов, поскольку единые принципы построения интерфейсов экономят время на их проработке, а проверенные на практике решения снижают количество потенциальных проблем в будущем. В третьих, это усиливает бренд за счет того что продукты визуально являются продолжением друг друга.

С ростом количества мобильных платформ задача усложнилась, сделав портфель продуктов из простого списка – матрицей (пересечение продуктов и их версий для разных ОС). Это значит, что приложения для разных ОС должны быть узнаваемыми как визуально, так и по поведению интерфейса. С другой стороны, в реальной жизни пользователь владеет только одним смартфоном или планшетом, а значит, приложение должно быть понятным в первую для него и следовать правилам его платформы. Кроме того, для компании как бизнеса важно иметь собственный узнаваемый визуальный стиль, который будет поддерживать бренд. Все это ведет к конфликту – с одной стороны, нужно придерживаться официальных руководств по построению интерфейсов для платформы, ведь они гарантируют продукту, что пользователь умеет работать с ним благодаря единым правилам. С другой – нарушать их для того чтобы вести портфель продуктов было проще.

Решение проблем, связанных с унификацией, обеспечивает эффективное построение интерфейса портфеля продуктов, однако, по мнению Джефа Раскина, и приводит к ряду неудобств, связанных, в первую очередь, с выработыванием у пользователя «вредных» привычек, которые приводят к отторжению им нововведений в пользовательском интерфейсе [Раскин, 2004].

2.2. Методы разработки интерфейса для новых продуктов

Перед современной компанией, создающей коммерческие продукты и сервисы, все более остро стоит задача поиска новых рыночных ниш. Конкуренция, как всегда, усиливается и для успеха

бизнеса зачастую важны новые направления, в которых еще нет множества соперничающих компаний, зато есть перспективы роста. Это далеко не новая проблема и за последние десятилетия ее уже решали различными методами – например, с помощью ТРИЗ [Wikipedia, 2013d]. Одним из свежих подходов является «дизайн-мышление».

Дизайн-мышление – это появившийся в 80-х годах двадцатого века и ставший особенно актуальным в последнее время подход к созданию новых продуктов. Одной из главных его особенностей является применение для нетривиальных и плохо описанных задач, а также хорошо поставленный итеративный процесс их решения и постоянного пересмотра изначальных установок.

Хороший пример подхода в действии – история про африканские больницы. Команду дизайнеров попросили предложить недорогие в производстве поддоны для новорожденных, которые можно часто менять при ограниченном бюджете. Дизайнеры долго работали над задачей и в итоге создали действительно дешевое и надежное решение. А когда они приехали протестировать продукт, больница оказалась полупустой. Почему? Жители деревень часто не могли добраться до больницы. После пересмотра исходной проблемы был предложен другой продукт – недорогие одеяла, которые сохраняют младенцев в тепле даже в холодной пустыне [Embrace, 2013].

Пять этапов процесса дизайн-мышления очень похожи на те, из которых состоит проектирование интерфейсов – исследование, продумывание концепции, проектирование, прототипирование и тестирование. Но есть нюансы:

1. Акцент на постоянном пересмотре изначальной проблемы. Ее первое понимание, как правило, поверхностно, но только знание сути и всех нюансов позволяет найти лучшее решение.

2. Как следствие – стремление как можно быстрее ошибиться, получив первый вариант решения. Важно как можно чаще получать результат, для того чтобы можно было оценить его и доработать.

3. Четко прописанный набор методов и практик, которые позволяют пройти процесс решения задачи от начала до конца. Это значит, что у теории есть хорошая практическая база.

4. Ориентация на командную работу. Именно в команде проще охватить весь спектр возможных решений и более объективно оценивать выбранные варианты.

Множество управляемых мозговых штурмов с методами выбора лучших решений из массы предлагаемых. Так можно охватить все пространство решений, не ограничиваясь поверхностными идеями.

2.3. Промышленные методы оценки качества пользовательских интерфейсов

Оценка качества пользовательского интерфейса является важной и сложной научной задачей. В настоящее время разработан ряд подходов и методов. Их можно разбить на две большие группы:

- методы непосредственно тестирования интерфейса группой пользователей, представляющих целевую аудиторию продукта: юзабилити-тестирование, интервью, этнографические и дневниковые исследования;
- методы, основанные на формальных расчетах: модель скорости печати GOMS, экспертная оценка, опросы.

Специфика пользовательских интерфейсов такова, что обе группы подходов одинаково применимы для оценки качества любых пользовательских интерфейсов, в том числе и интеллектуальных.

Выбор подхода для оценки качества зависит от того, насколько осуществима оценка качества на той или иной стадии выполнения проекта и отведенного на такую оценку времени и бюджета. Однако не всегда приведенные подходы к оценке качества могут обеспечить эффективность интерфейса в глазах заказчика [Пономарёв, 2013].

Современный бизнес старается принимать решения на основе метрик и показателей эффективности. Неудивительно, что с этими же требованиями приходят и к специалистам по интерфейсам. Поэтому зачастую перед выделением средств на доработку существующего продукта необходимо аргументировать, что эти вложения дадут краткосрочную или долгосрочную материальную отдачу. А после завершения работ – доказать, что изменения дали ожидаемый эффект. Поэтому перед проектировщиком и дизайнером стоит задача оценки влияния своих решений на продукт.

Решить эту проблему помогают метрики. Их можно поделить на три группы:

- Заработанные деньги – повышение конверсии и ARPU (average revenue per user, средний заработок на одного пользователя);
- Экономленные деньги – сокращение времени выполнения ключевых операций с помощью продукта, экономия на службе поддержки;
- Нематериальная выгода – лояльность, количество рекомендаций друзьям и т.п.

Это позволяет перевести общение специалистов с представителями бизнеса в более предметное русло. И говорить не об артефактах («нам нужно нарисовать 5 прототипов»), а о целях и способах их достижения («чтобы повысить конверсию, нам нужно будет переработать ключевые страницы процесса покупки – это такие-то страницы, итого 5 штук. Для этого мы с помощью прототипов покажем подходящее решение проблемы, а дальше

будем участвовать в процессе внедрения и тестирования решений.»). Кроме того, метрики позволяют сформировать четкие критерии успешного завершения проекта.

Существуют формулы для расчета метрик и экономического эффекта от их достижения [Ветров, 2012]. Они позволяют понять, каких улучшений в интерфейсе необходимо добиться и за какой срок, чтобы получить отдачу от вложений. А также, какую сумму экономии или дополнительной прибыли может получить компания.

Однако такой подход имеет свои проблемы. Не всегда можно учесть влияние разных факторов на ключевые показатели эффективности продукта – интерфейс, маркетинг, специальные акции, изменение тарифов и т.п. На них могут также влиять сезонность, разная эффективность различных каналов привлечения клиентов, отложенный спрос. Но даже с этими ограничениями использование метрик позволяет найти общий язык с бизнесом и предлагать более качественные интерфейсные решения. Да и сами ограничения вызваны во многом текущим отсутствием этого общего языка.

2.4. Краткий обзор других проблем

В предыдущих разделах статьи описаны одни из самых крупных и актуальных задач, которые стоят перед современным проектировщиком и дизайнером интерфейсов. Не все они касаются такого специалиста напрямую, но, будучи частью современной команды по разработке продукта, он так или иначе сталкивается с ними, а значит должен хотя бы иметь представление об их сути. В разных компаниях структура команды и роль такого специалиста имеет разные виды. Поэтому и набор обязанностей и сфер ответственности может различаться, иногда включая в себя самые широкие навыки и знания.

Ниже перечислены другие проблемы проектирования современных интерфейсов, в числе которых интерфейсы, тесно связанные с направлениями исследований по искусственному интеллекту, [Поспелов, 1989] которые могут попасть в набор навыков современного специалиста по интерфейсам.

"Второй экран". С ростом количества различных устройств у одного пользователя возник интересный сценарий использования. При просмотре ТВ-каналов он зачастую обращается к смартфону или планшету для того чтобы уточнить показанную в фильме или передаче информацию, либо просто отвлечься в малоинтересный момент. Это значит, что интерфейсы телевизора и сопроводительного устройства дополняют друг друга и на этом стыке рождаются интересные задачи.

Вычленение смысла из большого объема данных. Одна из главных тенденций современных технологий – хранение огромных наборов данных по любому вопросу, как в общедоступном виде, так

и в рамках конкретного продукта. Однако эта информация имеет мало пользы, если не знать, как строятся взаимосвязи внутри нее. Наиболее простое решение – выделить набор ключевых показателей и следить за их изменениями. Но еще лучше связать события из нескольких разных систем. Например, в зависимости от размера и характера денежных трат клиента банка по дебетовой карте ему могут быть предложены дополнительные продукты и услуги.

В рамках этого направления просматривается необходимость использования специальных моделей представления знаний, которые обеспечивают с одной стороны возможность представления сложноструктурированной информации, а с другой стороны обеспечивают понятность этого представления.

Персональные помощники. На потребительском рынке этой темой активно интересовалась компания Microsoft в начале 1990-х, однако после серии провальных продуктов (MS Bob, MS Office Clippy) в отрасли надолго сформировалось негативное отношение к этому направлению. Хотя потребность в информационных системах поддержки принятия решений существует давно и в узкоспециализированных областях они успешно работают. Несколько лет назад начался новый этап в популяризации персональных помощников с выходом Google Now для платформы Android и Apple Siri для iOS. Как и в случае с вычлениением смысла из большого объема данных, такие решения позволяют пользователю получать больше полезной информации и при этом меньше взаимодействовать с интерфейсом.

Особо эффективными в рамках данного направления в последнее время оказываются естественно-языковые и речевые интерфейсы. Такие интерфейсы обеспечивают не только естественно-языковой или речевой ввод информации в систему, но и вывод этой информации в речевом или естественно-языковом виде [Лобанов, 2011].

Естественные интерфейсы. Изначально взаимодействие с компьютером строилось исходя из того, что пользователю необходимо приспосабливаться к набору абстракций, понятных ранним алгоритмам. Однако по мере совершенствования методов вычисления, ввода и вывода информации, уровень абстракции снижался – от двоичных данных перфокарт к набору команд на почти естественном языке. Затем – к использованию метафор реального мира, тактильному и голосовому управлению. Сейчас на массовом рынке доступны датчики, позволяющие управлять техникой бесконтактно, с помощью жестов или движения глаз. Это дает новые возможности специалистам по интерфейсам. И новые сложности – необходимо снова устанавливать общепринятые правила их создания и учить пользователей взаимодействию с ними.

Проецируемые интерфейсы. Сейчас для работы с интерфейсом требуется

специализированное устройство. Но что если оно не понадобится и рабочей поверхностью может быть что угодно – письменный стол, холодильник, коробка из под обуви? А если такой проекции и вовсе не нужна поверхность? Активные исследования и пробные запуски в этой области ведутся давно и в ближайшие годы на рынке появятся массовые продукты. Сложно судить, каким именно будет взаимодействие с ними на практике и какую нишу они займут. Однако это одно из наиболее вероятных направлений работы специалистов по интерфейсам на ближайшие годы.

Дополненная реальность и носимые интерфейсы. Один из основоположников видения современных компьютеров, Ванневар Буш, описывал их как дополнение способностей и мозга человека [Memex, 2013]. Современная концепция дополненной реальности пытается решить эти же проблемы, усиливая возможности зрительного восприятия. С помощью видеокамеры телефона, специальных очков или контактных линз картина мира, видимая глазом человека, расширяется вспомогательной информацией о его объектах. Текущее применение технологии ограничено узкоспециализированными задачами (например, автомеханики), либо сводится к навигационным и развлекательным сервисам. Во многом это связано с техническими ограничениями и после их решения сфера использования дополненной реальности может расширяться.

Эмоциональные интерфейсы и нейромаркетинг. Благодаря тому, что дисциплина проектирования компьютерных интерфейсов развивается достаточно давно и ее важность доказана бизнесу, крупные продукты уже решили базовые проблемы во взаимодействии и имеют сопоставимые по качеству дизайн-решения. На высоко конкурентных рынках это приводит к вопросу о том, что выделить свое предложение все сложнее. Поэтому многие компании обращают особое внимание на качество сервиса – как во взаимодействии с обслуживающим персоналом, так и самим продуктом. И здесь очень важна эмоциональная составляющая – как именно потребитель воспринимает продукт и работу с ним, понимание психологии его поведения и принятия решений. Сейчас появляется все больше оборудования, позволяющего изучать эти процессы. А значит и возможностей для специалиста по интерфейсам влиять на них с помощью тех или иных решений. Это, в свою очередь, затрагивает вопросы этики – корректно ли влиять на поведение пользователя?

ЗАКЛЮЧЕНИЕ

Одно из названий дисциплины, занимающейся пользовательскими интерфейсами – человеко-компьютерное взаимодействие. Технологическое развитие показывает, что с каждым годом количество различных устройств, в той или иной мере являющихся компьютерами, стремительно

растет. При этом расширяются сферы их применения, меняются способы взаимодействия. Это значит, что задач для специалиста по интерфейсам в будущем станет только больше. А с их изменением наверняка поменяется и суть самой профессии. И лучший способ не потерять квалификацию – быть в курсе текущих и будущих проблем, стоящих перед нами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Ветров, 2012] Vetrov, Yu. A. How to Calculate the ROI of UX Using Metrics// UXMatters [Электронный ресурс] – 2012. Режим доступа: <http://www.uxmatters.com/mt/archives/2012/07/how-to-calculate-the-roi-of-ux-using-metrics.php>

[Лобанов, 2011] Лобанов, Б.М. Методы семантического анализа для построения голосовых интерфейсов: синтез речи / Б. М. Лобанов // Материалы международной научно-технической конференции OSTIS-2011, – 2011 С. 387-394.

[Поспелов, 1989] Поспелов, Д. А. интеллектуальные интерфейсы для ЭВМ новых поколений // Электронная вычислительная техника. Сборник статей. Вып.3. – М.: Радио и связь, 1989. – С.4–20.

[Пономарёв, 2013] Пономарёв, И.А. Методы оценки качества пользовательского интерфейса [Электронный ресурс] – 2012. Режим доступа: <http://it-claim.ru/Library/Books/ITS/wwwbook/ist6/ponomarev2/ponomarev2.htm>

[Раскин, 2004] Раскин Д. Интерфейс: новые направления в развитии компьютерных систем: пер. с англ. /Д. Раскин.– СПб: Символ-Плюс, 2004.–272 с.

[Wikipedia, 2013a] Эргономика [Электронный ресурс] – 2012. Режим доступа: <http://en.wikipedia.org/wiki/Ergonomics>

[Wikipedia, 2013b] Социотехнические системы [Электронный ресурс] – 2013. Режим доступа: http://en.wikipedia.org/wiki/Sociotechnical_system

[Wikipedia, 2013c] LEO Computer [Электронный ресурс] – 2012. Режим доступа: http://en.wikipedia.org/wiki/LEO_computer

[Wikipedia, 2013d] Теория решения изобретательских задач [Электронный ресурс] – 2013. Режим доступа: http://ru.wikipedia.org/wiki/Теория_решения_изобретательских_задач

[Embrace, 2013] Программа Embrace [Электронный ресурс] – 2013. Режим доступа: <http://embraceglobal.org/>

[Memex, 2013] Memex [Электронный ресурс] – 2013. Режим доступа: <http://en.wikipedia.org/wiki/Memex>

CURRENT PROBLEMS OF INDUSTRIAL USER EXPERIENCE DESIGN DISCIPLINE

Vetrov Y.A.

Mail.Ru Group, Moscow, Russia

jvetrau@gmail.com

The need in well thought-out user interfaces was born with complicated machinery that required safe and efficient work of an operator. Now the discipline of user experience (UX) design is in especially high demand. Thanks to the high competition in a consumer market for physical and digital products that requires decent UX as one of key success factors. Mail.Ru Group works on a market like this and I'll write about problems and tasks we deal with while designing large web services.

INTRODUCTION

The term “Ergonomics” was brought to the modern language by Wojciech Jastrzebowski in 1857 – industrial era demanded high efficiency of work force.

Although the term has Greek origin (Εργον means “work” and Νόμος means “natural laws”). This science had a huge boost during WWI and WWII with the introduction of complex machinery – it needed to train new pilots quick and the price of error became incredibly high. That led us to socio-technical systems and commercial usage of computers so they've become a commonplace in our homes and offices. However, Human-Computer Interaction has got attention only in 1980s and this was the beginning of the road to the broader User Experience discipline.

MAIN PART

Modern user experience designers mostly deal with applications for one of desktop and mobile OSes or the web. Although the new areas of interest are growing now in automotive industry, smart TVs and even kitchen appliances. We also have a more complex task – the design of platforms and OSes themselves. And even more epic mission in a whole ecosystem design.

There are several big problems large companies are dealing with:

- unification of a product portfolio;
- developing user experiences for new products;
- industrial methods of user experience quality evaluation.

And a lot of new user experience paradigms and problems are developing now:

- second screen;
- getting sense from big flows of data;
- personal assistants;
- natural user interfaces;
- projected user interfaces;
- augmented reality and wearable interfaces;
- emotional interfaces and neuromarketing.

CONCLUSION

We see an always increasing growth of number and types of computing devices every year. They're getting into new areas and interaction methods are constantly changing. This means user experience designers are becoming assigned to new tasks and new responsibilities. And they need to be in touch with all these innovations to stay highly qualified and valued.



УДК 681.3.06

ИНТЕЛЛЕКТУАЛЬНАЯ ВИЗУАЛИЗАЦИЯ ПРИ АВТОМАТИЧЕСКОМ КАРТОГРАФИРОВАНИИ

Беляков С.Л., Белякова М.Л., Савельева М.Н.

Южный федеральный университет,

г. Таганрог, Россия

Beliacov@yandex.ru

asni@fep.tsure.ru

marina.n.savelyeva@gmail.com

В работе рассматривается модель визуализации электронных карт с картографическими объектами, которые получены путем распознавания содержимого мультимедиа-источников информации. Предлагается формализация, основанная на использовании функции информативности картографических изображений. Анализируются стратегии визуализации для ряда практически важных ситуаций картографического анализа.

Ключевые слова: интеллектуальные геоинформационные системы, визуализация, картографирование.

ВВЕДЕНИЕ

Настоящая работа ставит своей целью анализ модели интеллектуального управления процессом картографической визуализации не полностью определённых ситуаций. Неполнота касается пространственных, временных координат и семантических атрибутов. Интеллектуальное управление, основанное на знаниях, представляется рациональной альтернативой традиционным детерминированным процедурам управления визуализацией.

Следует предположить, что совершенствование систем распознавания изображений, текстов и речи в недалёком будущем приведёт к появлению новой функции геоинформационных систем – автоматическому картографированию не полностью определённых ситуаций. Соответствующая подсистема будет представлять в образно-знаковом виде объекты и явления, описание которых получено из источников информации в сети Интернет. Очевидно, что требуется особый подход к процедурам визуализации и оценки качества подобных карт. Его основой должно стать использование принципов интеллектуализации информационных систем.

1. Управление визуализацией в ГИС

Рассмотрим модель использования электронной карты. Решая прикладную задачу с помощью ГИС, пользователь реализует процедуру

картографического анализа [Берлянт, 1986]. В основе процедуры лежит создание рабочей области общей карты, составляющую информационную основу ГИС. Под рабочей областью карты $\Omega = \{ \omega_1, \omega_2, \dots, \omega_n \}$, состоящей из картографических объектов ω_i , понимается подмножество объектов $m_w \subseteq \Omega$, описывающих фрагмент карты с границами

$$L_w = \{ S_w, T_w, C_w, E_w \},$$

где S_w – пространственная, T_w – временная, C_w – семантическая (список типов объектов), E_w – прагматическая границы (описание границ применимости). Рабочая область порождается последовательностью

$$Q_i(X_S, X_T, X_C, X_E), i = \overline{0, N},$$

запросов клиента серверу ГИС, где X_* – соответственно, пространственные, временные, семантические и прагматические параметры запроса. $Q_i(X_S, X_T, X_C, X_E)$ – это предикат запроса, который в современных ГИС строится пользователем с помощью диалоговых меню либо напрямую задаётся как выражение на языке SQL. С содержательной точки зрения, запросы обеспечивают манипулирование картографическим изображением (зуммирование, панорамирование, наложение слоев, манипулирование видами) и синтез картографических объектов (зон, путей,

распределений).

Структура рабочей области $m_w \subseteq \Omega$ может быть представлена в виде объединения двух множеств:

$$m_w = B \cup E,$$

$$B \subseteq \Omega : \forall \omega_i \in B \Rightarrow \exists Q_j (X_s, X_T, X_C, X_E) = true,$$

$$j = \overline{1, N}, i = \overline{1, |\Omega|}, B \cap E = \emptyset, E \subseteq \Omega.$$

Множество B – это остов запроса, определяющийся только предикатами запросов, E – окружение остова, т.е. подмножество картографических объектов, обеспечивающее смысловую целостность рабочей области. Окружение формируется применением экспертных правил построения образа ответа $K(B, \Omega)$ к остову запроса B :

$$\omega_i \in E \Rightarrow K(B, \Omega) = true, i = \overline{1, |m_w|}.$$

Экспертные правила $K(B, \Omega)$ отображают знания о том, как строятся полезные для решения задачи картографические изображения [Розенберг, 2010]. Применение знаний ведёт к сокращению избыточности картографических изображений и повышению их информативности.

Информативность $I(m_w)$ всякой рабочей области является относительной субъективно определяемой величиной. Её значение достоверно оценивается только в узких рамках конкретного класса прикладных задач, которые решаются определённой группой пользователей. Как мера полезности картографического изображения, информативность оценивается конечным множеством значений I_{c_i}

$$c_i \in C : I_{c_i} < I_{c_{i+1}}, i = \overline{1, |C|}, C \geq 2,$$

которые определяются на основе знаний, описываемых экспертными правилами $K_I(m_w)$. Обязательным требованием к знаниям является то, что функция информативности должна иметь единственный максимум при известном числе объектов рабочей области

$$m^* = |m_w|.$$

Данное требование отражает практически важный факт ограниченности возможностей визуального восприятия картографического изображения человеком. При небольшом числе графических объектов в рабочей области информативность невелика и растёт по мере добавления новых объектов. Однако, дальнейшее увеличение числа объектов усложняет изображение, из-за чего информативность начинает снижаться. Скорость роста информативности следует связывать с добавлением объектов из E , поскольку информативность остова B представляет тот начальный минимальный уровень, который рабочая область приобретает благодаря запросам пользователя.

Таким образом,

$$m^* \neq |m_w| \Rightarrow I(m_w) < I_{max},$$

где I_{max} – максимально возможный уровень информативности рабочей области m_w . Заметим, что, в отличие от оценки $I(m_w)$, значение m^* является достаточно устойчивым для профессиональных групп пользователей. Задав эту величину в качестве одного из фактов в правилах $K_I(m_w)$, получаем окончательную «интеллектуальную» аппроксимацию кривой информативности.

Экспертные знания в виде правил $K(B, \Omega)$ и $K_I(m_w)$ воспроизводят «разумную» стратегию визуализации картографических изображений. Отличие описанной «интеллектуальной» визуализации от традиционной – в поддержке субъективной полезности картографических изображений. Визуализируются не только объекты, отвечающие предикату запроса, но и все те, которые наполняют картографическое изображение смыслом.

Управление визуализацией заключается в решении оптимизационной задачи

$$\begin{cases} I(m_w) \rightarrow max, \\ |m_w| \leq m^*, \\ m_w : Q_i(X_s, X_T, X_C, X_E) = true, i = \overline{1, N} \end{cases}$$

Интеллектуальная ГИС, таким образом, контролирует сложность рабочей области, добавляя или удаляя картографические объекты.

2. Визуализация при наличии дефектов

Оперативность отображения реального мира в картографическом виде предполагает запись в базу картографических данных новых картографических объектов. Каждый из них является результатом работы некоторой программы распознавания и анализа снимков, текстов, сообщений, и т.д. Этот результат не может не содержать неопределённости. Следовательно, картографические объекты отображают неопределённые ситуации. Появление в картографической базе данных ГИС новых объектов, не согласованных с существующей картографической основой, требует специальных мер для поддержания информативности. Рассмотрим особенности управления визуализацией в этом случае.

Обозначим через $S = \{s_i\}$ множество картографических объектов, построенных на основании анализа состояния источников информации в Интернет. Предполагается, что ГИС обладает поисковой системой, способной находить информационные ресурсы для извлечения пространственных данных, и набором программ

распознавания ситуаций.

Информационная база ГИС дополняется множеством картографических объектов S :

$$\bar{\Omega} = \Omega \cup S,$$

причём, в отличие от любого объекта $\omega_i \in \Omega$, пространственно-временные координаты и семантические атрибуты $s_i \in S$ не являются абсолютно достоверными.

Простое добавление ситуаций в рабочую область и её визуализация с максимизацией информативности

$$I(B \cup E \cup S)$$

не может выполняться рассмотренным выше путем. Дело в том, что визуализация ситуаций $s_i \in S$ порождает дефекты картографического отображения, причиной которых является наложение изображений ситуаций $V(s_i)$ и объектов картографической основы:

$$V(s_i) \cap V(\omega_j) \neq \emptyset.$$

В качестве примера подобного дефекта можно привести отображение проложенных маршрутов сервисом Яндекс.Карты (см.рис.) Линия маршрута накладывается на названия улиц и номера домов, на контуры кварталов и зданий.



Рисунок 1 – Маршрут на карте

Как показал анализ, дефекты на картографическом изображении можно разделить на два класса:

- визуальные, создающие у пользователя неверное представление об объектах, изображения которых наложились друг на друга. При визуальном анализе карты возникает опасность того, что любой из участвующих в наложении объектов будет воспринят неверно. Заметим, что программно выявить наложение несложно, однако оценка результата восприятия образовавшегося наложения представляет собой самостоятельную задачу. Например, может быть использовано нормализованное представление картографических

объектов [Розенберг, 2010];

- процедурные, порождающие неверные результаты при выполнении процедур картографического анализа. В примере на рисунке возникает ошибка оценки топологического отношения «находиться рядом». Если выполняется запрос «найти все объекты, находящиеся рядом с маршрутом», результат будет неверным. Некоторые объекты находятся в отношении «пересекается». Обнаружение процедурных дефектов также представляет собой самостоятельную задачу и здесь не рассматривается. Важно то, что наложение изображений картографических объектов способно порождать ошибки в работе аналитических процедур.

Всякий дефект снижает информативность картографического изображения. Будем считать, что для устранения дефекта часть объектов удаляется. Пусть сформировано множество $D = \{d_i\}$ удалённых из рабочей области объектов. Тогда новая рабочая область описывается множеством

$$\tilde{m}_w = B \cup E \cup S \setminus D.$$

Условием того, что добавление неопределённых ситуаций даёт эффект, должно стать выполнение неравенства

$$I(B \cup E \cup S \setminus D) > I(B \cup E).$$

Данное неравенство является условием целесообразности включения в рабочую область объектов, описывающих неопределённые ситуации.

Анализ неравенства позволяет сделать ряд выводов, касающихся качества базовой картографической основы и стратегий отображения неопределённых ситуаций.

Если

$$I(B \cup E) = I_{max} \text{ и } |S| \neq |D|$$

то улучшить информативность не удастся, поскольку число m^* останется постоянным. Содержательно это означает, что базовая картографическая основа насыщена полезными данными до предела их человеческого восприятия. При этом уровень распознавания ситуаций таков, что автоматическое отображение объектов ухудшает качество рабочей области. Данный случай можно рассматривать как следствие недостаточной эффективности подсистемы распознавания.

Случай, когда

$$I(B \cup E) = I_{max} \text{ и } |S| \gg |D|$$

представляет эффективную систему распознавания. Согласованность автоматически картографируемых ситуаций с базовой картографической основой высока и это позволяет с помощью правил $K(B, \Omega)$ сконструировать новое изображение высокой информативности. В состав изображения

войдут более значимые классы объектов из множества S , за счёт чего будет обеспечена высокая содержательность изображения. К подсистеме распознавания не предъявляется никаких дополнительных требований, ее работа считается удовлетворительной.

Базовая картографическая основа может оказаться недостаточно информативной для решаемой задачи. Формально это означает, что

$$|m_w| \rightarrow 0.$$

В таком случае эффективное распознавание, означающее $|S| \gg |D|$, неизбежно повышает информативность. Это свидетельствует о безусловной целесообразности автоматического картографирования.

Практически важным случаем является «перегруженная» информацией рабочая область. Подобная ситуация возникает в продолжительных сеансах работы над прикладной задачей. Пользователь в таких случаях комбинирует слои и фрагменты карты с целью нахождения решения, намеренно игнорируя падение информативности из-за увеличения сложности рабочей области. Здесь имеет место «ручная работа» по созданию рабочей области, когда интеллектуальная система управления информативностью намеренно отключена. Формально этому случаю соответствует

$$I(B \cup E) < I_{max} \text{ и } |m_w| > m^*.$$

Тогда даже эффективное распознавание, предполагающее $|S| \gg |D|$, приводит к дальнейшему падению информативности. Соответственно, малоэффективное распознавание, для которого $|S| \approx |D|$, подобных последствий не вызывает. Данный частный случай может рассматриваться как условие отключения интеллектуального управления и переход на «ручной» режим построения изображений.

ЗАКЛЮЧЕНИЕ

В данной работе, таким образом, проанализирована модель, являющаяся основой геоинформационного сервиса для решения оперативных задач. Модель описывает стратегию конструирования рабочей области карты максимальной информативности. При этом учитывается ограниченная достоверность распознавания ситуаций, которое обеспечивается внешними автономными системами наблюдения реального мира.

Работа выполнена при поддержке грантов РФФИ 12-01-00032-а, 11-01-00011-а.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Берлянт, 1986] Берлянт А.М. Образ пространства: карта и информация. – М.: Мысль, 1986.

[Розенберг, 2010] Розенберг И.Н., Беляков С.Л. Программные интеллектуальные оболочки геоинформационных систем / Под редакцией Л.С.Берштейна. – М.: Научный мир, 2010.

INTELLECTUAL IMAGING IN AUTOMATIC MAPPING

Belyakov S.L., Belyakova M.L., Savelyeva M.N.

Southern Federal University, Taganrog, Russia

Beliacov@yandex.ru

asni@fep.tsure.ru

marina.n.savelyeva@gmail.com

We consider the rendering model of electronic maps with map objects, which are obtained by recognizing the content of multimedia information sources. It is proposed a formalization based on using the information content of cartographic images. Examines strategies for rendering a number of practically important cases of cartographic analysis.

INTRODUCTION

This work aims to analyze the model predictive control of cartographic visualization process is not completely certain situations. Incompleteness for spatial, temporal coordinates and semantic attributes.

MAIN PART

Describes the set-theoretic model of imaging that includes the task of maximizing the information content of the working area of electronic GIS maps. Information content of any work area is determined by the relative subjective value. Its value is reliably estimated only in the narrow framework of a specific class of applications that can be solved a specific group of users.

Introduce the concept of defect chart picture. Defects are divided into two classes. First class - it's visual, the user creating the wrong impression about the objects whose images are superimposed on each other. By visual analysis of the map there is a risk that any of the objects involved in the imposition will be perceived incorrectly.

The second class contains the procedural defects that generate incorrect results when performing procedures cartographic analysis.

CONCLUSION

In this paper we analyze the model, which is based on GIS services to solve operational problems. The model describes the design strategy of the working areas of the map as informative. This takes into account the limited accuracy of the recognition of a situation, which is provided by external autonomous observing systems of the real world.



УДК 004.932.2

ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ СЦЕНЫ ДЛЯ ОКРАШИВАНИЯ ЧЕРНО-БЕЛЫХ ИЗОБРАЖЕНИЙ

Алексеев А. В., Шпирко А.А.

Волгоградский государственный технический университет,

г. Волгоград, Россия

Alekseev.yeskela@gmail.com

alexs555@yandex.ru

В данной статье описан метод автоматизированного окрашивания черно-белого изображения с интеллектуальным анализом сцены изображения для повышения качества результата. Рассмотрены существующие аналоги, описана методика окрашивания.

Ключевые слова: окрашивание изображений; распознавание образов.

ВВЕДЕНИЕ

Проблема окрашивание черно-белых изображений на данный момент полностью не решена, это связано со многими сложностями: во-первых, при удалении цвета теряется информация, точно восстановить которую невозможно; во-вторых, необходимо понимать, что изображено на изображении, то есть необходимо решить задачу распознавания всех объектов, которая сейчас тоже не решена. При этом, окрашивание черно-белых фильмов и старых фотографий сейчас происходит вручную, например, фильм «17 мгновений весны» был раскрашен за три года, стоимость одной минуты – 3000\$. Кроме того, черно-белое изображение занимает втрое меньше дискового пространства, чем цветное, что позволяет при сохранении качества хранить втрое больше информации. На данный момент уже существует ряд решений автоматического окрашивания изображений [Vieira et al., 2007] [Rathore et al., 2010], однако у них есть значительные недостатки.

Актуальность работы обусловлена необходимостью сокращения затрат и времени при минимальных потерях качества. [Розалиев и др., 2010]

Целью работы является повышение эффективности окрашивания черно-белого изображения за счет автоматизации распознавания сигнатур изображения. Для реализации поставленной цели необходимо решить следующие задачи:

- 1) рассмотреть существующие аналоги, выявить достоинства и недостатки;
- 2) разработать методику окрашивания черно-белых изображений;
- 3) проанализировать методы сравнения изображений;
- 4) разработать способ описания объекта на изображении;
- 5) разработать метод сравнения объектов;
- 6) разработать систему окрашивания черно-белого изображения;
- 7) исследовать разработанную систему на предмет достижения поставленной цели.

1. Существующие аналоги

На данный момент существует ряд аналогов, рассмотрим их сильные и слабые стороны

Система авторов LIU Shi-Guang и др. предполагает действия пользователя в виде ввода полиномов. В ней отсутствует интеллектуальный анализ сцены, дополнительные входные данные формируются с помощью сходного цветного изображения.

Системы авторов Yogesh Rathore и L.F.M. Vieira не предполагают действий пользователя. В ней отсутствует интеллектуальный анализ сцены, дополнительные входные данные формируются с помощью сходного цветного изображения.

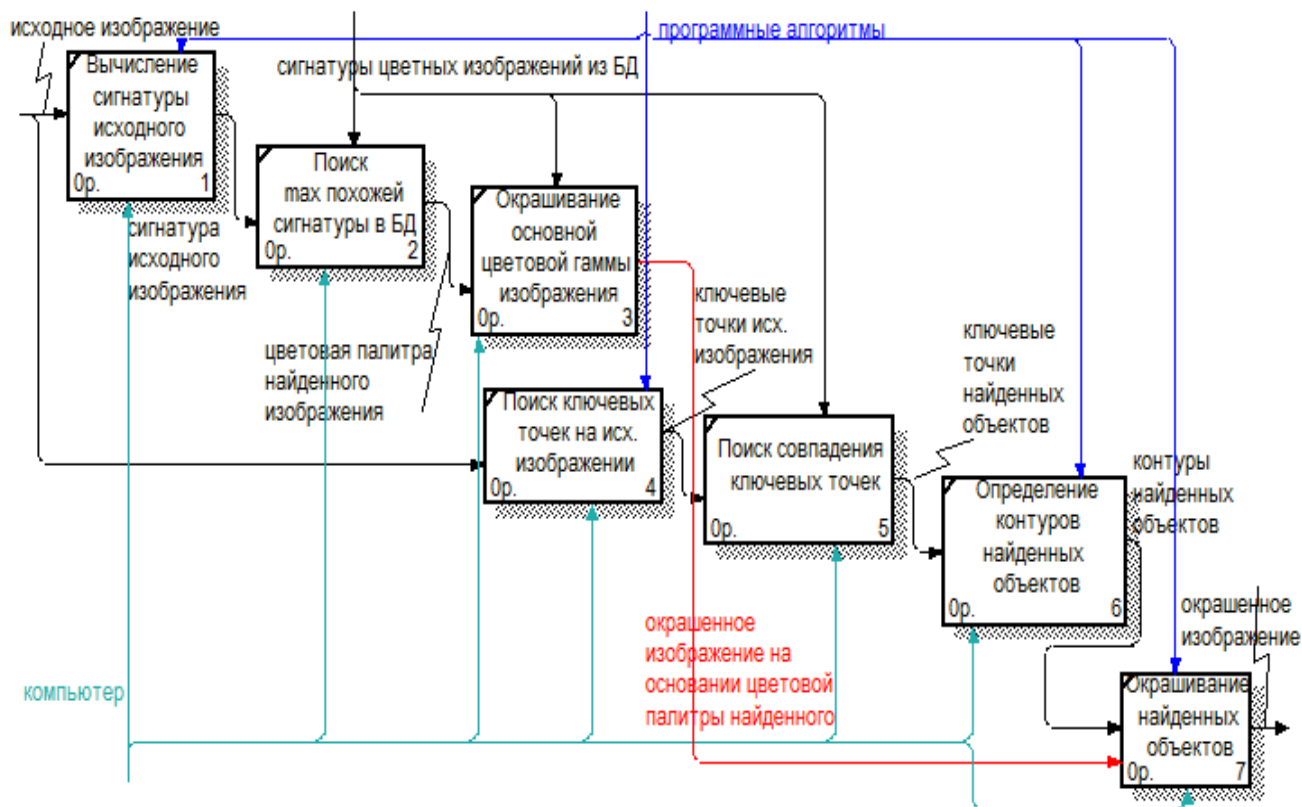


Рисунок 1 – Методика окрашивания

Системы Recolored и AKVIS coloriage работают после выделения областей и указания их цвета пользователем. В них отсутствуют интеллектуальный анализ сцены.

Рассмотренные аналоги работают с изображением как набором пикселей, не пытаются определить, что находится на самом изображении, из-за этого на насыщенных цветом изображениях могут быть получены сильно некорректные результаты, в разрабатываемой системе планируется исправить этот недостаток.

2. Методика окрашивания

Окрашивание черно-белого изображения подразумевает замену скалярного значения каждого пиксела на вектор (например, значения красного, зеленого и синего цветов).

Весь процесс окрашивания делится на два основных этапа в соответствии с методикой разработки программного обеспечения предложенной в [Заболеева-Зотова и др., 2010]: поиск основной цветовой гаммы изображения и поиск и окрашивание известных объектов (рисунок 1).

Данные этапы не связаны между собой и могут выполняться независимо, что позволяет уменьшить почти вдвое (за исключением процесса объединения результатов) время окрашивания.

2.1. Определение основной цветовой гаммы изображения

Основная цветовая гамма изображения определяется путем сравнения сигнатуры входного изображения с сигнатурами изображений из базы данных.

Сигнатура изображения – 128 численных значений, 64 получаются из гистограммы яркости исходного изображения и 64 из гистограммы яркости изображения после применения детектора границ Кэнни (Рисунок 2).

В работе [L.F.M. Vieira et al., 2007] были проведены эксперименты для определения оптимального количества значений сигнатуры по критериям производительности и схожести с исходным изображением, 128 значений было выбрано как оптимальное.

В базе данных кроме сигнатуры хранятся связанные с ней два канала цвета, полученные путем преобразования изображения в цветовую модель lab, каналы a и b.

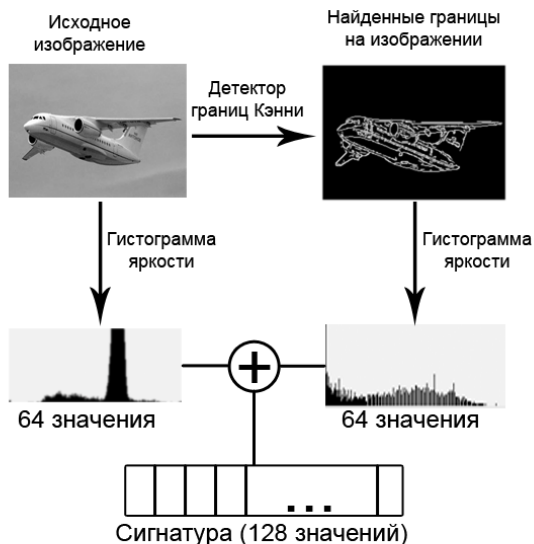


Рисунок 2 – Сигнатура изображения

2.2. Поиск и окрашивание известных объектов

Методы распознавания основываются на двух ключевых признаках – цвете и форме объекта, в условиях задачи мы вынуждены отказаться от методов распознавания, основанных на цвете объекта.

Рассмотрим методы распознавания, основанные на форме объекта.

1) Сравнение объектов на основе характеристики их контуров – инвариантных Ну моментов [Алексеев и др., 2012]:

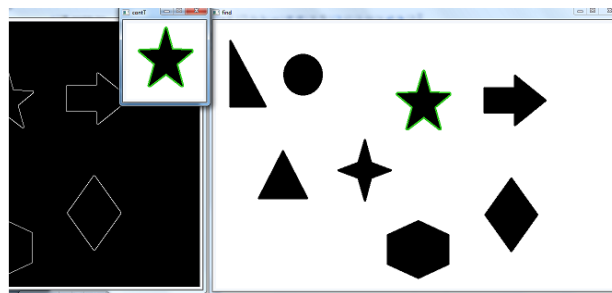
- + Сравнение не зависит от масштаба, вращения и отражения;
- + Простота реализации;
- Не учитываются внутренние особенности объекта;
- Если часть объекта закрыта другим объектом – метод дает неверный результат;
- Необходимо решить задачу поиска контуров.

3) Шаблонное сравнение:

Неизвестный объект сравнивается с известным, объекты представлены в виде бинарных матриц.

2) Speeded Up Robust Features (SURF):

Задача сравнения объектов сводится к задаче сравнения ключевых точек (точки перепадов яркости, углы и т.д.).



- Рисунок 3 – инвариантные Ну моменты контуров
- + Инвариантность к масштабу, повороту;
 - + Локализация даже при частичном перекрытии;
 - Не инвариантен к аффинным преобразованиям;
 - Не подходит для простых объектов;
 - Идентифицирует ключевые точки, а не объект.



Рисунок 4 – Метод SURF

- + Простота реализации;
- + Скорость работы;
- + Хорошо подходит для простых объектов;
- Сильно зависит от поворота, отражений;
- Неверный результат при перекрытии другим объектом.

Объекты на изображении будут локализоваться методом SURF, уточняться их границы будут с помощью поиска контуров и сравнения Ну моментов.

Опишем объект: Объект={Конт., Кл.Т., Цв}, где:

Конт. – инвариантный Ну-момент контура объекта;

Кл. Т. – ключевые точки объекта, полученные методом SURF;

Цв. – цветовая палитра объекта, соотношения цвета и яркости.

ЗАКЛЮЧЕНИЕ

В данной статье был проведен обзор существующих аналогов для выявления их слабых и сильных сторон, на основе анализа было сформулировано направление работ.

Рассмотренный подход позволяет за счет интеллектуального анализа сцены повысить качество окрашивания черно-белых изображений. Описанная методика позволяет этого добиться без увеличения времени обработки, благодаря независимости этапов.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 12-07-00266, 12-07-00270).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Vieira et al., 2007] L.F.M. Vieira, Fully automatic coloring of grayscale images/ L.F.M. Vieira, E.R.D. Nascimento, F.A.F. Jr., R.L. Carceroni, R.D. Vilela, A.D.A. Araújo, //Image Vision Comput., 2007, pp.50-60.

[Rathore et al., 2010] Yogesh Rathore, Colorization of Gray Scale Images using Fully Automated Approach/ Yogesh Rathore, Avinash Dhole, Ramnivas Giri, Umesh Agrawal // IJECT Vol. 1, Issue 1., 2010.

[Алексеев и др., 2012] Алексеев, А.В. Автоматизация определения шрифтов по изображению / Алексеев А.В., Розалиев В.Л. // Труды конгресса по интеллектуальным системам и информационным технологиям "IS&IT'12" (Дивноморское, Краснодарский край, 2-9 сент. 2012 г.). В 4 т. Т. 1 : тр. конф. "Интеллектуальные системы '12" и "Интеллектуальные САПР – 2012" / ЮФУ [и др.]. - М., 2012. - С. 292-293.

[Розалиев и др., 2010] В.Л. Розалиев, А.С. Бобков, О.С. Федоров Применение нейронных сетей и грануляции при построении автоматизированной системы определения эмоциональных реакций человека / Изв. ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах»: межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2010. - Вып. 9, № 11. - С. 63-68.

[Заболеева-Зотова и др., 2010] Заболеева-Зотова А.В., Орлова Ю.А. Автоматизация начальных этапов проектирования программного обеспечения // Изв. ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах". Вып. 8 : межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2010. - № 6. - С. 121-124.

INTELLIGENT SCENE ANALYSIS FOR COLORING BLACK AND WHITE IMAGES

Alekseev A.V., Shpirko A.A.

*Volgograd State Technical University,
Volgograd, Russia*

**Alekseev.yeskela@gmail.com
alexs555@yandex.ru**

This paper describes an automated method for coloring black and white images with intelligent scene analysis to improve the image quality of the result. The existing commands described staining technique.

INTRODUCTION

The problem of coloring black and white images at the moment is not completely solved, it is connected with many difficulties: first, removing the color

information is lost, which can not be accurately restored, and secondly, we must understand what is depicted in the image, that is to be solved problem of the recognition of all the objects, which is now also solved.

MAIN PART

Analogy works with the image as a set of pixels, not trying to determine what is on the image, because of this on the color in the image may be obtained highly incorrect results in the system being developed will correct this deficiency.

System authors LIU Shi-Guang et al suggests to user input in the form of polynomials. There is no intelligent Scene Analysis, additional input data are generated using a similar color image.

System authors Yogesh Rathore and LFM Vieira not involve user interaction. There is no intelligent Scene Analysis, additional input data are generated using a similar color image.

Recolored system and AKVIS coloriage work after the selection and designation of their color by the user.

Coloring black and white refers to replacing a scalar value of each pixel vector. The whole process of painting is divided into two main stages: the search for the basic color range and image search and staining of known objects. The main colors of the image is determined by comparing the signature of the input image to the signatures of images from the database.

The main colors of the image is determined by comparing the signature of the input image to the signatures of images from the database. The signature image - 128 numerical values, 64 are obtained from the original image histogram and 64 of the histogram of the image after applying the detection limits Kenni.

Recognition methods are based on two key attributes - color and shape of the object in the problem we have to abandon the methods of recognition based on the color of the object.

CONCLUSION

Approach makes it possible through intelligent scene analysis to improve the quality of staining black and white images. The described method allows to achieve this without increasing the processing time, due to the independence of stages.

This work was partially supported by the Russian Foundation for Basic Research (projects 12-07-00266, 12-07-00270).

СЕКЦИЯ 6.

**КОМПЛЕКСНЫЕ МЕТОДЫ, ОНТОЛОГИЯ И МЕНЕДЖМЕНТ
КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ
СИСТЕМ**

SECTION 6.

**COMPREHENSIVE METHODS, ONTOLOGY AND MANAGEMENT OF
COMPONENT DESIGN OF INTELLIGENT SYSTEMS**



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПРОГРАММНОЕ УПРАВЛЕНИЕ ПОТОКАМИ РАБОТ В КОНЦЕПТУАЛЬНОМ ПРОЕКТИРОВАНИИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Соснин П.И. *, Лапшов Ю.А. *, Маклаев В.А. **

* *Ульяновский государственный технический университет, г. Ульяновск, Россия*

sosnin@ulstu.ru

y.lapshov@ulstu.ru

** *Федеральный Научно-Производственный Центр ОАО «НПО «МАРС», г. Ульяновск, Россия*

mars@mv.ru

Представляются средства и язык псевдо-кодowego программирования потоков работ, предназначенные для алгоритмического представления бизнес-процессов, исполняемых в проектировании автоматизированных систем (АС). К специфике языка относится его ориентация на интеллектуальный процессор (роль проектировщика), разделяющий общую работу с компьютерным процессором и с другими исполнителями бизнес-процессов.

Ключевые слова: автоматизированное проектирование, потоки работ, программное управление, псевдо-кодowego программирование.

ВВЕДЕНИЕ

Успешность создания любой АС в существенной мере зависит от того, «какие средства используют проектировщики для управляемого овладения её сложностью?». В оценках степени сложности систем, чаще всего, применяют различные интерпретации колмогоровской меры сложности, «как минимальной длины программы P построения системы S из её заданного исходного описания D » [Li et al, 2008]. Различия в интерпретациях обусловлены, в первую очередь, особенностями системы S , а также тем, «какое содержание вкладывается в конструкции P и D , и как это содержание специфицируется?»

В разработках АС конструкции типа P приходится создавать по ходу её проектирования, используя определённый «метод программирования M », включающий теоретическую, методическую и практическую составляющие. К одной из важнейших особенностей АС относится то, что в их разработках сложностью P необходимо овладеть не в меньшей мере, чем сложностью АС, причём исходя из того же исходного описания D . Такое отношение между АС и P позволяет вводить структуризацию в жизненный цикл создания АС, разделяя P на связную совокупность частей $\{P^i\}$,

каждая из которых «отвечает» за формирование определённого состояния АС в её жизненном цикле.

Разделение P на части и, соответственно, выделение состояний $\{AC^i\}$, каждое из которых AC^i выполняет функции описания D для построения очередного состояния AC^{i+1} является традиционным приёмом овладения сложностью в разработках АС. Такое овладение находит своё материальное выражение в технологиях разработки, предоставляющих создателям АС определённые возможности построения P , а значит и частей $\{P^i\}$ этого конструкта. Профессиональная зрелость конкретной технологии T^* , в первую очередь, определяется тем, какие средства она предоставляет коллективу разработчиков АС для построения конструктов $\{P^i\}$, которые они же должны и оперативно исполнять.

В большинстве технологий разработок АС для достижения отмеченных целей разработчикам предоставляются средства традиционного планирования, которые не ориентированы на программные представления конструктов типа P .

В ряде технологий, например в Rational Unified Process [Кролл, 2004], традиционные средства планирования комбинируют со средствами моделирования бизнес-процессов в форме потоков работ, в том числе и представляющих их с помощью

исполняемых специализированных языков (BPEL, YAWL, XLANG, ...), но с ограниченными возможностями программного управления.

По глубокому убеждению авторов, для конструктивного учёта сложности и её оперативной минимизации (упрощения) разработчики АС должны программировать свою деятельность, применяя для этого средства унифицированного программного управления любыми потоками работ, которые порождаются в их деятельности.

Сложность является характеристикой взаимодействия проектировщиков с состояниями проекта в процессах решения задач. Проблемы со сложностью в тех взаимодействиях, которые являются источниками опасных и дорогостоящих ошибок. В этом плане наиболее опасны задачи концептуального проектирования АС.

Именно с программным управлением работой проектировщиков на концептуальном этапе связаны интересы статьи, в которой авторы представляют подход к управлению работами, в основу которого положено псевдо-кодовое программирование, ориентированное на его оперативное применение в процессах согласованного решения задач. Подход доведён до его практического применения в инструментально-моделирующей среде WIQA (Working In Questions and Answers) [Соснин, 2012].

Потоки работ концептуального проектирования

Потоки работ используются как базовые модельные представления динамики бизнес-процессов практически во всех современных технологиях разработки автоматизированных систем.

Каждый поток работ должен адекватно представлять совокупность задач проектирования, процесс согласованного решения которых приводит к осуществлению соответствующего БП. Более того, в общем случае, в процесс решения может быть вовлечена группа проектировщиков $\{D_v\}$, каждый из которых несёт ответственность за решение назначенных ему проектных задач $\{Z^{S_{iv}}\}$ в конкретных условиях. Обобщённое представление потока работ, раскрывающее его структурное содержание, приведено на рисунке 1.

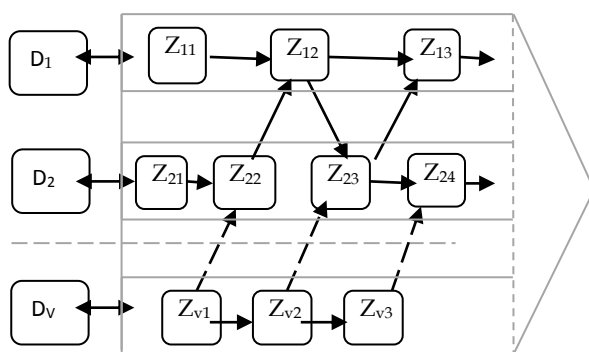


Рисунок 1 - Пример структуры потока работ

Основное предназначение потоков работ, как моделей, – автоматизация бизнес-процессов, в которой выделяются автоматизация потоков работ, автоматизация согласованного исполнения задач в потоках и автоматизация решения задач по образцам, причём во всех видах автоматизации компьютеризуется работа лиц, вовлеченных в исполнение бизнес-процессов [Held et al., 2009].

В теории и практике потоков работ, которые не привязаны к конкретным видам деятельности, среди выделенных направлений автоматизации основное внимание уделяется первым двум из-за намеренного абстрагирования от того, в какие действия человека и с чем выливается решение назначенных ему задач.

В любых его приложениях проектирование АС является деятельностью команды проектировщиков $K(\{P_v\})$, которая, автоматизируя определённую систему бизнес-процессов $S(\{BP_m\})$, должна согласованно решать определённую совокупность нормативных задач $\{Z^T_j\}$ используемой технологии T^* . По сути дела, в такой деятельности систему решений задач $S(\{Z^{S_{im}}, t\})$, вложенных в $S(\{BP_m\})$, пытаются представить через их «разложение» в базисе задач технологии T^* .

В любой технологии T^* её задачи объединяют в потоки работ $\{W_n(\{Z^{T_{nj}}\})\}$, а потоки работ объединяют в подсистемы $C^q(\{W_j(\{Z^{T_{nj}}\})\})$ в соответствии со специализацией этапов проектирования. Обычно задачи представлены инструкциями, раскрывающими схемы действий, каждая из которых должна быть адаптирована к соответствующим бизнес-процессам АС.

Необходимость адаптации задач $\{Z^{S_{mi}}\}$, обусловленная инвариантностью задач технологии к потенциальным предметным областям АС, выводит процесс проектирования на специфический класс творческих задач адаптации $\{Z^A_k\}$, включающих задачи когнитивного анализа, оценивания, принятия решений и другие задачи, требующие от проектировщика ситуативной (незапланированной заранее) интеллектуальной активности [Sosnin, 2011]. Более того, такая адаптация должна осуществляться проектировщиками согласованно. Именно необходимость согласованной интеллектуальной адаптации задач в реализации потоков работ технологии определяет важнейшую их специфику.

Реальная практика такой адаптации является источником многочисленных и разнородных ошибок и «дефектов», оказывающих негативное воздействие на процесс проектирования и его результаты. Основным источником негативов является человеческий фактор, а вернее, исполнение проектировщиками их функций в условиях недостаточной степени автоматизации активности в решении задач адаптации. Такое положение дел указывает на несовершенство существующих технологий $\{T^n\}$, подтверждаемое мировой статистикой – степень успешности разработок АС около 35 %.

За последние 20 лет использовались различные средства совершенствования технологий класса $\{T^n\}$, например, модели совершенствования профессиональной деятельности [Roglinger et al., 2012], базы опыта [Basili et al., 2001] и фабрики опыта [Henninger, 2003], унифицированный язык моделирования [Кролл, 2004], нормативные средства описания и реализации архитектурных решений [IEEE, 2000]. Шаг за шагом появлялись новые полезные средства и совершенствовались уже освоенные, однако степень успешности разработок АС почти не повышалась. А значит, проблема успешности проектирования АС всё ещё существует и поиски путей её решения за счёт технологического совершенствования деятельности проектировщиков актуальны.

Представление потоков работ в среде WIQA

Инструментальная среда WIQA изначально разрабатывалась для моделирования совокупностей технологических (нормативных) и предметных проектных задач, которые приходится решать разработчикам АС. Как источник нормативных задач были использованы типовые задачи потоков работ концептуального проектирования в технологии Rational Unified Process (RUP) [Кролл, 2004]. По его предназначению и содержанию, комплекс WIQA разрабатывался как специализированная АСТ технологического типа.

Одной из важнейших особенностей концептуального проектирования в среде WIQA является потенциальное применение всех средств этого инструментария к любой задаче Z_i дерева $T(\{Z_i\})$ проектных задач $\{Z_i\}$, если в этом будет необходимость. В таком применении задача Z_i представляется её вопросно-ответной моделью (QA-моделью, $QA(Z_i)$), структурирующей процесс решения задачи в формах вопросно-ответных рассуждений. Интерфейсная форма доступа к дереву задач (Tree of Tasks, TT) и QA-моделям представлена обобщённо на рисунке 2.

Информационный потенциал дерева задач

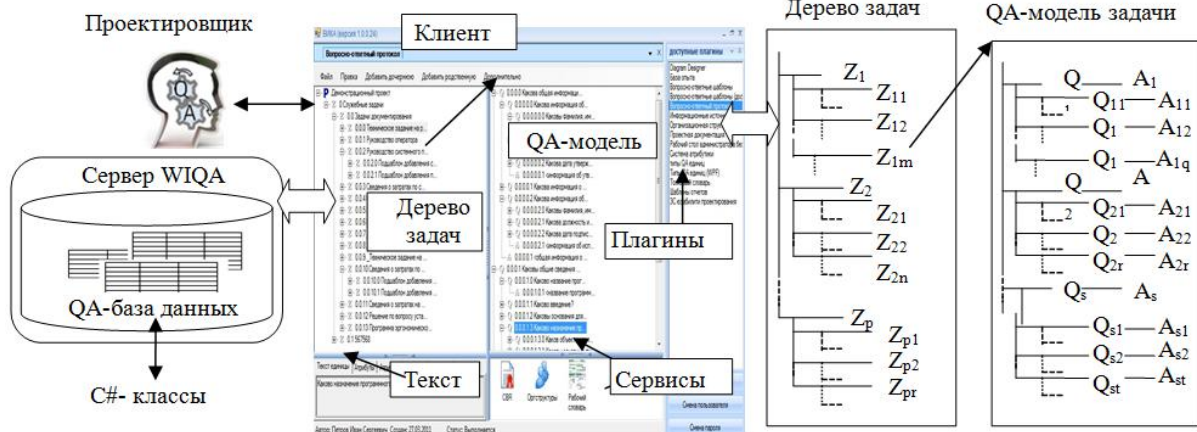


Рисунок 2 - Инструментально-моделирующая среда WIQA

$TT(\{Z_i\})$ и связанного с ним множества моделей $\{QA(Z_i)\}$ достаточен для представления текущего состояния совокупности потоков работ $\{W_m\}$ любого проекта $\Pi(TT(\{Z_i\}), \{QA(Z_i)\}, K(\{D_v\}), t)$, разрабатываемого коллективом $K(\{D_v\})$ проектировщиков $\{D_v\}$. Для представления и использования конструкта $K(\{D_v\})$ в комплекс WIQA встроено расширение (плагин), названный «Оргструктура».

На основе этого потенциала авторами разработан комплекс средств для представления проектов, разрабатываемых в инструментальной среде WIQA, в виде динамической схемы $S(\{W_m\}, t)$ потоков работ, предназначенной для управления процессами проектирования. В схеме $S(\{W_m\}, t)$ с каждым потоком W_m связана определённая задача Z_m^W дерева задач, которой приписан тип «поток работ» и символическое обозначение типа W .

Таким образом, в дерево задач может быть введена разметка, регистрирующая «какие задачи включены в каждый выделенный поток работ в текущем состоянии проекта?». Введение разметки приводит к новой версии систематизации $\Pi(G(\{Z_i\}, \{Z_m^W\}), \{QA(Z_i)\}, \{QA(Z_m^W)\}, K(\{D_v\}), t)$ представления задач проектирования. Эта версия систематизации используется как основной источник информации для решения задач управления потоками работ.

Для управления потоками разработан и включён в состав комплекса WIQA ряд плагинов, обеспечивающих:

- псевдо-кодовое программирование задач, ориентированное на исполнение программ проектировщиком, выполняющим роль интеллектуального процессора (I-процессора) и использующим компьютерный процессор (K-процессор) как подчинённый процессор;
- управление прерываниями человеко-компьютерной деятельности, в частности управление прерываниями при исполнении псевдо-кодовых программ для технологических задач, кодирующих нормативные инструкции;

- оперативное использование паттернов потоков работ, псевдо-кодовые аналоги которых хранятся в специальной библиотеке.

Кроме того, для представления событий всех типов используется атрибутика объектов, входящих в дерево задач и QA-модели, в частности их уникальные индексные имена, имена проектировщиков, время последней модификации и состояния («в работе», «прервана» и «завершена»). Для учёта временных характеристик работ (решений задач всех типов) и их соответствия договорённостям разработана специализированная система «Контроля поручений». Общая картина управления потоками обобщённо представлена на рисунке 3.

Инструментарий обслуживает управляемое оперативное формирование, регистрацию, визуализацию и использование:

1. Текущего состояния дерева задач $T(\{Z_i\}, t)$, включающего:

- совокупность задач $Z^S = \{Z_{mi}\}$ предметной области AC, которые в проектируемой AC будут решать её пользователи;
- совокупность нормативных задач $Z^T = \{Z_{ni}\}$ технологии, в рамках которой коллектив проектировщиков $K(\{D_v\})$ создаёт AC;
- совокупность задач адаптации $Z^A = \{Z^A_{qi}\}$, решаемых проектировщиками для настройки задач типа Z^T , инвариантных к проблемной области AC, в их использовании при решении задач типа Z^S ;
- совокупность задач управления $Z^W = \{W^m(\{Z_{mi}\}) \cup \{W^n(\{Z_{ni}\})\}$ и $Z^C = \{Z^W_{ri}\}$, решение которых обслуживает работу с задачами в потоках работ, а также согласованное управление в группах потоков работ;
- объединения задач, зарегистрированных в дереве задач, в потоки работ или группу потоков, каждому из которых соответствует задача типа Z^W

или Z^C , демонстрирующая динамические отношения между решениями задач во времени (средства формирования потоков работ) [Sosnin, 2012b].

2. Распределение задач, зарегистрированных в дереве, между их исполнителями $\{D_v\}$ в коллективе $K(\{D_v\})$ с использованием процедуры назначения задач (средства работ с оргструктурой коллектива и деревом задач) [Соснин, 2012].

3. Предварительное планирование сроков исполнения работ в процессах решения задач всех типов (с использованием средств контроля поручений).

4. Оперативное псевдо-кодовое программирование (с использованием средств вопросно-ответного программирования, QA-программирования) задач [Sosnin, 2012a], выбранных для решения на текущем шаге работ:

- построение псевдо-кода для назначенной проектировщику D_v задачи Z_i с возможностью регистрации псевдо-кода в виде QA-модели задачи Z_i по месту её «расположения» в дереве задач или сохранение псевдо-кода в библиотеке QA-программ (средства редактирования и сопровождения библиотек);
- использование в построении QA-программ паттернов потоков работ (средства библиотеки паттернов);
- оперативное исполнение QA-программы проектировщиком в режиме интерпретации (средства интерпретации, интерпретатор) [Sosnin, 2012c];
- компиляция QA-программы и её автоматическое исполнение компьютерным процессором (средства формирования очереди задач для их исполнения в потоках работ).

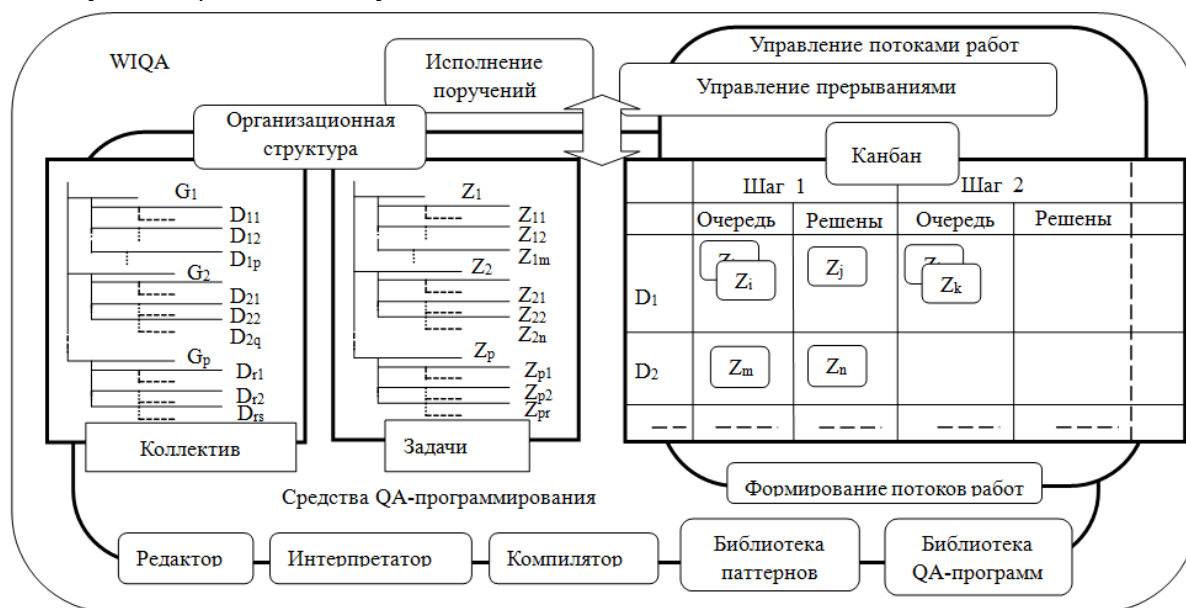


Рисунок 3 - Инструментальное обеспечение управления потоками

5. Параллельное решение задач в коллективе проектировщиков на очередном шаге работ (средства работы с очередями задач с использованием визуализатора «Канбан»).

6. Псевдопараллельная работа каждого проектировщика с назначенными ему задачами (средства прерывания задач).

Для ряда перечисленных функциональных возможностей использованы ссылки на публикации, в которых раскрыты детали их реализации в среде WIQA. Из тех, для которых ссылок нет, ниже раскрыто детали языка L^{WIQA} и P-модели для паттернов потоков работ и взаимодействия проектировщиков с очередями задач.

В документации комплекса WIQA используется его интерпретация как процессора, обеспечивающего регистрацию и использование моделей вопросно-ответных рассуждений (QA-рассуждений) проектировщиками в процессах решения назначенных им задач. В такой интерпретации процессора WIQA модели типов «задачи», «вопросы» и «ответы» определяются как интерактивные объекты, к которым применимы соответствующие команды.

Модели Z-, Q- и A-типов размещаются в QA-базе данных процессора, представляющей собой память (QA-память) с богатой атрибутикой «ячеек». В «ячейках» памяти можно хранить любые данные, которые будут намеренно или нет наследовать атрибутику «ячеек». Отметим, что совокупность

атрибутов «ячейки» включает кроме атрибута «символьное описание», например, атрибуты «уникальное имя», «проектировщик, использующий ячейку», «тип ячейки», «момент времени записи». Более того, в комплекс WIQA встроены механизмы, позволяющие любой ячейке оперативно присписывать любые полезные атрибуты и любое их количество.

QA-память оказалась очень полезной для погружения в её структуры любых задач и их QA-моделей, в том числе и в форме QA-программ с их данными (QA-данными) и операторами (QA-операторами). Отметим, что квалификатор «QA» указывает на специфичность конструкторов, подчёркивая их отличие от конструкторов, родственных по предназначению. Описанная интерпретация в обобщённой форме приведена на рисунке 4.

Атрибутика, наследуемая данными и операторами, привела к созданию псевдо-кодового языка, ориентированного на исполнение QA-программ проектировщиками, которые сами же могут их строить, если в этом появилась необходимость.

В такой работе проектировщики исполняют роль, названную I-процессором [Соснин, 2012], обеспеченную полезными инструментами по образцу других ролей [Borges et al., 2012], исполняемых ими в используемой технологии.



Рисунок 4 - Представление задач в QA-памяти процессора WIQA

Язык псевдо-кодowego программирования WIQA

Псевдо-кодový язык QA-программирования принципиальным образом зависит от среды его употребления, из-за чего он получил имя «WIQA» и обозначение L^{WIQA} [Соснин, 2012]. Специфику языка L^{WIQA} определяют следующие его особенности:

1. Язык L^{WIQA} специально приближен к естественному языку в его алгоритмическом употреблении, обслуживающему взаимодействие человека с доступным опытом в деятельности. По этой причине язык ориентирован на решение задач, за которыми стоят прецеденты и их связанные совокупности.

2. Текст исходного псевдо-кода программы на языке L^{WIQA} построен из представлений его структурных единиц (предложение за предложением, оператор за оператором) с помощью QA-данных. Так что исходный код представляет собой версию QA-модели задачи, для которой псевдокод был разработан. По этой причине, весь потенциал инструментария WIQA, применимый для QA-моделирования [Соснин, 2012], применим и к исходным псевдо-кодам программируемых задач.

3. Язык L^{WIQA} – объектно-ориентирован, поддерживает программирование задач с базами данных и программирование задач управления потоками работ в коллективе.

4. Если работа, запрограммированная на языке L^{WIQA} , может быть «передана» компьютеру, то соответствующая QA-программа компилируется в код, исполняемый компьютерным процессором (К-процессором).

Язык разрабатывался как открытый для включения в его структуру дополнительных типов QA-данных и QA-операторов. Существующая структура языка приведена на рисунке 5.

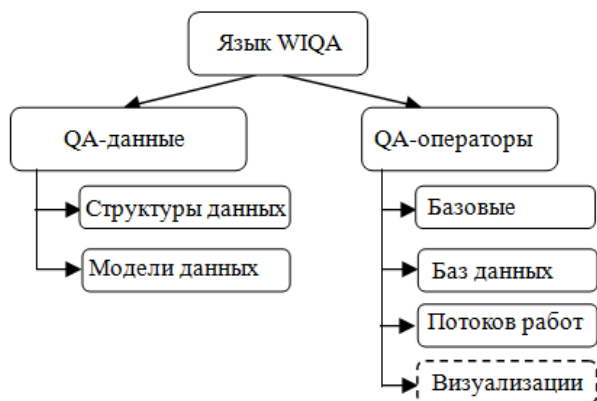


Рисунок 5 - Состав языка WIQA

На рисунке 5 отмечены отдельными группами:

- «Структуры данных», за которыми стоит эмуляция любых традиционных типов данных

(скаляры, массивы, записи, множества, стеки, очереди, ...) с помощью средств QA-данных;

- «Модели данных», используемые для спецификации табличных структур и их совокупностей, а также иерархических структур;

- «Базовые операторы» традиционных псевдо-кодových языков программирования (Appoint, Input, Output, If, GOTO другие);

- «Операторы баз данных», эмулирующие основные SQL-операторы;

- «Операторы потоков работ», эмулирующие основные операторы языков имитационного моделирования:

- «Операторы визуализации», которые разрабатываются в настоящее время для просмотра по запросам проектировщиков ячеек QA-памяти по сценариям, запрограммированным предварительно или оперативно.

Отметим, что средства объявления данных, базовый набор операторов и операторы баз данных детально представлены в публикации [Соснин, 2012]. Там же детально раскрыты спецификации роли «I-процессор» в его сопоставлении с традиционным набором ролей (около 40 ролей), используемых в технологиях разработки АС. По этой причине, ниже раскрываются детали синтаксиса и употреблений языка L^{WIQA} только в задачах управления потоками работ.

Псевдо-кодowego программирование паттернов потоков работ

В управлении бизнес-процессами накоплен богатейший опыт, очень важная часть которого аккумулирована в паттернах потоков работ [Aalst et al., 2005]. Роль паттернов настолько значительна, что существует международная ассоциация учёных и практиков, активность которых регистрируется на специальном Интернет-сайте (<http://www.workflowpatterns.com>). На текущий момент на этом сайте зарегистрировано 43 паттерна, содержание которых было использовано авторами для выявления совокупности псевдо-кодových операторов, включение которых в язык L^{WIQA} позволит использовать этот язык и в QA-программировании задач типов Z^W и Z^C .

При отборе операторов, расширяющих язык L^{WIQA} для программирования задач управления, проводились параллели между расширением и языками имитационного моделирования. Более того, для родственных операторов были сохранены их имена, устойчиво используемые в имитационном моделировании.

Для отобранной совокупности операторов произведена доработка интерпретатора и компилятора псевдо-кодов, позволяющая исполнять программы, созданные с использованием базовых операторов и операторов расширения, что позволило создать библиотек паттернов. Для примера рассмотрим паттерн «параллельное расщепление» (parallel split) [Aalst et al., 2005], одна

из версий которого приведена на рисунке 6, где названы имена задач, используемые в псевдо-кодовой программе (группа – субъект, которому назначена задача).

В этой версии паттерна, исполняемого тремя (группами) проектировщиков, после того как решена Задача_1, поток расщепляется на три ветки (Задача_2, Задача_3 и Задача_4). Унификацию версий (количество расщеплений, исполнители задач) обеспечивает оператор SET, позволяющий установить значения элементам списка переменных

```
SET <Переменная_1>, <Значение_1>;
    <Переменная_2>, <Значение_2>;...;
    <Переменная_N>, <Значение_N>.
```

Функции переменных в данном операторе могут выполнять, как псевдо-кодовые переменные простых типов, так и элементы массивов и поля объектов. Оператор SET можно вызывать в любом месте программы для любого набора переменных.

Рассматриваемый паттерн представлен в библиотеке следующим псевдокодом:

```
//Инициализация
SET &in&, 1; &outs[0]&, 2; &outs[1]&, 3;
&outs[2]&, 4; &outgroup[0]&, 1; &outgroup[1]&, 2;
&outgroup[2]&, 3; &cnt&, 0
LABEL &L1&
SEIZE &outs[&cnt&]&, &outgroup[&cnt&]&
INC &cnt&
TEST L, CNT, outs.length &L1&
SET &cnt&, 0;
//Выполнение задач
LABEL &L2&
QUEUE &outs[&cnt&]&, &in&.state == done
INC &cnt&
TEST L, &cnt&, outs.length &L2&
FINISH
```

В псевдокоде паттерна присутствуют две части, разделенные комментариями. Первая часть обеспечивает распределение задач между исполнителями, в котором основная нагрузка падает на оператор SEIZE (ключевое слово заимствовано из имитационного моделирования). Этот оператор «называет» типовую работу, представленную библиотечной QA-программой со следующим псевдокодом:

// QA-программа SEIZE

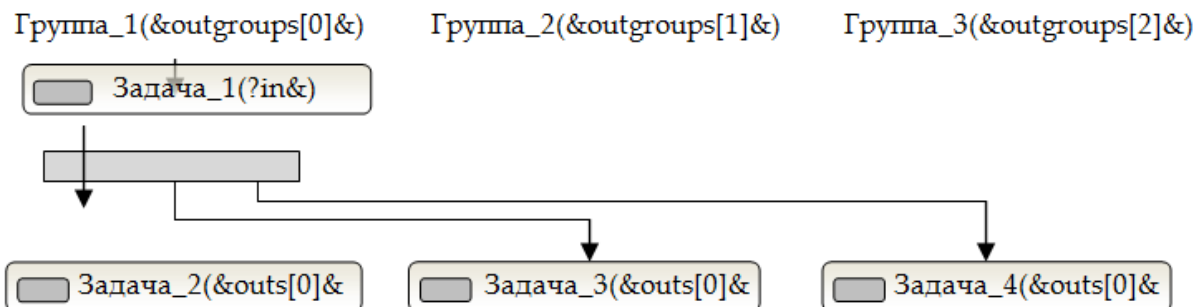


Рисунок 6 - Структура паттерна «Parallel split» (Параллельное расщепление)

```
//Установка ассоциации между группой и
задачей в оргструктуре
CALL &AssignTask&
//Добавление ассоциации в систему контроля
поручений
&ProjectId& := QA_GetProjectId("Workflow Data
bases")
&BaseId& := QA_GetQAId(&ProjectId&,
"Commissions")
//Получение идентификатора БД
&Database& := OpenDB(&BaseId&)
//Вдруг задача уже назначена? Проверка
&Query& := "SELECT GroupId FROM
Commissions WHERE TaskId="
&Query& := concat (&Query&, <Задача>)
&Query& := concat (&Query&, ";") //Построение
запроса
&Res& := ExecuteSQL (&Database&, &Query&)
if (&Res& <> "NULL") then return //Задача уже
назначена
//Формирование запроса, добавляющего задачу в
очередь
&Query& := "INSERT INTO Commission
(GroupId, Task) VALUES ("
&Query& := concat(&Query&, <Исполнитель>)
&Query& := concat(&Query&, ", ")
&Query& := concat(&Query&, <Задача>)
&Query& := concat(&Query&, ");")
//Выполнение запроса – ассоциирование задачи
ExecuteSQL (&Database&, &Query&)
```

Принципиальной составляющей псевдокода SEIZE является назначение задачи исполнителю с использованием плагина «Оргструктура» и дерева задач текущего проекта:

1. procedure &AssignTask& . вызывается для назначений по каждой задаче (_Z1, ...)
2. Запустить плагин «Оргструктура»
3. Выбрать группу <Исполнитель>, которой требуется назначить задачу
4. В правой части окна нажать кнопку Выбор проекта
5. Выбрать проект, содержащий требуемую задачу
6. Найти задачу <Задача> в дереве проекта
7. Нажать кнопку «Назначить задачу»
8. endproc &AssignTask&

Каждое произведённое назначение регистрируется в основной таблице плагина «Контроль поручений», атрибутика которой включает «Уникальный идентификатор поручения», «Идентификатор исполнителя поручения», «Тип поручения», «Идентификатор назначенной задачи», «Резолюция», «Важность», «Срок исполнения», «Контрольные точки» и «Время выполнения задачи».

Библиотека паттернов разработана для облегчения оперативного программирования потоков работ, необходимость в которых выявлена в процессе проектирования. Одним из важнейших источников таких конструктов является пошаговая детализация, по ходу которой, например, задача Z_j определённого уровня (пусть уровня J), разбивается на связную совокупность задач $S(\{Z_{j+1}\})$ уровня $J+1$. Разумеется, до QA-программирования такого потока динамические отношения между подчинёнными задачами должны быть представлены схемой потока W^k .

Управление очередями задач

Как уже отмечалось выше, QA-программирование управляющей программы начинается с подготовки потока W^k к исполнению, включающей назначения задач исполнителям и установлению плановых характеристик для их решения.

Предположим, что эта работа и все необходимые остальные работы учтены, и QA-программа для потока W^k построена. Разумеется, построенную программу необходимо проверить. Одной из версий проверки является представление QA-программы в базе паттернов проектирования, псевдокоды которых заимствуются из библиотеки паттернов и встраиваются в проверяемую программу.

Именно такая нагрузка и возлагается в совокупности средств программного управления на библиотеку паттернов потоков работ. Сам факт разложения QA-программы потока в базе паттернов является важным аргументом в пользу корректности построенной QA-программы.

Предположим, что QA-программа для потока (пусть W^k) построена и проверена, причём не только через её разложение в базе паттернов. После этого её следует исполнить. В авторской версии программного управления потоками исполнение завершающей фазы QA-программы потока осуществляется компилятором, который по информации о задачах, вложенной в псевдокод потока, дополняет соответствующие очереди задач новыми составляющими. Другими словами, программное управление потоками работ нацелено на формирование очередей задач для каждого члена коллектива разработчиков.

Более того, авторы предлагают различать и строить очереди задач в виде специализированных программ двух типов (M^1 -программ и M^2 -

программ), состоящих из операторов, каждый из которых может исполняться только в случае истинности зафиксированного в операторе условия.

То есть каждый элемент в любой очереди независимо от её типа оформляется и интерпретируется как условный оператор

If <условие доступа к элементу очереди> Then <Доступ к задаче по её идентификатору>, исполняемый проектировщиком в роли I-процессора (рисунок 7).

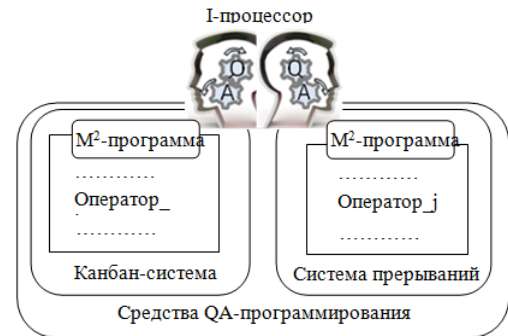


Рисунок 7 - Взаимодействие I-процессора с очередями задач

Различия между очередями M^1 - и M^2 -типов состоит в том, что:

- в M^1 -программах условия доступа регистрируют логику отношений между задачами потока во времени, управляя тем самым параллельно-согласованным решением задач группой проектировщиков;
- в то время как условия операторов M^2 -программ управляют псевдопараллельным решением задач конкретным проектировщиком, каждая из которых может быть прервана по ходу решения, если в этом возникла необходимость.

Использование очередей двух типов позволяет отделить:

- визуализацию общей картины исполнения задач группой проектировщиков;
- от интерактивных взаимодействий проектировщика с задачами, которые он может выполнять псевдопараллельно.

Для формирования очередей в расширение языка L^{WQA} введён оператор QUEUE. Основное назначение оператора зарегистрировать условие, истинность которого открывает возможность для начала работы с задачей

QUEUE <Задача>, <Условие>.

За этим оператором также как и за оператором SEIZE стоит библиотечная QA-программа, псевдокод которой выглядит следующим образом:

```
&ProjectId& := QA_GetProjectId("WorkflowDatabases")
&BaseId& := QA_GetQAId(&ProjectId&,
"Commissions")
&Database& := OpenDB(&BaseId&) //Получение
идентификатора БД
&Query& := "SELECT GroupId FROM Commissions
WHERE TaskId="
```

```

&Query& := concat (&Query&, <Задача>)
&Query& := concat (&Query&, ";") //Построение
запроса
&Res& := ExecuteSQL(&Database&, &Query&)
//Выполнение запроса
//Заполнение переменных, используемых функциями
управления очередями
&QueueId& := UnpackInt(&Res&)
&TaskId& := <Задача>
//Автоопределение приоритета для размещения
задачи в конце очереди
&Priority& := -1
&TaskCondition& := "<Условие>";
//Вызов функции добавления задачи &TaskId& в
очередь &QueueId&
CALL &AddTask&

```

Процедура `&AddTask&` подсистемы управления очередями задач выполняет взаимодействие с таблицей базы данных очередей задач, выполняет проверки, и после них заносит в таблицу очередей новую задачу.

Визуализация общей картины осуществляется с помощью средств интерактивной «доски» Канбан [Wang et al., 2010], фиксирующей для выбранной совокупности шагов проекта состояния работ с задачами «в очереди работ» или «завешена». Идентификаторы задач регистрируются в очередях системы Канбан в результате выполнения соответствующих операторов управления QUEUE.

Псевдопараллельное решение задач обслуживается средствами системы прерываний I-процессора с учетом информации по контролю за поручениями, которая используется для оперативных вычислений динамических приоритетов. Коррекция приоритетов осуществляется автоматически при очередном обращении проектировщика к очереди. Приоритеты помогают в выборе, но не диктуют его проектировщику, который свободен в действиях такого типа.

ЗАКЛЮЧЕНИЕ

Разработанные и представленные в статье средства управления потоками работ предназначены для программного формирования очередей задач, взаимодействие с которыми обеспечивает согласованную работу коллектива проектировщиков в разработке АС. Для программирования задач управления разработано расширение псевдо-кодowego языка L^{WQA} до его версии, позволяющей создавать программные модели, как для процессов решения задач, так и для управления их согласованным решением в потоках работ. Операторы расширения (по своему содержанию, а не по реализации) частично заимствованы из традиционных языков имитационного моделирования.

В разработке средств учитывалась конструктивная работа проектировщиков со сложностью АС с позиций колмогоровской меры, ориентированной на «программное построение сложных систем». Использование в таких

построениях псевдо-языковых средств способствует пониманию планируемой и исполняемой работы, а также её потенциальному упрощению за счёт создания прототипов человеко-компьютерной деятельности.

ЗАКЛЮЧЕНИЕ

Представленные в статье средства управления потоками работ обеспечивают единообразный подхода к автоматизации согласованного управления потоками работ, задачами в потоках и действиями проектировщиков в решении нормативных задач на базе псевдо-кодowego программирования в инструментальной среде WQA.

Для программирования задач управления разработано расширение псевдо-кодowego языка L^{WQA} , позволяющее создавать программные модели, как для процессов решения задач, так и для управления их согласованным решением в потоках работ. Операторы расширения, по своему содержанию, а не по реализации, частично заимствованы из традиционных языков имитационного моделирования.

Разработанные средства управления обеспечивают конструктивную работу проектировщиков со сложностью АС с позиций колмогоровской меры, ориентированной на программируемое построение сложных систем. Использование в работе псевдо-языковых средств способствует пониманию планируемых и исполняемых работ, а также их потенциальному упрощению за счёт создания прототипов и моделей прецедентов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Кролл, 2004] Кролл П. Rational Unified Process – это легко: Руководство по RUP для практиков / П. Кролл, Ф. Крачтен – М.: КУДИЦ-Образ, 2004. – 427 с
- [Соснин, 2012] Соснин П.И. Программирование человеко-компьютерной деятельности. / П.И.Соснин. – Saarbruken: Lambert Academic Publishing, 2012 – 343 с.
- [Aalst et al., 2005] Aalst M., Hofstede A.H.M. and M. Dumas. Patterns of Process Modeling./ In Process-Aware Information Systems: Bridging People and Software through Process Technology, Wiley & Sons, (2005), pp 179-203.
- [Basili et al., 2001] Basili A. V., M. Lindvall M. and Costa P. Implementing the experience factory concepts as a set of experience bases, In Proc. of the 13 th International Conference on Software Engineering & Knowledge Engineering, (2001), 102-109.
- [Borges et al., 2012] Borges P., Machado R.J. & Ribeiro P. Mapping RUP Roles to Small Software Development Teams, In Proc. of International Conference on Software and System Process (ICSSP), Portugal, (2012), pp. 190-199.
- [Charette, 2005] Charette R.N. Why software falls, IEEE Spectrum, 42(9), (2005), 36-43.
- [IEEE, 2000] IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. Institute of Electrical and Electronics Engineers, Sept. 2000. IEEE Std 1471-2000
- [Li et al., 2008] Li M. and P. M.B.Vitanui, An Introduction to Kolmogorov Complexity and Its Applications. Series: Text in Computer Science, 3rd ed., Springer, 2008.
- [Held et al., 2009] Held M. and Blochinger W. Structured collaborative workflow design, Future Generation Computer Systems, 25(6), (2009), 638-653.

[Henninger, 2003] Henninger S. Tool Support for Experience-based Software Development Methodologies, *Advances in Computers*, 59, (2003), 29-82.

[Roglinger et al., 2012] Roglinger M., Poppelbuth J. & Becker J., Maturity models in business process management, *Business Process Management Journal*, Vol. 18 (2), (2012), pp. 328 – 346.

[Sosnin, 2011] Sosnin P. Question-Answer Shell for Personal Expert System // Chapter in the book “Expert Systems for Human, Materials and Automation.” Published by Intech, (2011), pp. 51-74.

[Sosnin, 2012a] Sosnin P. Question-Answer Approach to Human-Computer Interaction in Collaborative Designing. Chapter in the book “Cognitively Informed Intelligent Interfaces: Systems Design and Development” Published IGI Global, (2012), pp. 157-176.

[Sosnin, 2012b] Sosnin P. Pseudo-code Programming of Designer Activity in Development of Software Intensive Systems// In Proc. of the 25-th International conference on Industrial Engineering and other Applications of Applied Intelligent Systems (IEA/AIE 2012), Dalian, China, (2012), pp. 457-466.

[Sosnin, 2012c] Sosnin P. Experiential Human-Computer Interaction in Collaborative Designing of Software Intensive Systems// In Proc. of the 11-th International conference on Software Methodology and Technics (SoMeT'2012), Genua, Itali, (2012), pp. 180-197.

[Wang et al., 2010] Wang J. X . Kanban: Align Manufacturing Flow with Demand Pull, Chapter in the book: *Lean Manufacturing Business Bottom-Line Based*, CRC Press (2010), 185-204.

PROGRAM MANAGEMENT OF WORKFLOWS IN CONCEPTUAL DESIGNING OF AUTOMATED SYSTEMS

Sosnin P.I. *, Lapshov Y.A. *, Maklaev V.A.**

** Ulyanovsk State Technical University,
Ulyanovsk, Russia*

*sosnin@ulstu.ru
y.lapshov@ulstu.ru*

*** Federal Science-Production Centre “MARS”,
Ulyanovsk, Russia*

mars@mv.ru

The paper presents means and the language of pseudo-code programming of workflows, intended for algorithmic representation of business-processes in conceptual designing of automated systems (AS). The suggested approach facilitates the decreasing of the complexity in processes of designing.

INTRODUCTION

Programming of workflows is the useful way for reducing the complexity of the designer work in collaborative conceptual designing of automated systems (AS). Efficiency of collective works can be essentially increased if human activities in workflows will be based on executions of a special kind of programs by designers any of which plays a role of an “intellectual processor” when it can lead to positive effects. Such role is a model of a designer behavior providing the adjustment of workflows to the content of AS. This role is additional to the traditional set of roles used in development technologies of AS. The role implementation is being supported by specialized means of pseudo-code programming embedded to the toolkit WIQA (Working In Questions and Answers) the use of which is oriented on conceptual designing of AS.

Main Part

The successful creation of any AS in an essential measure depends on what means are used by designers for operated mastering by the system complexity. In general sense the complexity (or simplicity) of AS reflects the degree of difficulty for designers in their interactions with definite models of AS (or its components) in solving the definite tasks.

This paper presents new means for mastering of the complexity which are based on pseudo-code programming of workflows in collaborative conceptual designing. Programming of the conceptual activity helps to increase the level of its automation and by that to reduce the complexity of collaborative works and quantity of semantic defects in designer solutions. The suggested means manage the designer who should fulfill the role of I-processor working under control of QA-programs in decisions of typical project tasks.

Possibilities of QA-programming are implemented in the instrumental system WIQA, supporting the work of designers with the precedent base in the corporate network. Means of QA-programming include the interpreter of QA-programs for their execution by designers and the compiler of QA-programs for the computer execution.

The suggested pseudo-code language is defined and built as the object-oriented language the potential of which is sufficient for the expression of the necessary semantics. QA-programming is being applied as to the precedents already mastered by designer and for the precedents being created during designing.

Developed means support the creation of QA-programs for the management of workflows. To provide such possibility the traditional set of pseudo-code operators are evolved for programming the dynamics relations between tasks embedded to workflows. QA-programs of this type of are compiled to M²-programs which support the work with queues of project tasks.

Similar approach is used for the pseudo-parallel execution of appointed tasks by I-processor. The queue of interrupted tasks is interpreted as programs of M¹-type any of which combines conditional operators activated by events also.

CONCLUSION

The mastering of the complexity is the reliable way to increase the degree of the success in designing of AS. The useful means of this way is the program management of workflows in the conceptual stage of designing.

The complexity is being reduced because the library of programmed assets is the source of automated resources each of which can be included to the program of designer activity through calling the name of the asset. In WIQA-environment the assets are embedded to the repository which is a kernel of Experience Factory based on models of precedents.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.896

ТЕХНОЛОГИЯ ОТКРЫТОГО ПРОЕКТИРОВАНИЯ СХЕМ АВТОМАТИЗАЦИИ

Филатова Н.Н., Требухин А.Г.

*Тверской государственной технической университет,
г. Тверь, Российская Федерация*

nfilatova99@mail.ru

trebuchin2010@yandex.ru

Описаны основные компоненты интеллектуальной САПР схем автоматизации «Controlics». Приводится задача обобщения функциональных схем автоматизации, являющаяся предпосылкой к использованию технологий открытого проектирования. Описываются варианты реализации клиент-серверной архитектуры САПР схем автоматизации. Рассматриваются преимущества взаимодействия клиентов и сервера на основе построения виртуальной частной сети.

Ключевые слова: схема автоматизации; обобщение схем, клиент-серверная архитектура.

ВВЕДЕНИЕ

Проектирование системы автоматизации технологического процесса осуществляется на основе создания иерархически взаимосвязанных схем: структурная → функциональная → электрическая → монтажная.

С использованием математического аппарата теории автоматического управления определяется динамическая модель системы, схема которой включает множество блоков, заданных передаточными функциями, и связи между ними. Для получения модели структурной схемы осуществляется переход от динамических звеньев к атрибутивным моделям блоков, количество которых ограничено базовым набором. При построении функциональных схем решается задача назначения всем элементам структурной схемы блоков технических средств, которые могут реализовать соответствующие передаточные функции. Построение функциональной схемы означает нахождение конкретного элемента ТСА для каждого блока структурной и промежуточных элементов, необходимых для реализации каждой присутствующей в ней связи. В данном случае происходит идентификация всех вершин структурной схемы. Для одной структурной схемы можно построить несколько функциональных, а для одной функциональной - множество вариантов электрических, причем выбор любого из них не будет противоречить функциональным и эксплуатационно-техническим требованиям к системе автоматизации. В данном случае

проводится идентификация всех связей структурной схемы. На монтажной схеме по заданной принципиальной изображаются элементы, их соединители, зажимы и подводимые к ним концы проводов и кабелей.

Для создания схемной документации на системы автоматизации используются узкоспециализированные САПР (AutomatiCS ADT, САПР Альфа, CADElectro, SchematiCS и др.). В этих системах создание проекта системы автоматизации производится в интерактивном режиме, что ведет к проработке единственного варианта ее технической реализации.

Для повышения качества проектных решений необходимо расширить возможности САПР с помощью:

- средств автоматической генерации для заданной структурной схемы множества вариантов моделей функциональных схем. Идентификация всех вершин структурной схемы производится только на этапе генерации функциональных схем. От того, какие технические средства автоматизации (ТСА) будут подобраны в качестве истоков и стоков соответствующих цепей, напрямую зависит качество системы управления;
- механизма автоматического расширения набора правил построения функциональных схем, в том числе автоматического исключения из результатов генерации схем наименее перспективных вариантов. В некоторых современных САПР база данных технических средств содержит архив типовых проектных решений. Это приводит к хранению значительного

количества довольно малых по объему множеств схем, что затрудняет их использование. Поэтому необходимо разрабатывать подход, который позволит преодолеть описанное затруднение путем расширения множеств примеров технических решений из базы данных САПР специально сформированными выборками.

- использования открытых технологий проектирования. В частности, необходимо предусмотреть возможность работы удаленных пользователей с базой данных технических средств.

1. Интеллектуальная система проектирования схем автоматизации

В Тверском государственном техническом университете создана экспериментальная версия интеллектуальной системы «Controlics» для автоматизированного проектирования схем автоматизации. В состав системы включены базы данных и знаний. База данных [Ахремчик, 2009] включает в себя описание технических средств (наименование, функции преобразования, тип и диапазон используемых сигналов, привязка разъемов и контактов к входной и выходной составляющей каждой функции преобразования). Также в базе данных присутствуют правила генерации схем: эвристические правила выявления возможности создания функциональной связи между двумя элементами, правила построения контуров управления и др. Предусмотрена возможность сохранения выбранных проектных решений в специализированном формате. База знаний включает набор продукций, позволяющих оценить перспективность той или иной схемы.

Клиентское приложение включает в себя:

- редактор базы данных, который предназначен для добавления в базу данных новых технических средств и редактирования уже содержащейся в ней информации;

- редактор структурных схем. Служит для создания графической формы структурной схемы системы автоматизации и ее сохранения в специализированном формате;

- построитель задания на проектирование. В «Controlics» под заданием на проектирование понимается задание ограничений на область поиска в базе данных для каждого блока структурной схемы. Указание конкретного технического средства для каждого блока структурной схемы существенно ограничивает пространство поиска и количество допустимых альтернатив функциональных схем. Построение функциональных схем в условиях не полностью заданных требований к технической реализации предполагает решение указанной выше задачи на основе структурной схемы и задания по меньшей мере для одного ее блока хотя бы одной характеристики: для регуляторов может быть задан производитель, для датчиков – измеряемая физическая величина, принадлежность к

некоторому классу (термометр сопротивления, термопара и др.), ограничения на диапазон измеряемой величины и др.;

- генератор функциональных и принципиальных электрических схем. Предназначен для автоматического построения на основе задания на проектирование множества вариантов технической реализации структурной схемы в виде альтернатив функциональных схем. Методы и алгоритмы генерации описаны в [Филатова, 2011]. Предусмотрена возможность исключения из состава функциональной схемы определенного технического средства с обязательной передачей его функций оставшимся элементам, например, замена двух двухканальных блоков, у которых задействовано по одному каналу, одним двухканальным с задействованием всех его ресурсов. В автоматическом режиме для выбранной функциональной схемы возможно формирование принципиальной электрической. Это включает процесс привязки к каждой функциональной связи элементов наборов разъемов и контактов, построение цепей электропитания, проверка замкнутости цепей питания. Интерфейсные цепи строятся в диалоговом режиме;

- модуль обобщения схем. Если набор функциональных схем сгенерирован, то он будет включать множество технически реализуемых схем, состоящее из двух подмножеств: неэффективные с точки зрения проектировщика и остальные. Важной задачей является автоматическое выявление наименее эффективных схем и исключение их из набора схем, предоставляемых проектировщику для дальнейшего анализа. Указанная задача может быть решена обобщением схем автоматизации и созданием механизма автоматического формирования и расширения базы знаний САПР. На современном этапе развития интеллектуальных технологий достаточно хорошо развиты методы построения обобщений о множестве однотипных объектов, заданных набором дискретных признаков [Вагин и др., 2008]. В некоторых случаях схемы оцениваются набором качественных критериев [Емельянов, 1978]. Методика обобщения функциональных схем автоматизации описана в [Филатова, 2012]. В результате анализа обучающих выборок создаются продукционные правила $P_{iu} = \langle u, Q, Fv \rightarrow dk, R \rangle$, где u - индекс (порядковый номер, имя) правила в БЗ САПР, Fv - упорядоченный перечень связей, определяющих структуру измерительных или исполнительных цепей, Q – множество управляющих устройств, которые могут использоваться с Fv , $dk = \{1, 0\}$, если схема признана перспективной; 0, если схема признана неперспективной; R – постусловие продукции, определяющее необходимость проверки функциональной согласованности измерительных и исполнительных цепей канала управления (регулирования) схемы, построенной с помощью продукций;

- модуль анализа и выбора схем, предназначенный для оценки перспективности функциональных схем на основе правил *Pi*;
- модуль интеграции с внешними приложениями. С его помощью описания сформированных функциональных и принципиальных электрических схем экспортируются в MS Excel, AutoCAD и др.

2. Технология открытого проектирования схем автоматизации

Так как в результате обобщения схем автоматизации фактически формируются условия (правила) отнесения схемы к множествам перспективных или неперспективных, то интерес представляет распространение этих правил среди экспертов в области проектирования схем автоматизации для определения их достоверности и последующего применения на практике. Встает вопрос об организации технологии открытого проектирования схем автоматизации.

Архитектура «Controlics» реализована на основе технологии «клиент-сервер». База данных хранится на сервере и с помощью средств СУБД определяются разграничения на доступ к ней. Клиентское приложение представляет собой программный модуль, реализующий все функции по генерации схем на элементах из базы данных и обработке описаний этих схем. Возникает необходимость организации многопользовательского доступа к базе данных. Существует несколько приемлемых вариантов:

- при небольшом числе одновременных обращений к базе данных ТСА и хорошей пропускной способности каналов связи возможна одновременная работа всех клиентов в одной базе данных;
- репликация базы данных с сохранением ее на сервере и последующей работой с репликой. В этом случае каждый клиент работает с копией базы данных ТСА. Достоинства – каждому пользователю «своя» база данных, отсутствие необходимости на рабочих местах клиентов устанавливать СУБД. Недостатки – такие копии нужно создавать постоянно для поддержания информации о ТСА в актуальном состоянии, необходимость многократного экспорта сохраненных проектов из одной реплики и импорта в другую, постоянная значительная нагрузка на канал связи клиента и сервера;
- репликация базы данных с сохранением ее на рабочей станции клиента и последующей работой с репликой. Достоинства – каждому пользователю «своя» база данных, нагрузка на канал связи клиента и сервера только до момента завершения переноса реплики на рабочую станцию клиента. Недостатки – необходимость на рабочих местах клиентов устанавливать СУБД, реплики нужно создавать постоянно, необходимость

многократного экспорта сохраненных проектов из одной реплики и импорта в другую.

В любом случае, процесс пополнения базы данных информацией о новых ТСА должен производиться достаточно ограниченным количеством высококвалифицированных специалистов в режиме отключения от сервера всех клиентов.

На этапе проектирования серверной части САПР необходимо предусмотреть, чтобы в структуре базы данных технических средств отсутствовали таблицы, строки которых используются для временного хранения каких-либо данных.

Так как объем базы знаний САПР (на уровне хранения данных – одной таблицы) существенно меньше объема информации о технических средствах и содержание БЗ доступно только для чтения, то ее целесообразно выделить в отдельную базу данных и организовать доступ клиентов напрямую, без создания репликаций.

Клиент-серверное в «Controlics» взаимодействие реализовано с помощью технологии VPN (виртуальной частной сети). Применение в VPN криптографии позволяет использовать в неизменном виде транспортные протоколы (TCP, UDP).

VPN для удаленных пользователей представляет собой объединение отдельных машин или локальных сетей в виртуальной сети, которая обеспечивает целостность и безопасность передаваемых данных. Она обладает свойствами выделенной частной сети и позволяет передавать данные между двумя компьютерами через промежуточную сеть, например Интернет. Подключение удаленного пользователя к VPN производится посредством сервера доступа, который подключен как к внутренней, так и к внешней (общедоступной) сети. При подключении удаленного пользователя, либо при установке соединения с другой защищенной сетью, сервер доступа требует прохождения процесса аутентификации.

Если сетевая версия САПР используется в пределах одной локальной вычислительной сети, то использование VPN не обязательно. Однако, для повышения степени защищенности проектных данных возможно использование VPN, причем в роли сервера может выступать любая рабочая станция.

В случае интеграции разных локальных сетей с целью совместного доступа необходимо каждое рабочее место обеспечить широкополосным доступом к сети Интернет.

Достоинства VPN для использования в САПР схем автоматизации:

- в одну «виртуальную» сеть через Интернет можно объединить территориально разобщенные ЭВМ, физически расположенные в составе разных локальных сетей;

- сервер VPN одновременно является сервером базы данных ТСА;
- для корректной работы сервера через Интернет у провайдера необходимо заказать только внешний статический IP-адрес;
- с использованием VPN между ЭВМ создается безопасный (шифрованный) IP-туннель через единый UDP- или TCP-порт;
- режим аутентификации клиентов – по сертификатам, причем все сертификаты генерируются на сервере;
- возможность присвоения клиентам по их сертификатам статических IP-адресов, что позволяет создать гибкую систему распределения и учета доступа к базе данных ТСА.

ЗАКЛЮЧЕНИЕ

Создание механизмов автоматического расширения опыта САПР дало возможность формирования и пополнения архива наиболее и наименее перспективных схем. Для тестирования базы знаний САПР необходимо привлечение опытных экспертов, которые работают в разных проектных организациях. Возникает задача разработки технологий открытого проектирования схем автоматизации, которая решается с использованием клиент-серверной архитектуры САПР и выбора технических решений по организации доступа клиентов к базе данных технических средств.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Филатова, 2011] Филатова, Н.Н. Автоматическая генерация деревьев схемотехнических решений / Н.Н. Филатова, А.Г. Требухин, О.Л. Ахремчик // Труды международного конгресса по интеллектуальным системам и информационным технологиям «AIS-IT'11». М.: Физматлит, 2011. Т.2, С 122-130.
- [Ахремчик, 2009] Ахремчик, О.Л. Информационная база для автоматизированного проектирования схем систем управления технологическими объектами / О.Л. Ахремчик // Информационные технологии, 2009, № 8, С. 17-21.
- [Вагин и др., 2008] Достоверный и правдоподобный вывод в интеллектуальных системах / Вагин В.Н. [и др.]; – М.: Физматлит, 2008.
- [Емельянов, 1978] Системное проектирование средств автоматизации / Емельянов С.В. [и др.] – М.: Машиностроение, 1978. 190с.
- [Филатова, 2012] Филатова, Н.Н. Автоматическое обобщение вариантов технической реализации структурных схем систем автоматизации / Н.Н. Филатова, А.Г. Требухин // Труды международного конгресса по интеллектуальным системам и информационным технологиям «AIS-IT'12». М.: Физматлит, 2012. Т.1, С 174-181.

TECHNOLOGY OPEN DESIGNING OF CIRCUITS AUTOMATION

Filatova N.N., Trebukhin A.G.

Tver State Technical University, Tver, Russia

nfilatova99@mail.ru

trebuchin2010@yandex.ru

The basic components of the intelligent CAD «Controlics» have been considered. Describes the

benefits of client-server-based virtual private network. Provided to summarize the functional schemes of automation systems, which is a prerequisite for the use of open design technology.

INTRODUCTION

To create the automation systems circuit documentation uses highly specialized CAD. In these systems, the creation of automation system project is online, which leads to the elaboration of a single variant of its technical implementation. To improve the quality of design decisions must be empowered with the CAD tools for the automatic generation of a given set of variants of the block diagram models of functional circuits, a mechanism to automatically exclude the results of the generation of the least promising options, the use of open source design.

MAIN PART

In Tver State Technical University, an experimental version of intelligent system «Controlics» for automation schemes design. The system includes a database of technical equipment and knowledge. The client application includes a database editor, builder design assignment, the function and schematic diagrams generator, circuit generalization module, a module for integration with external applications.

There are several acceptable options for architecture: the simultaneous operation of all clients in a single database, database replication keeping it on the server and follow-up with a replica, the replication database, preserving its client workstation and follow-up with a replica.

Because as a result of the generalized scheme of automation actually creates the conditions (rules) classification scheme to a variety of promising or unpromising, the interest spread of these rules among experts in the field of design automation schemes to determine their validity and the subsequent application in practice. Client-server communication based on the technology VPN (virtual private network). Applications in cryptography VPN allows unchanged transport protocols (TCP, UDP).

The main advantages of VPN - the ability to control the number of simultaneous connections to the database and create an encrypted communications channel.

CONCLUSION

Creating a mechanism for automatic extension of CAD experience has enabled the formation and replenishment of the archive the most and least promising schemes. To test the knowledge of CAD is necessary to attract skilled experts who are working in different design organizations. The task of development technologies open circuit design automation, which is solved by using a client-server architecture, CAD and the choice of technical solutions for the organization of client access to database technology.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 658.004.89

СЕМАНТИКО-ОНТОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ ТЕХНИКО-ТЕХНОЛОГИЧЕСКОГО ПОТЕНЦИАЛА ПРЕДПРИЯТИЯ

Иванченко Н.А.* , Иванченко Г.Ф.**

**Национальный авиационный университет, г. Киев, Украина*

ivan730@ukr.net

***ГВУЗ “Киевский национальный экономический университет имени В. Гетьмана”, г. Киев, Украина*

ivan730@ukr.net

В статье представлен подход к построению семантико-онтологической модели знаний технико-технологической составляющей экономической безопасности предприятия. В качестве представления концептуализации этой составляющей используется онтология.

Ключевые слова: семантическая модель, представление знаний, модель, онтология, экономическая безопасность, *Protégé*.

ВВЕДЕНИЕ

Для современности характерным является постоянное изменение внешней среды, в которой работают, адаптируются и выживают люди, предприятия, организации и страны. В условиях конкуренции рыночная судьба предприятия во многом определяется скоростью и точностью реакции на смену внешней среды, а это требует применение в управлении новых не стандартных концепций, техники и инструментария.

Развитие компьютерных технологий дало возможность проектирования баз знаний (БЗ), которые организуют деятельность в предметных отраслях (ПРО) экономической безопасности предприятия (ЭкБП).

Технико-технологический потенциал (ТТП) является составляющей экономической безопасности предприятия. Он тесно связан с финансовой, инновационной и другими составляющими и, вместе с тем, имеет собственную специфику. Поэтому, решение интеллектуальных задач в отрасли ТТП требует учета большого объема информации.

Для создания БЗ в области ЭкБП необходимо описать семантико-онтологическую модель знаний.

Онтология ЭкБП – формальное явное описание понятий и показателей ТТП ЭкБП, которые описывают разные свойства и атрибуты понятий (слов) и ограничений, инкапсулированных в слоты. Онтология вместе с набором индивидуальных экземпляров классов образует семантическую модель БЗ. Онтология ЭкБП представляет совокупность соглашений, определения терминов, их толкование и содержит утверждения которые ограничивают смысл этих терминов, дает толкование утверждениям, а также методику расчета основных показателей.

Визуальные модели, например, онтологические владеют особенной когнитивной (то есть познавательной) силой. Визуализация онтологии позволяет специалистам по *knowledge engineering* непосредственно проектировать, формулировать и объяснить природу и структуру экономических процессов.

1. Описание предметной области.

Основной причиной циклических кризисов хозяйственных систем является износ основных производственных фондов. Данное правило действует и для отдельного субъекта системы – субъекта ведения хозяйства. В этой связи становится очевидная необходимость оценки технического состояния и движения основных фондов предприятия. Эта оценка нужна также для

планирования и создания амортизационного фонда предприятия, что формируется с целью накопления финансовых ресурсов, необходимых для обновления основных средств.

Технико-технологический потенциал ЭкБП субъекта ведения хозяйства зависит от технического и технологического уровня производства и определяет уровень продукции, ее конкурентоспособность и стоимостные расходы. Этот потенциал оценивается прежде всего уровнем технического совершенствования средств труда и технологии производства. При этом проявляются структура и состав технологического оборудования, его производительность, степень механизации и автоматизации, экстенсивное и интенсивное использование и т.д.

Таблица 1 – Индикаторы ТТП ЭкБП

Название индикатора потенциала	Допустимые значения индикаторов	Определение (расчет в модели)
1. Фондоотдача основных средств	$C_{1,1} > 0$	$C_{1,1} = \frac{Z_{3,10}}{C_{1,2}}$
2. Фондовооруженность труда	$C_{1,3} > 0$	$C_{1,3} = \frac{C_{1,2}}{C_{1,4}}$
3. Коэффициент выбытия основных средств	$C_{1,5} > 0$	$C_{1,5} = \frac{C_{1,6}}{C_{1,7}}$
4. Коэффициент обновления основного капитала	$C_{1,8} > 0$	$C_{1,8} = \frac{C_{1,9}}{C_{1,7}}$
5. Коэффициент износа основных средств	$C_{1,10} < 50\%$	$C_{1,10} = \frac{C_{1,11}}{C_{1,7}}$
6. Материалоемкость	$10\% << 30\%$	$C_{1,12} = \frac{C_{1,13}}{Z_{3,10}}$
7. Коэффициент полезного использования материалов	$C_{1,14} < 1$	$C_{1,14} = \frac{C_{1,13} - C_{1,15}}{C_{1,13}}$
8. Коэффициент брака	$1\% << 3\%$	$C_{1,16} = \frac{C_{1,17}}{q_{1,10}}$

Таблица 2 – Уровни безопасности ТТП ЭкБП

Название индикатора потенциала	Уровни безопасности		
	Низкий	Средний	Высокий
1. Фондоотдача основных средств	$C_{1,1} < 0$	$C_{1,1} = 0$	$C_{1,1} > 0$
2. Фондовооруженность труда	$C_{1,3} < 0$	$C_{1,3} = 0$	$C_{1,3} > 0$
3. Коэффициент выбытия основных средств	$C_{1,5} < 0$	$C_{1,5} = 0$	$C_{1,5} > 0$
4. Коэффициент обновления основного	$C_{1,8} < 0$	$C_{1,8} = 0$	$C_{1,8} > 0$

капитала			
5. Коэффициент износа основных средств	$C_{1,10} < 50\%$	$C_{1,10} = 50\%$	$C_{1,10} > 50\%$
6. Материалоемкость	$C_{1,12} < 10\%$	$C_{1,12} > 10\%$	$10\% < C_{1,12} < 30\%$
7. Коэффициент полезного использования материалов	$C_{1,14} = 0$	$C_{1,14} > 0$	$C_{1,14} < 1$
8. Коэффициент брака	$C_{1,16} < 1\%$	$C_{1,16} > 1\%$	$1\% < C_{1,16} < 3\%$

Где: $C_{1,1}$ – фондоотдача основных средств; $C_{1,2}$ – среднегодовая стоимость основных средств предприятия, грн.; $C_{1,3}$ – фондовооруженность труда; $C_{1,4}$ – среднесписочная численность работников; $C_{1,5}$ – коэффициент выбытия основных средств; $C_{1,6}$ – стоимость выбывших в отчетном периоде основных средств, грн.; $C_{1,7}$ – стоимость основных средств на начало отчетного периода, грн.; $C_{1,8}$ – коэффициент обновления основного капитала; $C_{1,9}$ – стоимость приобретенных в отчетном периоде основных средств; $C_{1,10}$ – коэффициент износа основных средств; $C_{1,11}$ – стоимость изношенности основных средств за соответствующий период, грн.; $C_{1,12}$ – материалоемкость; $C_{1,13}$ – материальные расходы, грн. $C_{1,14}$ – коэффициент полезного использования материалов; $C_{1,15}$ – стоимость отходов по ценам полноценного сырья, грн.; $C_{1,16}$ – коэффициент брака; $C_{1,17}$ – стоимость брака по себестоимости реализованной продукции, грн.; $Z_{3,10}$ – общая выручка; $q_{1,10}$ – полная себестоимость реализованной продукции.

Не менее важным направлением такой оценки становится выявление морального износа основных фондов, то есть их соответствия требованиям не только текущего производственного процесса, но и инновационной деятельности субъекта ведения хозяйства. Оценка производственных возможностей предприятия тесно связана с технологическим уровнем производства. Индикаторы ТТП ЭкБП представлены в таблице

Процесс обеспечения ТТП ЭкБП состоит из нескольких последовательных этапов:

1) анализ рынка технологий в производстве продукции, аналогичной профилю определенного предприятия или организации проектировщика системы: сбор и анализ информации об особенностях технологических процессов на предприятиях, которые изготавливают аналогичную продукцию; анализ научно-технической информации относительно новых разработок в определенной отрасли, а также технологий, способных осуществить интервенцию на отраслевой технологический рынок; формирование БЗ ТТП ЭкБП.

2) анализ конкретных технологических процессов и обнаружение внутренних резервов для

улучшения используемых технологий. Производится расчет индикаторов ТТП ЭкБП.

3) оценка перспектив развития рынков продукции предприятия и прогнозирование возможной специфики необходимых технологических процессов для выпуска конкурентоспособных инновационных товаров.

4) разработка технологической стратегии развития предприятия используя накопленные знания БЗ ТТП ЭкБП, в частности:

- обнаруживают перспективные товары;
- планируют комплекс технологий для производства перспективных товарных позиций;
- финансируют технологическое развитие предприятия на основе оптимизации расходов по программе технологического развития;
- разрабатывают общий план технологического развития предприятия;
- составляют план собственных корпоративных ресурсов в соответствии с планом технологического развития предприятия.

5) оперативная реализация планов технологического развития предприятия в процессе осуществления им производственно-хозяйственной деятельности.

6) анализ результатов практического использования мероприятий по обеспечению ТТП ЭкБП на основании специальной карты расчетов эффективности.

Кроме выше перечисленных индикаторов ТТП ЭкБП характеризуется следующими индикаторами:

- уровень прогрессивности технологий;
- уровень прогрессивной продукции;
- уровень технологического потенциала.

2. Построение модели.

Для представления знаний ЭкБП целесообразно использовать сетевую модель.

Семантика определяет смысл знаков и отношений между символами и объектами, какие они определяют.

Семантическая сеть ЭкБП – графическая система обозначений для представления знаний в шаблонах связанных узлов и дуг. Более формально: семантическая сеть – это ориентированный граф, вершины которого – понятие, а дуги – отношения между ними.

Сетевые модели ЭкБП формально можно задать в виде $H = \langle I, C_1, C_2, \dots, C_n, G \rangle$, где I – множество информационных единиц; C_1, C_2, \dots, C_n – множество типов связей между информационными единицами; G – отображения, которые находятся между информационными единицами и которые входят в I , связи из заданного набора типов связей.

Для всех семантических сетей ЭкБП общим является декларативное графическое представление, что может использоваться для представления

знаний или создания автоматизированных систем принятия решений на основе знаний.

Вершины могут являть собой: понятия, события, свойства. Метки вершин имеют ссылочный характер и являют собой некоторые имена объектов ПРО. Метки дуг помечают элементы множества отношений.

При использовании семантической сети для представления знаний важна классификация типов объектов и выделения некоторых фундаментальных видов связей между объектами. Независимо от особенностей среды, которая моделируется, можно допускать, что любая более-менее сложная его модель отображает какие-либо обобщенные, конкретные и агрегатные объекты.

Конкретный объект ЭкБП – это определенным образом выделенная единичная сущность.

Агрегатный объект ЭкБП – объект проблемной среды, составлен определенным образом из других объектов, которые являются его частями. Агрегатным может быть как обобщенный, так и конкретный объект.

В описанной типизации объектов проблемной среды определяются и связи между объектами.

Родовая связь может существовать между двумя обобщенными объектами. Использование наследования обеспечивает эффективный способ упрощения представления знаний и сокращения объема информации, которую нужно запоминать для каждого конкретного узла. Это дает возможность в значительной мере ускорить процесс обработки знаний, а также вытягивать информацию с помощью запросов общего характера.

Связь «является представителем» может существовать между обобщенным и конкретным объектами. Она имеет место в том случае, когда конкретный объект принадлежит классу с соответствующим обобщенным объектом.

Первым этапом построения онтологической модели является - карта корпоративных знаний ТТП ЭкБП представленная на рисунке 1.

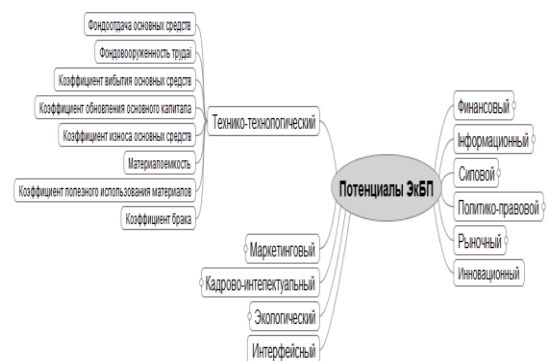


Рисунок 1 - Карта корпоративных знаний ТТП ЭкБП

Для построения семантико-онтологической и фреймовой модели ТТП ЭкБП использовано редактор онтологии *Protégé*, который позволяет проектировать онтологию, разворачивая иерархическую структуру абстрактных и конкретных классов и слотов. Результаты построения семантико-онтологической и фреймовой модели ТТП ЭкБП представлены на рисунках 2, 3.

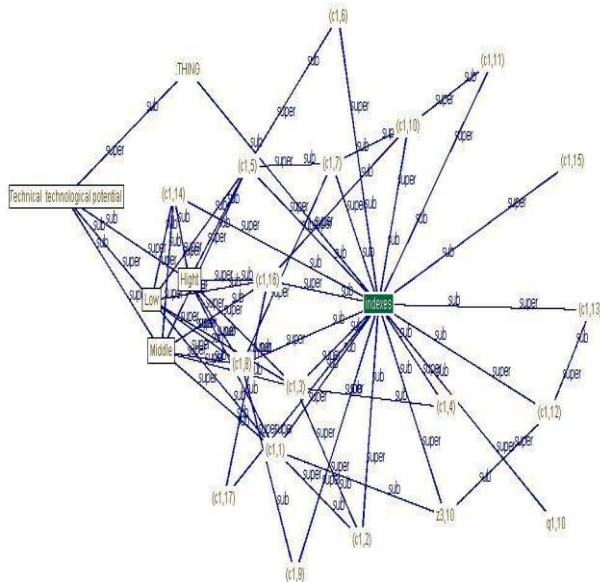


Рисунок 2 – Семантико-онтологическая модель ТТП ЭкБП

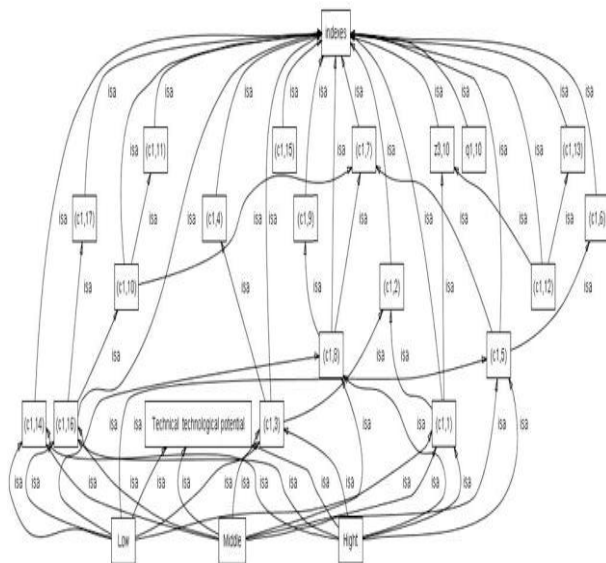


Рисунок 3 – Фреймовая модель ТТП ЭкБП

ЗАКЛЮЧЕНИЕ

Таким образом, семантико-онтологические модели предусматривается использовать в качестве концептуальных средств, помогающих проектировать и создать семантико-онтологическую БЗ ТТП ЭкБП которая предоставляет возможность разработки: надежного семантического базиса в определении содержания ТТП, общую логическую модель правил,

состоящую из словаря и набора утверждений на логическом языке, создающую основу для коммуникации между БЗ и компьютерными агентами с целью создания информационной системы управления ЭкБП.

Библиографический список

[Клещев А. С., Артемьева И. Л., 2001] Клещев А. С., Артемьева И. Л. Математические модели онтологий предметных областей. Часть 1. Существующие подходы к определению понятия «онтология». // Научно – техническая информация, серия 2 «Информационные процессы и системы», 2001, № 2, С. 20 – 27.

[Иванченко Г.Ф., 2011] Иванченко Г.Ф. Системы штучного интеллекта : навч. посіб. / Г.Ф. Иванченко. – К. : КНЕУ, 2011. — 382 с.

[Гаврилова Т.А., Хорошевский В.Ф., 2001] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб: Питер, 2001. – 384с.

SEMANTIC - ONTOLOGICAL DESIGN OF TECHNICAL - TECHNOLOGICAL POTENTIAL OF ENTERPRISE

Ivanchenko N. *, Ivanchenko G. **

* National aviation university, Kiev, Ukraine

ivan730@ukr.net

** Kiev national economic university name of V. Getmana ", Kiev, Ukraine

ivan730@ukr.net

In the paper a hike is represented to construction of semantic-ontological model of knowleges of technical-technological potential of economic safety of enterprise. As presentation of this constituent ontology is used.

INTRODUCTION

In the paper we represented the results of construction of semantic-ontological and frame model of technical-technological potential of economic safety of enterprise

MAIN PART

Visual models, for example, the ontological own the special cognitive force. Visualization of ontology allows to the specialists on knowledge engineering directly to design, formulate and account for nature and structure of economic processes.

For construction of semantic ontological and frame model of technical - technological economic safety of enterprise the editor of ontology of *Protégé*, which allows to design ontology, is used, opening out the hierarchical structure of abstract and concrete classes and slots.

CONCLUSION

In the paper the refined of technical - technological ontology's economic safety of enterprise.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ТРЕБОВАНИЯ К КОНСТРУКТОРУ ОНТОЛОГИЙ ДЛЯ МАШИНОСТРОЕНИЯ

Боргест Н.М., Коровин М.Д.

*Самарский государственный аэрокосмический университет имени академика С.П. Королева
(национальный исследовательский университет), г. Самара, Россия*

borgest@yandex.ru

maks.korovin@gmail.com

В работе рассмотрены ключевые особенности машиностроения как предметной области для построения онтологий. Проанализированы существующие стандарты построения онтологических моделей с учетом их применимости для машиностроительной отрасли. Сформулированы основные требования к конструктору онтологий для машиностроительной отрасли.

Ключевые слова: онтология; машиностроение; UML; W3C; OWL.

ВВЕДЕНИЕ

Онтологии определяются как совместно используемые формальные концепции конкретных предметных областей [Овдей и др, 2004].

Построение онтологий – сложный и длительный процесс. Создавая онтологию, пользователь может использовать языки представления онтологий или воспользоваться конструктором онтологий – приложением, поддерживающим создание и управление онтологиями, импорт/экспорт в разные форматы, доступ к библиотекам онтологий, визуализацию, машины вывода, языки запросов. Конструктор онтологий позволяет вносить изменения в систему специалистам без навыков программирования, а также без прерывания работы системы. Кроме этого на выбор того или иного конструктора онтологий оказывает существенное влияние предметная область и сфера деятельности. В статье рассматриваются несколько онтологических конструкторов и даются рекомендации по их применению в предметной области машиностроения.

1. СПЕЦИФИКА СОЗДАНИЯ ОНТОЛОГИИ МАШИНОСТРОИТЕЛЬНОГО ПРЕДПРИЯТИЯ

Современное машиностроительное предприятие представляет собой сложную систему, состоящую из множества элементов. Эффективное управление

предприятием, как совокупностью отдельных подсистем, постоянно взаимодействующих между собой, осложняется неопределенностью спроса и предложения, частым появлением нестандартных задач, требующих нетривиальных решений.

Зачастую решение текущих задач предприятия приводит к стихийному созданию горизонтальных связей между различными отделами предприятия, формируя динамически меняющиеся структуры.

В такой ситуации управление предприятием на основе классических методов менеджмента теряет свою эффективность. Необходимо обеспечения конкурентоспособности машиностроительного предприятия в условиях современного рынка заставляет искать новые подходы к управлению, где бы учитывался тот факт, что в функционирование предприятия происходит при участии людей, которые имеют собственное видение ситуации. Знания каждого человека, вовлеченного в процесс функционирования предприятия, используются им для принятия решений. Важно также учитывать, что в процессе решения проблемы знания действующих лиц могут дополняться и меняться. В условиях постоянно меняющейся информационной среды хорошую применимость имеет подход на основе интерсубъективной теории ситуационного управления [Скобелев, 2012]. В рамках данной теории в задачи онтологий машиностроительного предприятия входят описание предметной области, формирование моделей, пригодных для использования в системах поддержки принятия решений.

Интенсивность и объемы обмена информацией между подразделениями предприятия постоянно увеличивается. Направление основных потоков информации на предприятии [Бронникова и др., 1999]:

- по вертикали сверху вниз - плановая, нормативная, инструктивная, руководящая информация;
- по вертикали снизу вверх - аналитическая, рекомендательная, фактическая информация и запросы к руководству;
- по горизонтали - информация, обеспечивающая взаимосоординацию деятельности.

По сферам функционирования организации может быть выделено множество различных структур, соответствующих множеству самих видов деятельности. На предприятиях, например, как основные могут быть выделены, с одной стороны, технологическая, организационно-управленческая, экономическая, социально-психологическая структуры, с другой стороны - структура материальных и структура информационных потоков.

Технологические структуры представляют собой совокупность связей технологического процесса изготовления продукта предприятия, конструкторской и технологической подготовки производства, а также ремонтного, инструментального, энергетического и транспортного обслуживания.

Сложность и комплексный характер предметной области, как правило, не позволяют реализовать весь требующийся функционал в рамках одной онтологической модели, поэтому для решения задач предприятия обычно приходится создавать несколько взаимосвязанных онтологий, отличающихся по области применения, уровню обобщения, роли и степени формализованности языка (Рисунок 1) [Кудрявцев, 2006].

Для обеспечения эффективной работы всех вышеперечисленных структур и процессов на современных машиностроительных предприятиях применяется множество различных информационных систем, зачастую весьма слабо или вообще не интегрированных друг с другом.

В условиях гетерогенной информационной среды от конструктора онтологий требуется возможность валидации данных и обеспечение их единообразия по всему предприятию. В данном аспекте задача онтологии машиностроительного предприятия близка к задачам, обычно решаемым с помощью MDM-систем (систем управления мастер-данными) [Bedford, 2011]. Мастер-данные – это основные бизнес-данные, которые служат для принятия бизнес-решений. Задачей MDM является предоставление всем подразделениям предприятия доступа к одним и тем же актуальным данным для достижения высокой эффективности принятых

решений за счет повышения согласованности и качества информации.

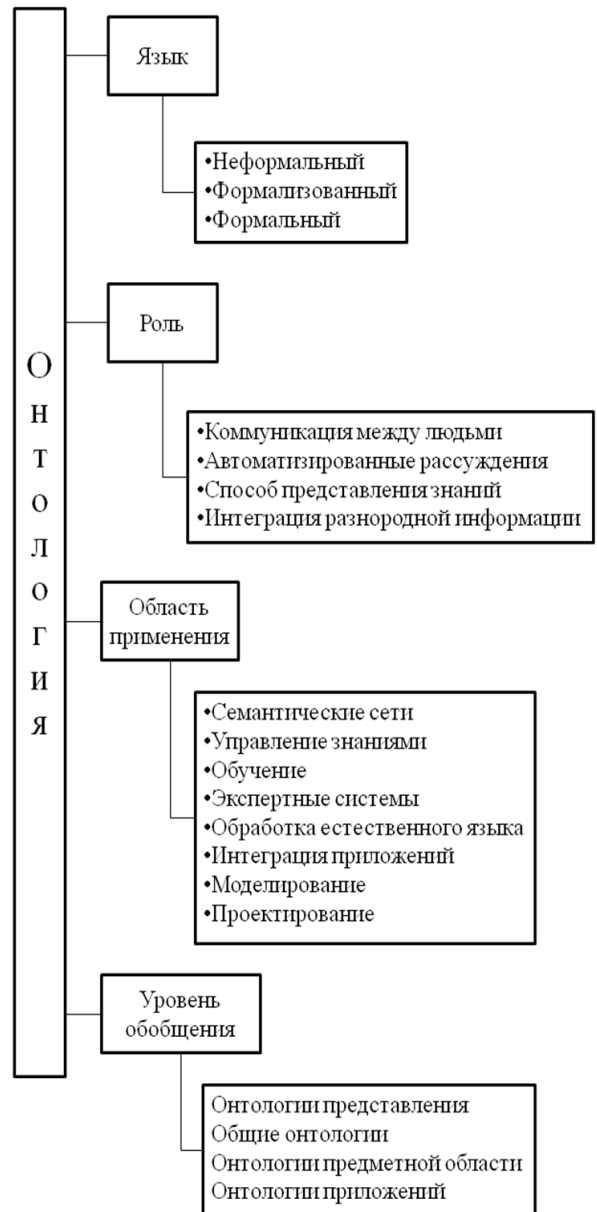


Рисунок 1 – Систематизация знаний в области онтологий

Онтологическая модель, создаваемая для предприятия, в идеале должна обеспечивать стандартизацию данных, поступающих от различных систем.

Информация, содержащаяся в онтологической модели, должна отвечать следующим требованиям:

- достоверность – получаемая информация должны быть верифицирована;
- релевантность – данные должны быть адекватными и помогать в принятии решений;
- понятность – информация должна предоставляться пользователю в наглядном виде;
- эффективность – эффект от получения информации должен перекрывать затраты ресурсов на её получение;
- регулярность – информация должна поступать регулярно.

Выполняемость этих требований закладывается в онтологическую модель на этапе её проектирования. Таким образом, функциональные характеристики конструктора онтологий в значительной степени определяют качество полученных онтологических моделей. Конструктор онтологий, обеспечивающий полную поддержку современных международных стандартов онтологического моделирования, позволяет создавать более эффективные онтологии, отвечающие высоким требованиям к качеству информации.

Стоит также отметить, что для предприятий, занимающихся мелкосерийным выпуском сложной продукции, важной особенностью производства является непрерывное изменение конфигурации продукции (постоянное совершенствование изделия), что требует от конструктора онтологий наличия функционала, позволяющего обеспечить описание всех версий и модификаций продукта, а так же полное и непротиворечивое задание назначения (применимости в конкретных модификациях) для каждого компонента изделия.

2. ОСНОВНЫЕ СТАНДАРТЫ ПОСТРОЕНИЯ ОНТОЛОГИЙ

Построение онтологии машиностроительного предприятия это сложный процесс, требующий как материальных, так и временных затрат. Инновационная деятельность предприятий имеет некоторые противоречивые особенности:

- Внедрение новаций в машиностроении очень редко приносит быстрый результат.
- Позднее внедрение новых технических и информационных средств грозит снижением конкурентоспособности продукции и большими финансовыми потерями.

Из вышесказанного можно сделать вывод, что для получения максимальной отдачи от внедрения результатов онтологического анализа на предприятии необходимо, чтобы процесс внедрения шёл как можно быстрее и создавал минимальное количество помех работе предприятия. Процесс создания онтологии значительно упрощается при возможности использования готовых базовых онтологий предметной области, поэтому для конструктора онтологий машиностроительного предприятия большим преимуществом будет являться наличие базовой онтологической модели машиностроительной области.

Сложность процессов на машиностроительном предприятии, развитая структура его подразделений и необходимость иметь максимально возможную репрезентативность онтологических моделей определяют требования к редактору онтологий поддерживать максимального количества классов, описанных в современных стандартах.

В настоящее время наиболее распространёнными стандартами являются W3C и

IDEF5. Стандарт IDEF5 подразумевает использование двух языков моделирования - IDEF5 Schematic Language (схематический язык) (SL) и IDEF5 Elaboration Language (язык доработок и уточнений) (EL).

SL в IDEF5 представляет собой наглядный графический язык, специально предназначенным для изложения экспертами в рассматриваемой области системы основных данных в форме онтологической информации. Язык SL используется только на первом этапе моделирования, структурирование информации с использованием этого языка сложно «механизировать», предполагается, что схематические диаграммы в данном случае создаются людьми.

Анализ данных и исследование полноты данных, полученных в результате построения онтологической структуры предметной области, являются задачей текстового языка EL.

Стандарт IDEF5 создавался специально для графического моделирования онтологий, однако, на сегодняшний день он имеет сравнительно мало инструментальных средств поддержки и, соответственно, существующих способов взаимодействия с другими методологиями. Частично стандарт IDEF5 можно моделировать средствами UML [Андреева, 2008], но не все элементы IDEF5 имеют прямые аналоги, при этом некоторые элементы в определенной ситуации могут иметь несколько аналогов.

Современный подход к проектированию онтологий предполагает построение web-онтологий. Стандарты, описывающие этот тип онтологий поддерживает The World Wide Web консорциум (W3C).

Одним из основных продуктов W3C является язык Web Ontology Language (OWL). Задачей языка OWL является представление онтологий и связанной информации в виде семантической сети. Язык разметки OWL построен на основе XML, что обеспечивает ему хорошую совместимость с web-ориентированными языками, а так же широкие возможности импорта-экспорта моделей в приложения, работающие с UML [Добров и др., 2008].

Язык OWL поддерживает следующие элементы онтологий:

- классы;
- иерархии классов;
- свойства;
- иерархии свойств;
- домены (области определения);
- диапазоны свойств;
- индивиды (представители свойств).

Достоинства языка OWL снизили ему широкую популярность среди разработчиков как конструкторов онтологий, так и среди специалистов, занятых созданием онтологических моделей. Возможность выбора из большого

количества редакторов онтологий и совместимость между ними делает стандарт W3C предпочтительным на фоне практического отсутствия программных средств для реализации онтологических моделей по стандарту IDEF5.

3. КРИТЕРИИ ВЫБОРА КОНСТРУКТОРА ОНТОЛОГИЙ ДЛЯ МАШИНОСТРОИТЕЛЬНОГО ПРЕДПРИЯТИЯ

Обобщая представленный анализ предметной области и современных средств её онтологического описания можно сделать вывод, что основными критериями или фактически требованиями при выборе конструктора онтологий для машиностроительного предприятия могут быть:

- поддержка современных стандартов моделирования онтологий;
- поддержка конструктором всех классов, необходимых при создании моделей онтологий;
- возможность внесения изменений в онтологическую модель без необходимости глубокого реинжиниринга модели;
- возможность реализации в онтологической модели функций валидации и очистки информации, поступающей в систему с последующей её выдачей для лиц, принимающих управленческие решения;
- актуальность версий и поддерживаемость программного продукта для создания онтологических моделей;
- способность создавать модели, адекватно работающие в условиях гетерогенной высоконагруженной информационной среды;
- возможность строить различные модели онтологий в рамках одного программного продукта;
- по возможности низкая стоимость лицензий.

ЗАКЛЮЧЕНИЕ

Рассмотрены основные особенности предметной области «машиностроительное предприятие» с точки зрения выбора конструктора онтологий. Проанализированы стандарты описания онтологических моделей. Предложены рекомендации к выбору конструктора онтологий для предприятия машиностроительной отрасли.

Работа выполняется в рамках Государственного контракта № 07.524.12.4022 от 11.07.2012 года.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Андреева, 2008] Андреева Н.В. Выбор методов и средств онтологического анализа стандартов информационной безопасности, Санкт-Петербург, 2008

[Скобелев, 2012] Скобелев П.О. Онтология деятельности для ситуационного управления предприятием [Текст] научный журнал «Онтология проектирования» №1, 2012, с. 7 – 38.

[Бронникова и др., 1999] Бронникова Т.С., Чернявский А.Г. Учебное пособие для студентов экономических специальностей,

Таганрогский государственный радиотехнический университет, Таганрог, 1999

[Добров и др., 2008] Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д. Онтологии и тезаурусы: модели, инструменты, приложения: учебное пособие М: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2012. – 173 с.

[Кудрявцев, 2006] Кудрявцев Д.В. Обзор применения онтологий в моделировании и управлении. //Фрагмент отчета по НИР «Эталонные модели организации деятельности в государственном секторе», выполненной сотрудниками АНО КМЦ «Бизнес-Инжиниринг» совместно с ИПГМУ ВШЭ, 2006 г.

http://bigc.ru/theory/experience/ontologies_for_modelling.php.

[Овдей и др, 2004] Овдей О.М., Проскудина Г.Ю. Обзор инструментов инженерии онтологий/ О.М. Овдей [и др.] //Электронные библиотеки. 2004 - Том 7 - Выпуск 4.

[Bedford, 2011] Denise A.D. Bedford Role of ontologies in master data management, Kent State University, Kent OH, 2011.

REQUIREMENTS FOR ONTOLOGY CONSTRUCTORS IN THE ENGINEERING PLANT SUBJECT AREA

Borgest N.M., Korovin M.D.

*Samara State aerospace University, Samara,
Russian Federation*

borgest@yandex.ru

maks.korovin@gmail.com

The paper describes the main factors that should be taken into account when choosing an ontology constructor for the engineering plant. Main complications of the subject areas are shown. The standards, most commonly used in the ontology modelling, are analysed.

INTRODUCTION

Ontology model is a considerable asset of a modern engineering plant. One of the most decisive factors of the model quality is the ontology constructor used to build the model as it defines many key characteristics of the model implementation process and of the model itself.

MAIN PART

The main characteristics of the ontology construction systems that should be taken into account when choosing one for a engineering plant:

- Implementation of the most commonly used standards.
- Compatibility with other ontology models.
- Actuality and customer support for the particular ontology modeling software.

CONCLUSION

The paper describes the main factors that should be taken into account when choosing a specific ontology constructor for an engineering plant.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПОДХОДЫ К ИНТЕГРАЦИИ ПРОГРАММНЫХ СИСТЕМ В ИНТЕЛЛЕКТУАЛЬНОМ ПОМОЩНИКЕ ПРОЕКТАНТА

Боргест Н.М., Коровин М.Д.

*Самарский государственный аэрокосмический университет имени академика С.П. Королева
(национальный исследовательский университет), г. Самара, Россия*

borgest@yandex.ru

maks.korovin@gmail.com

В работе рассмотрены основные подходы к интеграции программных систем с точки зрения их применимости к разработке интеллектуального помощника проектанта, реализуемого в предметной области предварительного проектирования самолета. Показана возможность создания программного комплекса, решающего типовые задачи проектирования в автоматическом режиме.

Ключевые слова: интеллектуальный помощник проектанта; интеграция; метамодель; тезаурус; матрица проекта.

ВВЕДЕНИЕ

Современный процесс проектирования базируется на основе применения множества программных средств (промышленные САПР, СУБД, конструкторы онтологий и др.), между которыми существуют сложные алгоритмы взаимодействия. Для развития и совершенствования программной системы, а также упрощения её взаимодействия с пользователем необходима выработка общих принципов интеграции и интерфейсного взаимодействия. Наиболее эффективным методом решения данной проблемы является интеграция программных средств, которая учитывает непрерывное развитие систем и максимальное приближение интерфейса к естественному для пользователей и разработчиков.

1. ПРОБЛЕМЫ ИНТЕГРАЦИИ ПРОГРАММНЫХ СИСТЕМ В ИНТЕЛЛЕКТУАЛЬНОМ ПОМОЩНИКЕ ПРОЕКТАНТА

Одним из ключевых условий эффективной работы интеллектуального помощника проектанта (ИПП) является обеспечение согласованной совместной работы всех его компонентов. Данная проблема решается интеграцией программных компонентов в единую среду. Можно выделить следующие проблемы интеграции систем:

- Ускорение процессов. В динамичных областях, где “изменчивость” является самой сутью

и природой системы, задача интеграции усугубляется и превращается в серьезную проблему.

- Распределенность. Задачи, решаемые при проектировании, становятся все более комплексными, появляется логическая, организационная и географическая рассредоточенность [Бейлезон, 2010].

- Гетерогенность. При проектировании сложных технических систем, например, самолёта, нет возможности придерживаться единой платформы и полностью совместимых инструментов, поэтому приходится учитывать и поддерживать особенности нескольких платформ и искать пути взаимодействия разнородных инструментов [Шемсединов, 2011].

- Наследственность. Невозможность полностью отказаться от устаревших технологий, старого аппаратного обеспечения, которые хотя и дают вполне приемлемые показатели по надежности и производительности, никак не способствуют интеграции [Лондон и др, 2005].

- Хаотичность. Не всегда есть возможность полностью формализовать, специфицировать и структурировать данные, и часть модели процесса проектирования остается “слабосвязанной”, не поддающейся или слабо поддающейся машинной обработке, анализу, индексации и пр.

- Интерактивность. Проектант, как потребитель информации, постоянно повышает свои ожидания о скорости реакции системы, быстродействии и оперативности доставки

информации. Большинство процессов стремятся к выполнению в реальном времени.

Рассматривая гетерогенность компонентов ИПП можно выделить следующие проблемы:

- Несовместимость форматов обмена данными, различие интерфейсов. Данная проблема обычно решается путем внедрения в систему модулей, предназначенных для обмена данными внутри системы (брокеров, конвертеров).
- Концептуальная разница между компонентами системы – в случае если разработчики разных систем приняли решения, допущения и предположения, концептуально не стыкующиеся между собой. Проблема концептуальной несовместимости может решаться добавлением дополнительного уровня абстракции, которое не противоречит обоим подходам. При этом осуществляется отделение деталей реализации конкретного множества функций от их способа использования.

2. ДВА ПОДХОДА К ИНТЕГРАЦИИ

В настоящий момент существует два основных подхода к интеграции:

- Построение централизованной структуры, где интегрируемые системы становятся подсистемами, а взаимодействие происходит через единую точку (единую сервисную шину). (рисунок 1) В этом случае модули-подсистемы осуществляют обмен информацией по стандартизованным алгоритмам, управляемым из единого центра контроля.
- Создание структуры с брокером (посредником, не являющимся центром). Системы остаются независимыми, брокер при этом обеспечивает взаимодействие между ними (рисунок 2).

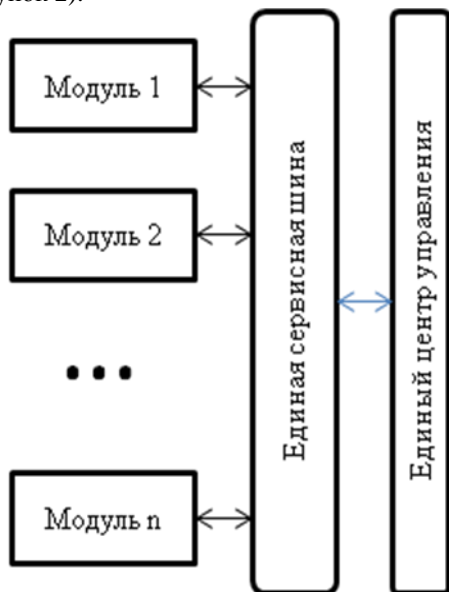


Рисунок 1 – Централизованная структура на основе единой сервисной шины

К преимуществам централизации с использованием единой сервисной шины

относятся:

- масштабируемость — возможность строить решения любого размера и информационной нагрузки;
- гибкость — возможность реализовывать и изменять интеграционные сценарии с минимальным вовлечением разработчиков системы;
- безопасность — встроенные средства аутентификации и авторизации обеспечивают контроль доступа к сервисам на уровне самой шины, избавляя разработчиков интеграционных сценариев от задач по реализации этих механизмов;
- использование открытых стандартов — позволяет уменьшить вовлеченность дорогостоящих специалистов по проприетарным технологиям;
- централизация средств контроля и администрирования — позволяет избежать «размытия» точки ответственности за интеграционные сценарии, обеспечить оперативное наблюдение и раннее оповещение в случае сбоев [Mason, 2011].



Рисунок 2 – Интеграция с использованием брокеров

Интеграция с использованием брокеров хорошо работает для небольших систем, где количество компонентов ограничено и не предполагается добавление дополнительных подсистем. Большим недостатком такого подхода является резкое возрастание необходимого количества брокеров. Так для системы из 3 компонентов достаточно 3 брокеров, а при добавлении всего 2 дополнительных модулей количество необходимых брокеров возрастает до 10. При этом каждый модуль уникален и требует индивидуального подхода. Данный недостаток делает интеграцию такого типа трудноприменимой для систем с большим количеством компонентов, где необходимы непосредственные связи между всеми компонентами. Стоит отметить, что для систем с относительно небольшим количеством модулей, которые соединены между собой последовательно и имеют ограниченное количество обратных связей (рисунок 3), метод интеграции по типу узел-узел все еще остается наиболее привлекательным, так как в этом случае затраты на построение интегральной структуры могут превысить выгоду от её внедрения.



Рисунок 3 – Пример системы, для которой эффективна интеграция с использованием брокеров

3. ИНТЕГРАЦИЯ КОМПОНЕНТОВ В ИНТЕЛЛЕКТУАЛЬНОМ ПОМОЩНИКЕ ПРОЕКТАНТА

Интеллектуальный помощник проектанта – сложная система с развитой структурой. В силу того, что компонентами данной системы могут являться как самостоятельные программные продукты (third party applications), так и модули, специально разработанные для нее, не во всех случаях можно обеспечить семантическую совместимость всех компонентов без внедрения в систему дополнительных модулей – брокеров.

В силу особенностей предметной области ИПП разработка структур баз данных, где для всех сущностей описываются только функциональные свойства, сильно затруднена.

Сложноструктурированная предметная область приводит к огромному числу таблиц и связей, при этом динамичность предметной области требует постоянного реинжиниринга структуры базы данных. Подъем уровня абстракции решает задачу лишь частично, так как при переходе к абстрактной модели теряется специфика предметной области, а значит, что нужно хранить в одной базе данных два логически связанных слоя, один для структурных свойств модели, второй для функциональных. Роль связки должен играть мета-слой, определяющий параметры взаимно однозначного отображения одного абстрактного слоя в другом.

Таким образом, для эффективной работы ИПП необходимо построение метамодели – модели предметной области, описанной на нескольких уровнях абстракции, где каждый высший уровень полностью, целостно и непротиворечиво задает структуру данных, функциональность, отображение и связи нижних уровней [Tarscott, 2008].

Метамодель описывает не отдельную задачу, а широкий круг задач с выделением в этих задачах общих абстракций, правил обработки данных и управления процессом проектирования. К конкретной задаче метамодель адаптируется в процессе исполнения.

В процессе функционирования данные, логика и визуализация должны динамически (в реальном времени) интерпретироваться из метамодели предметной области, что позволяет модифицировать структуры хранения данных без внесения

изменений в программный код систем.

Важной является также возможность создавать несколько параллельно существующих представлений одной модели, что повышает наглядность результата.

Для обеспечения взаимодействия внутри системы необходимо обеспечивать глубокую семантическую интеграцию между компонентами [Голенков и др., 2012]. В ИПП это реализуется с помощью развитого тезауруса предметной области, который связан с матрицей проекта.

Цели использования тезауруса:

- обеспечения общей терминологии для предметной области с целью совместного использования всеми пользователями;
- получение точных и непротиворечивых определений каждого термина;
- обеспечение задания семантики с помощью множества аксиом, которые автоматически позволяют получать ответ на множество вопросов о предметной области [Боргест, 2011].

Матрица проекта (МП) содержит в себе всё описание объекта, которое соответствует определенному этапу его проектирования [Боргест, 2011]. Структура МП контурно напоминает морфологическую таблицу, позволяющую вместить все возможные реализации проектируемого объекта. В момент начала работы проектанта с системой матрица пуста. Заполнение матрицы начинается с данных технического задания на проектирование. По мере заполнения МП система осуществляет запуск соответствующих модулей и, если вмешательство проектанта не требуется, происходит дозаполнение матрицы в автоматическом режиме (рисунок 4). Появление данных в матрице инициирует выполнение сценариев её заполнения сообразно текущей задаче. Когда в матрице появляется достаточно данных для выполнения какого-либо модуля, он начинает выполняться. Таким образом, выполнение независимых модулей может идти параллельно. Структура МП формируется динамически, то есть количество и длина строк МП являются переменной величиной, зависящей от степени детализированности текущей интерпретации метамодели. На рисунке 4 представлены несколько возможных последовательных состояний матрицы проекта в процессе её заполнения.

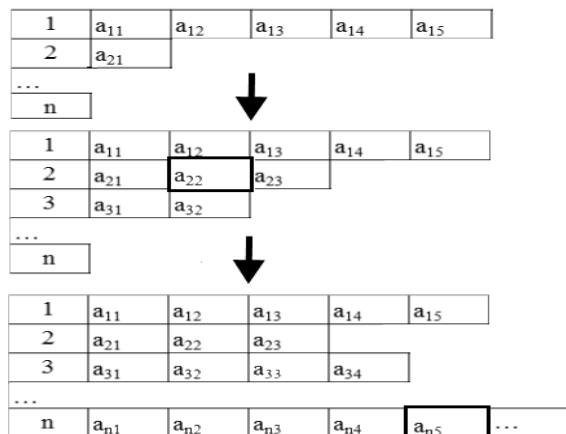
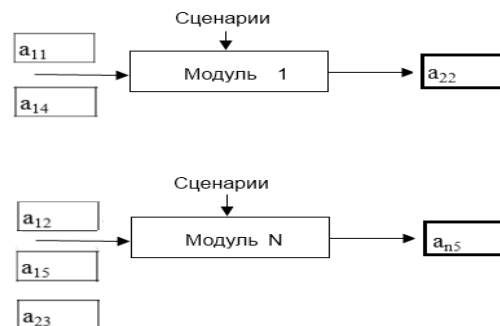


Рисунок 4 – Пример варианта заполнения матрицы проекта



ЗАКЛЮЧЕНИЕ

Рассмотрен прототип структуры интеллектуального помощника проектанта, применяемого в предметной области предварительного проектирования самолета. Реализация подобной системы позволит автоматизировать типовые задачи проектирования, наглядно представляя проектанту весь ход выполнения работ над проектом.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Бейлезон, 2010] Бейлезон, О. Подходы к интеграции приложений Enterprise Service Bus, Компьютер пресс 6'2010.
 [Боргест, 2011] Боргест, Н.М. Модель интеллектуального интерфейса робота «конструктор самолета» / Н.М. Боргест // XI международная научная конференция имени Г.А. Таран «Интеллектуальный анализ информации ИАИ-2011», Киев, Просвіта, 2011, с. 297-302.

[Боргест, 2011] Боргест, Н.М. Онтология проектирования самолета/ Н.М. Боргест // Искусственный интеллект, 2011, №4. с.260-265.

[Голенков и др., 2012] Голенков, В.В. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования// Голенков В.В., Гулякина Н.А. // Материалы международной научно-технической конференции OSTIS-2012, Минск, БГУИР, 2012 - с.23-52

[Лондон и др, 2011] Лондон Дж., Лондон К. Управление информационными системами. – 7-е изд., сер. «Классика МВА» / Пер. с англ. под ред. Д.Р. Трутнева – СПб.: Питер; 2005 – 912 с.

[Шемседдинов, 2011] Шемседдинов Т. Meta Modelling, 2011 <http://blog.meta-systems.com.ua/02.12.2012>

[Mason, 2011] Ross Mason Understanding Enterprise Application Integration - The Benefits of ESB for EAI, 2011

[Tapscott, 2008] Don Tapscott Grown Up Digital: How the net generation is changing your world, 2008

APPROACHES TO THE INTEGRATION OF SOFTWARE SYSTEMS IN THE INTELLECTUAL DESIGNER ASSISTANT

Borgest N.M., Korovin M.D.

Samara State aerospace University, Samara,
 Russian Federation
 borgest@yandex.ru

maks.korovin@gmail.com

The paper describes the main approaches to the integration of software systems in terms of their applicability to the usage in the intellectual designer assistant, that is implemented in the preliminary design of the aircraft domain. The possibility to create a software package capable of solving typical complex designing problems in automatic mode is shown.

INTRODUCTION

The modern design process involves a large number of various software systems (CAD, DBMS, etc). The interactions between those systems grow more and more complicated. To increase the efficiency of the process a unifying approach is proposed.

MAIN PART

To solve the problem of the creating a unified design software complex we suggest using the following proposals:

- Module structure, that allows to add and remove components without the need to reengineer the whole system.
- Usage of the meta-model as a basis for creating different models based on various interpretations of the meta-model.
- Deep semantic integration between the components of the system based on a complex thesaurus of the subject area.
- Matrix of the project as a storage for the design variables of the project. The structure of the matrix of the project dynamically changes to represent the detail level of the current meta-model interpretation.

CONCLUSION

The paper presents an approach to create a prototype of an intellectual designer assistant, applied to the subject area of the preliminary design of an aircraft. The realization of such a system will allow automating of typical designing tasks.



УДК 004.822:514

ПРОБЛЕМЫ СИНОНИМОВ В ТЕЗАУРУСЕ ИНТЕЛЛЕКТУАЛЬНОГО ПОМОЩНИКА ПРОЕКТАНТА

Боргест Н.М., Шустова Д.В., Одинцова С.А., Князихина Ю.Е.

Самарский государственный аэрокосмический университет имени академика С.П. Королева (национальный исследовательский университет), г. Самара, Россия

borgest@yandex.ru

ShustovaDV@yandex.ru

sv-odincova@mail.ru

jul4432@yandex.ru

В работе рассматриваются проблемы, возникающие при создании тезауруса предварительного проектирования самолета в рамках проекта создания интеллектуального помощника проектанта. Приведены результаты разработки пилотской версии тезауруса с использованием предложенных решений.

Ключевые слова: информационно-поисковый тезаурус, эквивалентные классы, синонимы, многозначность.

ВВЕДЕНИЕ

В русском языке, как и во многих других, есть большое количество многозначных слов, с которыми человек сталкивается в повседневной жизни. Вследствие различной формулировки и

длительный промежуток времени в различных предметных областях. Для описания терминов используется разрабатываемая онтология – система, состоящая из наборов понятий и

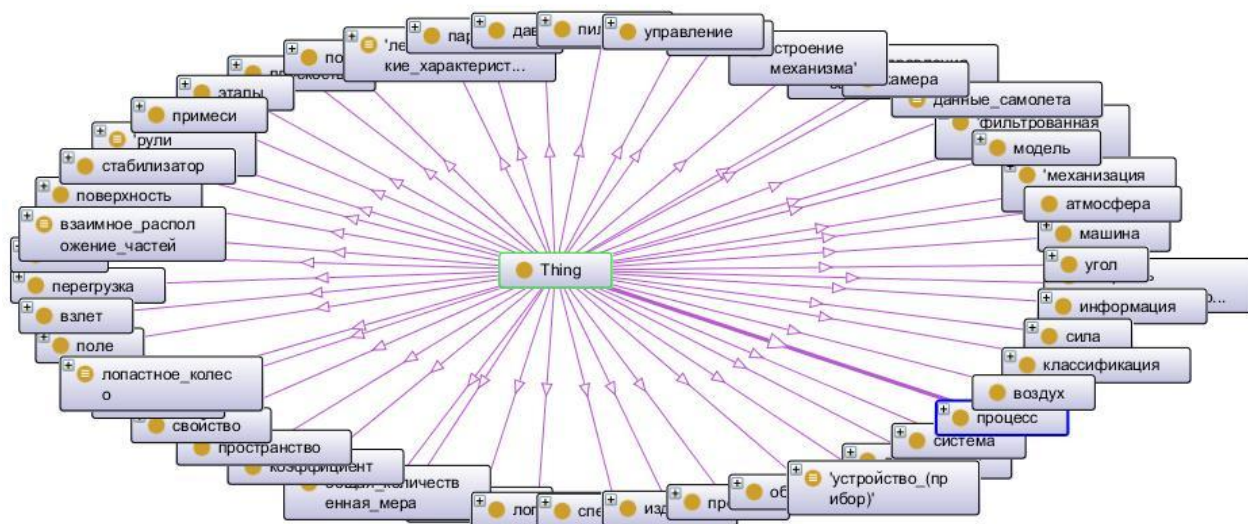


Рисунок 1- Основные классы тезауруса ПрО «Предварительное проектирование самолета»

интерпретации одних и тех же терминов, существует потребность в формировании словника, который даст единые и исчерпывающие знания о конкретном объекте. Разработка информационно-поискового тезауруса позволяет структурировать и унифицировать знания, накопленные за довольно

утверждений об этих понятиях, на основе которых можно описывать классы, отношения, функции и индивиды [Добров и др., 2009]. В данном случае, в качестве предметной области рассматривается этап предварительного проектирования самолета.

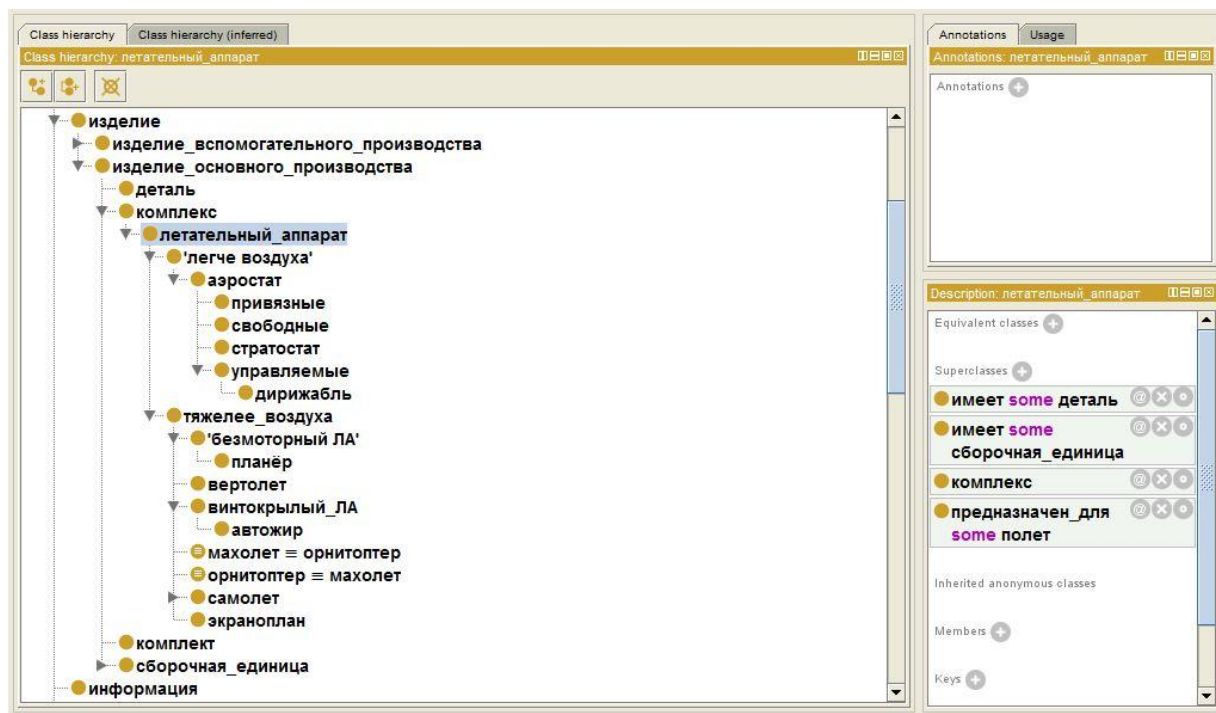


Рисунок 2 – Ассоциативные связи класса «Летательный аппарат»

Тезаурус представляет собой систематизированную совокупность понятий определенной отрасли науки, отражающих логические связи между терминами. Эти связи основываются на классовой иерархии, родо-видовых и ассоциативных связях. Процесс визуализации такой системы упрощает понимание структуры рассматриваемой области.

На рисунке 1 показаны основные классы, входящие в состав тезауруса предварительного проектирования самолета.

Разработка тезауруса осуществляется на базе онтологического редактора Protégé, созданного в Стэнфордском университете. Несмотря на то, что Protégé, как и большинство конструкторов онтологий, представлен в англоязычной версии, он имеет интуитивный интерфейс, доступный подготовленному пользователю. Однако это не защищает последних от ряда проблем и ошибок при его использовании.

В первую очередь, логическое построение структур влияет на восприятие пользователем информационно-поискового тезауруса. Это является следствием того, что у каждого человека существует свое собственное понимание окружающего мира и предметной области.

Таким образом, несмотря на то, что разрабатываемый тезаурус ориентирован на использование различными пользователями, и, казалось бы, должен быть универсальным и контекстно адаптирован к пользователям разной профессиональной культуры, но в конечном итоге он является весьма субъективным проектом. Существующие терминологические ГОСТы, ОСТы, профессиональные энциклопедии и словари

лишь частично снимают вопрос о субъективности «готового продукта».

Во-вторых, сама по себе классовая иерархия не является универсальной, так как далеко не все термины можно описать с помощью отношений типа род-вид. Зачастую приходится использовать дополнительные ассоциативные связи (см. рисунок 2) для доопределения слов. Но и это не является гарантом того, что можно будет увидеть полную картину создаваемой предметной области.

Ранее, в публикации [Боргест и др., 2012] рассматривалась проблема задания иерархических и ассоциативных связей в тезаурусе. Было принято решение включения вспомогательных слов для установления связей ассоциации. Тезаурус концептуального проектирования самолета является инновационным проектом, поэтому процесс его создания сопровождается выявлением проблем, не рассматриваемых ранее. В ходе разработки проекта было выявлено несколько важных вопросов.

ДОБАВЛЕНИЕ СИНОНИМОВ В ТЕЗАУРУС

Ограниченные возможности самого онтологического редактора Protégé устанавливают определенные рамки при его использовании. Например, в редакторе исключена возможность создания одинаковых терминов. С одной стороны, это помогает избежать повторного добавления слова. Этот фактор имеет большое значение в том случае, когда в тезаурус входит большое число терминов, так как не нужно проверять каждый добавляемый термин с помощью функции поиска, и таким образом, ускорить процесс работы. С другой стороны, отсутствие возможности создания

одинаковых терминов выявляет определенные трудности в случае добавления синонимов и омонимов.

В случае однозначности рассматриваемых терминов, в Protégé предусмотрена возможность создания эквивалентных классов, то есть значения этих слов приравниваются друг к другу. Но более

В случае задания термина «строение механизма» как синонима термина «устройство» получаем эквивалентный ряд: «прибор = устройство = строение механизма». Но при этом образуется нарушение логической цепочки, так как приравнивание терминов «прибор» и «строение механизма» не является верным. При поиске решения этой проблемы просматривались сайты и

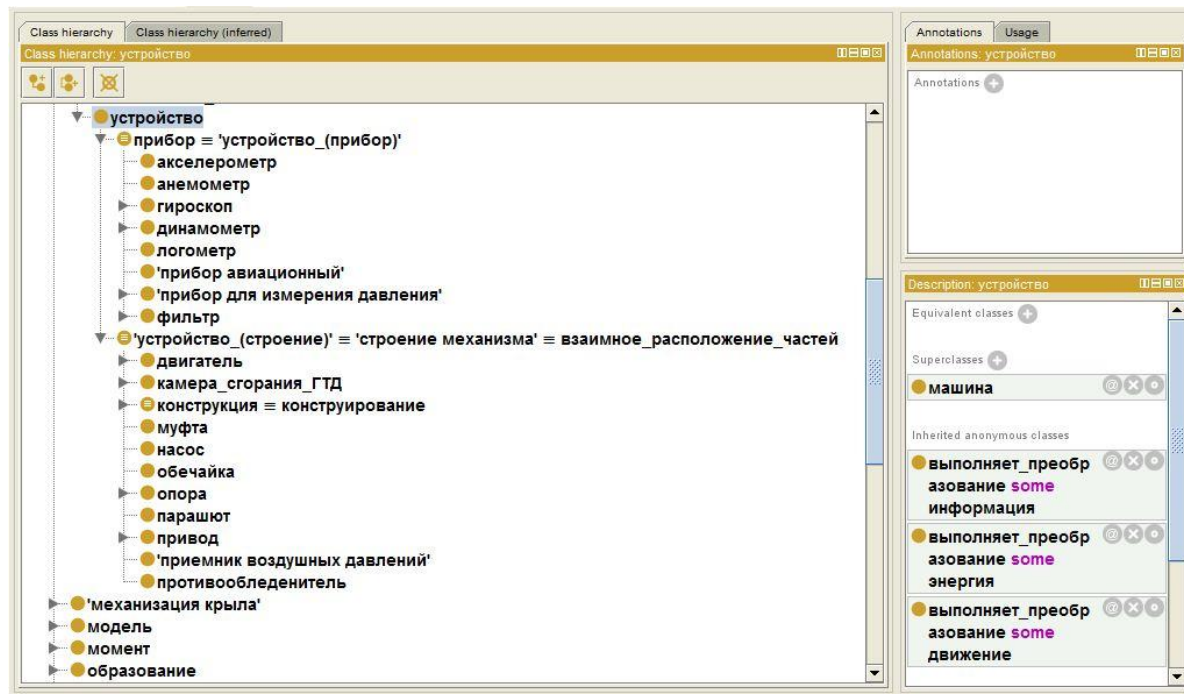


Рисунок 3 – Использование пояснительных слов в классе «Устройство»

сложная ситуация наблюдается в случае добавления многозначных, синонимичных терминов.

В качестве примера рассмотрим взаимосвязь слов «устройство», «строение механизма» и «прибор».

В процессе работы над тезаурусом в качестве главного, но не единственного источника используется словарь авиационных терминов [Боргест и др., 1992], который дает следующее определение прибора:

«ПРИБОР АВИАЦИОННЫЙ – измерительное устройство, входящее в состав авиационной системы индикации экипажу самолета или вертолета и сочетающее в одном корпусе индикационную и приемно-измерительную часть или датчик».

Таким образом, устройство и прибор являются синонимами, что задается в программе с помощью эквивалентных классов. В то же время, под устройством в авиационной отрасли зачастую понимается строение механизма.

форумы [Щербак, 2012], [Semantic, 2012], на которых ведется обсуждения вопросов, касающихся создания онтологий в Protégé. Однако эта тема пока не была затронута, поэтому было предложено следующее решение.

Для облегчения работы с синонимами и решения вопросов, касающихся эквивалентности классов в данном случае, было решено указывать значение, в котором употребляется данный термин, с помощью пояснительных слов. На рисунке 3 приведен пример использования пояснительных слов для класса «Устройство».

Таким образом, в графическом виде класс «Устройство» как «прибор» и как «строение механизма», будет иметь следующий вид (см. рисунок 4).

Подводя итог вышесказанному, можно сделать вывод, что использование пояснительных слов является рациональным способом выхода из ситуации, когда имеются 2 или более синонимичных слова.

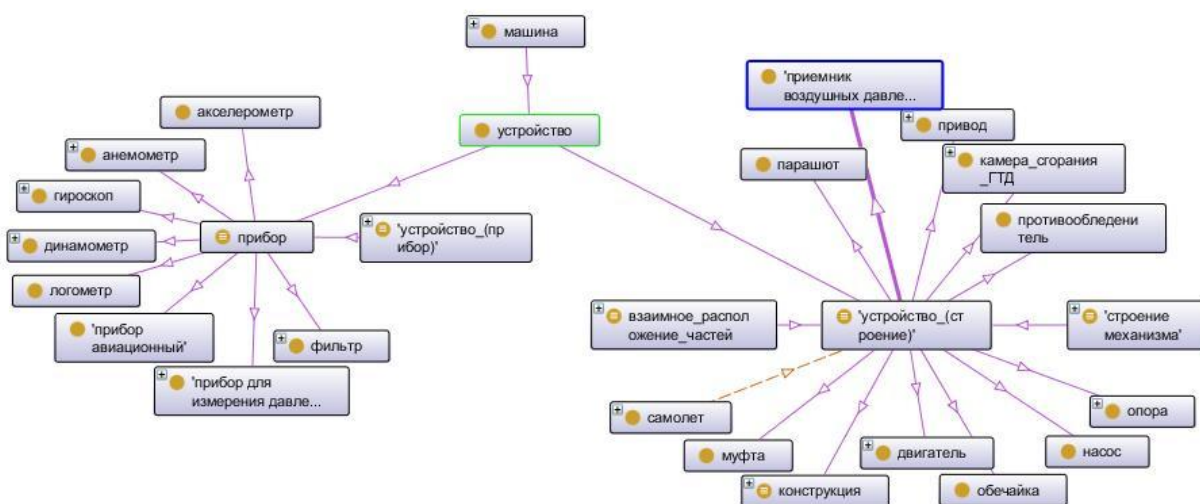


Рисунок 4 – Классификация термина «Устройство»

Одним из наиболее обсуждаемых в последнее время вопросов является объединение онтологий. С этой целью были созданы такие инструментальные средства, как PROMPT, Chimaera, OntoMerge и прочие [Овдей и др., 2004]. В интеллектуальном помощнике проектанта предметной области предварительного проектирования самолета используется онтологический редактор Protégé версии 4.1.beta.

ЗАКЛЮЧЕНИЕ

В данной статье описываются возможные варианты решения синонимических проблем, возникающих в процессе создания тезауруса предварительного проектирования самолета для интеллектуального помощника проектанта.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Боргест и др., 1992] Боргест Н.М., Данилин А.И., Комаров В.А. Краткий словарь авиационных терминов/ Под ред. В.А. Комарова. – М.: Изд-во МАИ, 1992. – 224 с.
- [Боргест и др., 2012] Боргест Н.М., Гиматинова С.Р., Шустова Д.В., Иерархические и ассоциативные связи между терминами в тезаурусе на примере словаря проектанта, Вестник СГАУ, №3, 2012
- [Добров и др., 2009] Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д.. Онтологии и тезаурусы: модели, инструменты, приложения. Изд-во: ИНТУИТру. 2009 – 176 с. <http://www.intuit.ru/department/expert/ontoth/1/>.
- [Овдей и др., 2004] Овдей О.М., Проскудина Г.Ю. Обзор инструментов инженерии онтологий. Электронные библиотеки. РНЭЖ. - 2004 - Том 7 - Выпуск 4.
- [Щербак, 2012] SHCHERBAK.NET - блог PhD Щербака Сергея - о том новом, что будет, когда мы забудем об HTML и вспомним об RDF. <http://shcherbak.net/>
- [Semantic, 2012] Semantic Future by SWUG. Форум Semantic Web User Group. <http://forum.semanticfuture.net/>

PROBLEMS OF SYNONYMS IN THE THESAURUS OF INTELLIGENT ASSISTANT OF DESIGNER

Borgest N.M., Shustova D.V., Odintsova S.A., Knyazihina Y.E.

Samara State Aerospace University named after academician Korolev S.P. (National Research University), Samara, Russia

borgest@yandex.ru
ShustovaDV@yandex.ru
sv-odincova@mail.ru
jul4432@yandex.ru

INTRODUCTION

Thesaurus allows you to structure the knowledge gained in different fields. Special systems, called ontologies, are used for these purposes. Process of visualization of such a system simplifies the understanding of the structure of a considered area.

MAIN PART

Ontological editor Protégé is used to develop a thesaurus. Its capacity is limited, and it can cause some problems. One of these problems is the difficulty of adding the same terms.

CONCLUSION

In work the solution of problem, arising during creation of thesaurus of preliminary aircraft design area are proposed. Some issues to be addressed in the future research are also named. Intermediate results of thesaurus developing are presented.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СИСТЕМА УПРАВЛЕНИЯ ПРОЕКТИРОВАНИЕМ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Жуков И.И. *, Гракова Н.В. *

** Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Ivan_Zhukau@hotmail.com

Natalia_Grakova@hotmail.com

В работе приводится обоснование актуальности создания системы управления проектированием интеллектуальных систем. Рассмотрены основные принципы, на основе которых осуществляется построение системы управления проектированием.

Ключевые слова: интеллектуальная система; база знаний; машина обработки знаний; пользовательский интерфейс; система управления проектированием интеллектуальной системы.

ВВЕДЕНИЕ

При использовании традиционных средств управления процессами проектирования предполагается, что процесс управления и управляемый процесс легко отделимы друг от друга. При проектировании интеллектуальных систем (ИС) данный подход к организации управления нуждается в пересмотре, так как устранение разрыва между системой управления проектированием интеллектуальной системы и самой проектируемой ИС даст большие преимущества. К этим преимуществам, прежде всего, следует отнести более высокий уровень автоматизации проектирования, поскольку предоставляется возможность управлять проектной деятельностью не только субъектов, входящих в команду разработчиков, но и целого ряда программных агентов, которые способны автоматически решать различные проектные задачи.

Если при этом система управления проектированием некоторой интеллектуальной системы сама будет построена как ИС (с соответствующей базой знаний, машиной обработки знаний и пользовательским интерфейсом) по той же самой технологии, что и проектируемая ИС, то их интеграция будет осуществляться более технологичным способом. К тому же повышение уровня интеллекта системы управления проектированием может дать неплохие результаты.

Процесс управления проектированием не должен прекращаться и после ввода в эксплуатацию

ИС. В то же время, срок использования такой системы должен только увеличиваться. Появляется проблема сопровождения и увеличения сроков использования внедренной ИС. В связи с этим, сопровождение и дальнейшее развитие ИС требует значительных трудовых затрат, а также предъявляет высокие требования к разработчикам и лицам сопровождающим данные ИС. Для решения данной проблемы при разработке ИС используется несколько подходов:

1. сопровождение ИС происходит только при разработке новой версии. Активное сопровождение прекращается после момента передачи разработанной версии конечному пользователю;

2. использование технологий облачных вычислений для разработки ИС [Грибова и др., OSTIS 2011]. При данном подходе сопровождение и замена версий продукта происходит незаметно для конечных пользователей.

Стоит учитывать тот факт, что большинство проектов в настоящее время, ориентируются на проекты с открытым исходным кодом (Open Source проекты), в том числе и ИС. В связи с этим, и разрабатываемая система управления проектированием интеллектуальных систем (СУП ИС) должна учитывать специфику проектов с исходным кодом.

В тоже время, любой класс ИС требует соблюдения определенных методик проектирования ИС, которые в большинстве случаев не учитываются существующими средствами управления процессами проектирования и разработки программных систем.

Целью создания предлагаемой СУП ИС является необходимость в использовании всех выше приведённые принципы для получения более качественного информационного продукта.

В частности, в данной работе будет рассматриваться СУП ИС, которые используют технологию компонентного проектирования OSTIS (Open Semantic Technology for Intelligent Systems).

1. Основные положения разработки интеллектуальных систем, предназначенных для управления проектированием интеллектуальных систем

Для того чтобы обеспечить качественное управление проектированием ИС необходимо произвести интеллектуализацию процесса управления. Интеллектуализацию процесса управления проектированием можно обеспечить за счет внедрения в сам процесс управления интеллектуальной системой СУП. Данную СУП можно приравнять к высокоуровневому механизму, о котором говорится в работе [Norvig P., Cohn D.]. Она должна сопровождать ИС на пути всего жизненного цикла: как в момент разработки, так и в момент активного использования конечным пользователем.

Через базу знаний (БЗ) интеллектуальной СУП осуществляется взаимодействие между разработчиками ИС, которые в некотором смысле представляются в виде агентов решающих некоторые проектные задачи. В аналогичной взаимосвязи находятся разработчик и разрабатываемая ИС. Сам процесс управления проектированием превращается в постоянное изменение состояния БЗ интеллектуальной СУП.

Но одного такого взаимодействия между самой ИС и интеллектуальной СУП недостаточно для устранения разрыва между процессом управления проектированием ИС и самой ИС. Здесь необходима полная интеграция двух систем: интеллектуальной СУП и самой ИС. Данный подход основан на методике компонентного проектирования ИС [Голенков В.В., Гулякина Н.А., 2011]. При интеграции интеллектуальной СУП и самой ИС будет учитываться следующее:

- взаимодействие и функционирование систем должно происходить в общем оперативном пространстве;
- системы должны строиться по одним и тем же принципам и с использованием одних и тех же технологий.

Соблюдение данных условий при интеграции ИС выведет каждую из них на новый качественный уровень. А именно:

- в процессе интеграции ИС приобретает дополнительные инструменты для своего развития и самосовершенствования от СУП. К ним можно

отнести, например, методику проектирования ИС [Давыденко И.Т., 2011], а также дополнительные методы планирования задач, распределения ресурсов проекта, разбиении проектов ИС на подпроекты, при создании задач для разработчиков и во многих других случаях и многое другое;

- упростится разработка самих ИС. Упрощение будет достигнуто за счет того, что многие компоненты уже будут присутствовать в интегрируемой интеллектуальной СУП. Например, при разработке ИС отпадет необходимость в создании подсистемы для генерации заданий.

Самое важное качество, которое получает интеллектуальная СУП при интеграции с управляемой ИС, это то, что она становится субъектом по отношению к управляемой ИС. Интеллектуальную СУП можно считать субъектом потому, что, как и разработчик ИС или программный агент, интеллектуальная СУП сможет воздействовать на внутреннее состояние управляемой ИС, при этом будут использоваться одни и те же механизмы воздействия и взаимодействия.

2. Структура системы управления проектированием интеллектуальных систем

Предлагаемая СУП интеллектуальных систем основывается на всех тех принципах интеллектуальной СУП, которые необходимо реализовать для того, чтобы получить систему, предназначенную для управления проектированием ИС построенных по технологии OSTIS. Описание модели включает описание трех основных компонентов интеллектуальных СУП:

- описание БЗ ИС;
- описание машины обработки знаний (МОЗ) ИС;
- описание пользовательского интерфейса ИС.

Перед тем, как перейти к описанию каждого из компонентов предлагаемой интеллектуальной СУП, следует отметить подсистемы традиционных СУП, которые оказывают непосредственное влияние на проектирование подсистем интеллектуальной СУП:

- подсистема, отвечающая за совместное взаимодействие участников проектов;
- подсистема, отвечающая за управление распределения ресурсов проекта;
- подсистема, отвечающая за управление документами проекта (система, используемая для отслеживания и хранения документов);
- подсистема, отвечающая за управление информации о проекте (устав проекта, описание проекта, цели и т.п.);
- подсистема, отвечающая за управление задачами проекта;
- подсистема, отвечающая за управление фазами жизненного цикла проекта;

– подсистема, позволяющая строить все возможные отчеты и производить анализы эффективности использования ресурсов и состояния поставленных задач;

– подсистема по защите информации всего проекта в целом.

К реализации данного набора подсистем стремятся практически все СУП.

2.1. Описание базы знаний интеллектуальной системы

Предлагаемая СУП включает в себя следующие онтологии:

– онтология области управления проектированием;

– онтология области управления ресурсами проектов;

– онтология области защиты информации.

Онтология области управления проектированием содержит формальное описание таких основных понятий, как проект, подпроект, задание, состояния, в которых могут находиться проект и/или задание, а также правила перехода из одного состояния в другое, исполнитель задания и т.д.

Онтология области управления ресурсами проектов содержит формальное описание понятия ресурса проекта, участника проекта, финансовых и технических средств проекта и многое другое.

Онтология области защиты информации включает в себя такие формальные описания понятий как право доступа участника, право доступа к узлам БЗ, правило аутентификации участников в системе, некоторого механизма разграничения прав доступа и т.д.

2.2. Описание машины обработки знаний интеллектуальной системы

МОЗ интеллектуальной СУП декомпозируется на следующие компоненты:

– компонент, отвечающий за управление информацией о проекте;

– компонент, отвечающий за управление распределением ресурсов проекта;

– компонент, отвечающий за управление заданиями;

– компонент, отвечающий за взаимодействие между разработчиками системы;

– компонент, отвечающий за защиту информации.

Хочется отметить тот факт, что данная декомпозиция БЗ и МОЗ не является конечной и в будущем будет расширяться.

2.3. Описание пользовательского интерфейса интеллектуальной системы

Пользовательский интерфейс интеллектуальной

СУП строиться по тем же принципам, что и интерфейс автоматизированной метасистемы OSTIS (AMS OSTIS) [OSTIS, 2012]. Поэтому в данной работе более детальное рассмотрение интерфейса производиться не будет. Но следует отметить, что основной особенностью интерфейса для интеллектуальной СУП, является то, что в нем присутствуют специализированные компоненты пользовательского интерфейса: компонент для отображения диаграммы Ганта или компонент для отображения некоторого отчета и многое другое.

2.4. Возможные классы пользователей интеллектуальной системы управления проектированием

В предлагаемой системе также выделяется несколько классов возможных пользователей системы. Каждому классу пользователей соответствует определенный набор прав, разрешающих выполнять определенные операции в интеллектуальной СУП и в проектируемой ИС.

К первой группе пользователей относят обычные гости/читатели, как проектируемой ИС, так и интеллектуальной СУП. Данным пользователям разрешен доступ к общественной проектной информации в режиме чтения без возможности ее изменения.

Ко второй группе пользователей относятся разработчики ИС и интеллектуальной СУП. У данной группы пользователей есть возможность вносить изменение в БЗ проектируемой ИС, настраивать интеллектуальную СУП, генерировать новые элементы системы.

К третьей группе пользователей относятся разработчики технологий используемых для построения ИС, а также разработчики самой интеллектуальной СУП. Данная группа пользователей обладает наибольшим набором прав в интеллектуальной СУП.

Каждая из выделенных групп пользователей может делиться в свою очередь на подгруппы, каждая из которых отличается набором обладаемых прав в ИС. Например, выделение из всех разработчиков некоторого проекта руководителя проекта, ответственных за выполнения конкретных заданий: разработка или тестирование фрагмента БЗ и т.д. Особенности данной декомпозиции привязываются к конкретному проекту разработки ИС.

После внедрения полученной СУП, мы получаем множество пар вида: < проектируемая ИС, интеллектуальная СУП >. В данном множестве пар можно заметить такую особенность, что не все компоненты интеллектуальной СУП являются универсальными, то есть некоторые из них без дополнительной доработки или настройки неудобно использовать для нескольких ИС одновременно. Для решения данной проблемы был предложен следующий

подход. Во-первых, выделить некоторое ядро интеллектуальной СУП, общее для всех интеллектуальных СУП и которое позволяет управлять проектами/подпроектами ИС, используя наиболее общие аспекты управления проектированием ИС. Во-вторых, разработать специализированные средства, позволяющие автоматизировать процесс настройки ядра интеллектуальной СУП, под конкретный проект ИС. В-третьих, разработка некоторого дополнительного расширения для ядра интеллектуальной СУП, позволяющего выполнить необходимые недостающие функции.

При реализации предлагаемой интеллектуальной СУП для представления знаний используются однородные семантические сети с базовой теоретико-множественной интерпретацией. Для кодирования таких сетей выбран язык SC (Semantic Code). Пользовательский интерфейс системы является web-ориентированным.

Заключение

В данной работе были предложены основные принципы, которые могут быть использованы при решении ряда проблем возникающих при проектировании и в процессе эксплуатации ИС. Описание путей решения данных проблем происходило в рамках класса ИС, использующих технологию компонентного проектирования OSTIS.

В работе также приводилось краткое описание некоторых составляющих компонентов СУП.

Библиографический список

[Грибова В.В., Клещев А.С. и др.] OSTIS-2011 – Облачная платформа для разработки управления интеллектуальными системами.

[Грибова В.В., Клещев А.С.] Процессы управления интеллектуальными системами / Грибова В.В. Клещев А.С.// Онтология проектирования. – 2011 – № 1 с 22-32.

[Norvig P., Cohn D.] Adaptive software. [Электронный ресурс]. – Режим доступа: <http://norvig.com/adaper-pcai.html>

[Давыденко И.Т.] Комплексная методика проектирования семантических моделей интеллектуальных справочных систем/ И.Т. Давыденко// Материалы международной научно-технической конференции OSTIS-2012. – Минск БГУИР – с 457

[Голенков В.В., Гулякина Н.А.] Графодинамические модели параллельной обработки знаний: принцип построения, реализации и проектирования / Голенков В.В., Гулякина Н.А. //Материалы международной научно-технической конференции OSTIS-2012. – с 23-52.

[Birgit Dippelreiter], Semantic based Project Management part 7, - International Conference on Electronic Commerce, Innsbruck, Austria, August 19, 2008. – [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://ceur-ws.org/Vol-404/>

[Lijun Shen, David K.H. Chua], A Framework for Collaborative Project Planning Using SemanticWeb Technology / Lijun Shen, David K.H. Chua // Proceedings of the 2010 International Conference on Engineering, Project, and Production Management – C. 41.

[OSTIS, 2012] Проект OSTIS – [Электронный ресурс]. – 2011 – Режим доступа: <http://www.ostis.net>

[Соснин П.И., Маклаев В.А.] Инструментальные средства для спецификации концептуализаций в проектировании автоматизированных систем/ Соснин П.И., Маклаев В.А.// Онтология проектирования. – 2012 – №1 с 39-52.

SYSTEM OF PROJECT MANAGEMENT FOR INTELLIGENT SYSTEMS

Zhukau I.I., Grakova N.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Ivan_Zhukau@hotmail.com

Natalia_Grakova@hotmail.com

The paper presents a rationale for the relevance of the creation of project management systems of intelligent systems. The basic principles which are used for construction of the project management system are discussed.

INTRODUCTION

With traditional tools for controls the processes of design assumes that the management and controlled process easily separated from each other. This idea has to be revised in scope of intelligent systems. Bridging the gap between the project management system (PMS) of intelligent system and intelligent system gives great advantages. Here some of them:

- the project management process will become more automated because new intellectual PMS will contain agents which can resolve project tasks;
- if PMS will be built on the same technologies like an itelligent system than the systems integration will be more flexible.

New PMS will can be used with any intelligent system which use the OSTIS technology for system creation.

MAIN PART

This section of the article contains description of five main principles which are used for PMS creation.They are:

- the intellectualization of project management process with help of intelligent PMS;
- integration between intelligent system with project management system;
- project management of open source project methodology is used for management of project of intelligent system;
- the component technology have to be used for integration of intelligent system and PMS.

The architecture of intelligent project management system is described also in this section.

CONCLUSION

The main principals are used for solving some problems available during design of intelligent system are described in this paper. All problems reviewed in scope of intelligent systems built on OSTIS technology.



УДК 004.7

ЖИВУЧЕСТЬ ИНФОРМАЦИОННЫХ ОБЪЕКТОВ В СЕТИ ИНТЕРНЕТ

Додонов А.Г.*, Ландэ Д.В.*

** Институт проблем регистрации информации НАН Украины
г. Киев, Украина*

dodonov@ipri.kiev.ua

dwlande@gmail.com

В работе рассматривается понятие живучести информационных объектов в сети Интернет, приводятся механизмы обеспечения живучести информационных объектов, дается формальное определение и формулы для расчета живучести информационных объектов в случае степенного распределения потерь информации на серверах.

Ключевые слова: информационный объект; Интернет; живучесть; степенное распределение

ВВЕДЕНИЕ

В связи с развитием сети Интернет в последнее время особое место среди задач, получивших актуальность, занимают задачи, связанные с обеспечением живучести информационных объектов и систем, которые связываются с моделированием их жизненного цикла: формирования и развития, реакции на деструктивные воздействия, восстановления, разрушения.

Под живучестью понимают способность системы (или ее части, объекта) адаптироваться к новым непредусмотренным условиям функционирования, противостояния нежелательным влияниям при одновременной реализации основной функции.

Существует несколько механизмов, обеспечивающих живучесть информационных объектов в Интернете.

В статье рассматриваются некоторые наиболее распространенные механизмы обеспечения живучести, которые в реальности применяются не в чистом виде, а как правило, комбинируются.

Для изучения проблем, связанных с живучестью необходимо четко определить как само это понятие, так и привести формальную модель, на основании которой можно рассчитывать уровень живучести для таких трудно формализуемых сущностей, как информационные объекты.

Кроме того, с живучестью информационных объектов сегодня связывают такие социально важные проблемы, как обеспечение

информационной безопасности, приватности в сети. Этим вопросам посвящена заключительная часть данной статьи.

1. Механизмы обеспечения живучести информационных объектов

Понятие живучести информационной составляющей сети Интернет подразумевает способность информационных объектов (новостных сообщений, статей, документов, видеороликов и т.д.) своевременно выполнять свои функции (информирования) в условиях действия дестабилизирующих факторов. Такими факторами могут быть устранение отдельных объектов из информационного пространства, потеря ими свойств актуальности, доступности [Додонов, 2011], [Knight, 2003] рассмотрим некоторые из них.

1. Копирование данных при размещении их на целевой ресурс. То есть автор размещает информацию, которая копируется хостинг-провайдером на некоторое количество зеркальных серверов. Пример – скандально известная служба WikiLeaks (несколько сотен серверов, на которых хранятся фрагменты копий).

2. Перепечатка информации (републикации, «копипаст») на другие сайты с целью их информационного наполнения. В качестве примера приводится соотношение оригинальной информации и общего объема информации, сканируемой системой InfoStream [Григорьев, 2007] за первые четыре месяца 2012 г. по дням (рисунок 1). При этом следует отметить, что наиболее важная и интересная информация перепечатывается сотни раз, в то время как неактуальная, неинтересная информация практически не дублируется.

3. Размещенная однажды информация навсегда попадает в архивные службы Интернета типа Архив Интернета (<http://archive.org>), который накапливает сетевую информацию. Библиотека Конгресса США (www.loc.gov) купила права на хранение всех публичных сообщений социальной сети Twitter с 2006 года и всех твитов, которые будут опубликованы впредь. Библиотека Конгресса также реализует и национальный проект сохранения и распространения цифрового контента Digital Preservation (www.digitalpreservation.gov – 1400 коллекций данных).

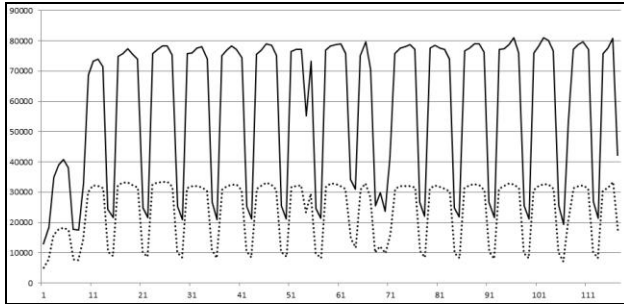


Рисунок 1 – Соотношение оригинальной информации (пунктирная линия) и общего объема информации (сплошная линия), сканируемой системой InfoStream

4. Информация часто остается в кешах поисковых систем, даже если она удалена с веб-страницы или страницы социальной сети. Информация индексируется глобальными информационно-поисковыми системами и остается в их кеш-памяти, откуда она доступна пользователям. Лишь относительно недавно у администраторов веб-ресурсов появилась возможность самостоятельного удаления своего контента из кешей Google и Яндекс. Часто многое, например, о человеке можно узнать в его блоге, онлайн-репутация – сегодня модный бренд. Что касается социальной сети Twitter, то twitFlink (<http://www.twitflink.com>), к примеру, который быстро соберет и выдаст твиты пациента. Сервис Google Replay позволяет находить и просматривать тематические сообщения в микроблогах за указанный период времени.

5. И наконец, информация с веб-сайта может сохраняться на локальных компьютерах конечных пользователей, которые получили к ней доступ либо непосредственно, либо через интеграторов информации.

2. Формальные модели живучести информационных объектов

Организационный комитет считает своим долгом указать на ряд часто встречающихся ошибок при оформлении статей:

Известно, что живучесть информационного объекта оценивается как вероятность того, что объект будет неповрежденным в течение определенного периода времени t при определенных условиях [Li, 2012].

Если информационный объект сохраняется на n серверах (носителях информации), то вероятность разрушения этого объекта оценивается как:

$$F_{lost}(t) = \prod_{i=1}^n F_i(t).$$

В этом произведении $F_i(t)$ – вероятность потери информационного объекта на i -м сервере за время t .

Соответственно живучесть оценивается как:

$$S_n(t) = 1 - F_{lost}(t) = 1 - \prod_{i=1}^n F_i(t).$$

Допуская, что вероятность уничтожения информационных объектов пропорциональна времени их существования, и то, что время их разрушения имеет степенное распределение (в соответствии с законом Парето), можно считать целесообразным и обоснованным исследование модели со степенным распределением потерь информационных объектов, что принципиально отличается от подходов, в которых используется пуассоновский поток ошибок (теория систем массового обслуживания) и распределение ошибок по Вейбулу. В этом случае, живучесть можно оценивать как:

$$S_n(t) = 1 - \prod_{i=1}^n F_i(t) = 1 - \prod_{i=1}^n C t^{-\beta} = 1 - C^n t^{-n\beta},$$

где C , β – некоторые константы.

Закономерности статистического распределения времени жизни информационных объектов позволяют делать выводы, связанные с их живучестью, а именно учитывать явления самого подобия, нерегулярности потери информации, наличие «тяжелого хвоста» в распределении, которое характеризует чрезвычайно большое количество фактически устаревших информационных объектов и т.п.

При анализе жизненного цикла информационного объекта можно использовать еще два больших класса моделей: булевы и марковские.

В булевой модели можно предположить, что копии информационного объекта содержатся на n серверах, при этом i -му серверу соответствует булева переменная x_i , которая может принимать значения $\{0,1\}$, то есть $x_i = 1$, если информационный объект на сервере i активен, и 0 – в противном случае. Состояние информационного объекта определяется структурной функцией его доступности (булевой функцией) $S(x_1, x_2, \dots, x_n)$, которая принимает значения 1, если информационный объект доступен, и 0 в противном случае.

Если доступность информационного объекта рассматривать как функцию времени, то состояние информационного объекта на i -м сервере можно рассматривать как случайный процесс $x_i(t)$, принимающий в произвольные моменты времени $t \geq 0$ значения 0 и 1. Для системы определяется вероятность ее безотказной работы по приведенным выше.

Среди недостатков булевых моделей можно назвать предположение только о двух состояниях информационного объекта – активности (доступности) и неактивности. Кроме того, в общем случае характер отказов отдельных копий информационного объекта зависит от состояния других копий.

Информационный объект можно описать также марковской моделью. Пусть система (множество копий информационного объекта) имеет m возможных состояний. Обозначим множество состояний через $M = \{z_1, z_2, \dots, z_m\}$. Для любого фиксированного момента времени $t \geq 0$ состояние системы $z(t)$ интерпретируется как случайная величина. Заданы множество всех состояний M , вектор распределения начальных вероятностей $p(0)$ и функция переходных вероятностей. Определяется вероятность активности, «жизни» системы в заданный момент времени t (готовность системы). Применимость марковских моделей также имеет свои границы. Интенсивности переходов между отдельными состояниями системы могут быть нестационарными, принимаемые при расчете допущения относительно распределения интенсивности отказов могут значительно снизить точность полученных результатов; число состояний системы может быть так велико, что расчет становится практически невозможным.

Оценка живучести информационных объектов может проводиться на всех этапах их жизненного цикла. Существует несколько подходов, к проведению оценки живучести, имеющих наиболее общий характер. Живучесть можно оценить относительно некоторого стандартного внешнего воздействия либо относительно множества внешних воздействий. В этом случае решается задача нахождения множества характеристических векторов состояний информационного объекта (в простейшем случае – распределение по серверам), в которых реализуется конфигурация, обеспечивающая выполнение цели функционирования. Мощностью этого множества может служить мерой живучести всего информационного объекта.

При анализе живучести информационных объектов рассматривается проблема информирования по их различным аспектам, независимо от наличия или отсутствия неблагоприятных факторов. В связи с этим, в качестве количественного критерия оценки живучести целесообразно использовать отношение

количества функций, выполняемых объектом при наличии определенных неблагоприятных воздействий либо множества таких воздействий, к общему количеству функций информационного объекта, с учетом критичности выполняемых и не выполняемых функций. Критичность каждой конкретной функции определяется индивидуально для каждого конкретного информационного объекта исходя из его специфики. Количественный показатель живучести конкретного информационного объекта в заданных условиях можно вычислять по формуле:

$$S = \sum_{i \in \Delta} \alpha_i / \sum_{j \in \Theta} \alpha_j,$$

где Θ – множество всех функций информирования, Δ – множество функций информационного объекта, выполняемых в заданных условиях ($\Delta \subseteq \Theta$), α_n – критичность n -й функции. Таким образом, количественная оценка живучести информационного объекта будет изменяться в интервале $[0, 1]$, живучесть тем выше, чем больше ее количественная оценка.

3. Цифровые следы и тени

Удаление информационного объекта с веб-ресурса не может гарантировать его исчезновения из Интернета. Остаются не только «цифровые следы» и «цифровые тени».

Выражение «цифровые следы» («digital footprint») относятся к той информации, которая оставляется самим пользователем при работе в Сети и по которой можно не только его идентифицировать, но и «привязать» к определенным действиям, событиям, восстановить какие-то фрагменты биографии.

Часто пользователи по доброй воле указывают свои ФИО, «привязывая» дальнейшую информацию к собственной личности, дату рождения, семейное положение, образование, профессию, места предыдущей работы и много чего еще, включая и контактные телефоны, и адреса электронной почты. Кроме «цифровых следов», которые мы оставляем сами, информация о пользователях постоянно тиражируется и без всякого его участия.

Информация о пользователе, создаваемая без его участия, получила название «цифровой тени» («digital shadow»), которые возникают и накапливаются всякий раз, когда кто-то ищет пользователя через поисковые системы, когда происходит электронная почтовая рассылка по спискам, в которых он фигурирует и во многих других случаях. Индексация роботами поисковых машин страниц с информацией пользователя и их последующее кэширование – это тоже создание «цифровой тени», доступной каждому. Кроме «цифровых теней открытого доступа», создаются и копятся «цифровые тени ограниченного доступа» – записи камер наблюдения, банковские транзакции,

биллинги Интернет-магазинов, сервисов продажи билетов, телефонных звонков и др.

По оценке International Data Corporation (IDC), аналитической компании, которая специализируется на исследованиях IT-рынка, объем «цифровой тени», т.е., информации о пользователе Интернет, которая создается без его участия, уже в 2007 г. превысил объем информации, которую создает сам пользователь.

С проблемой репутации в Интернете ежедневно сталкивается все больше пользователей. Об этом свидетельствует и появление особых сайтов (например, www.suicidemachine.org), позволяющих разом удалить регистрацию и все сделанные записи на различных форумах и в социальных сетях. Такая операция называется «покончить с собой в Интернете». Однако эта система пока несовершенна. С недавнего времени эту заботу берутся специальные компании, так называемые «Интернет-чистильщики».

ЗАКЛЮЧЕНИЕ

Понятие живучести информационного объекта подразумевает его способность своевременно выполнять свои функции (в данном случае – информирования) в условиях действия дестабилизирующих факторов. Такими факторами могут быть устранение отдельных информационных объектов из информационного пространства, потеря их актуальности, доступности. Необходимо отметить, что привлечение внимания аудитории к другой теме, порождение другого информационного объекта также может снизить актуальность текущего информационного объекта.

При этом следует учитывать, что самая важная информация, попав в Интернет, остается там практически навсегда, и как показывает практика, рассчитывать на ее легкое удаление или изменение не приходится. Лучшим методом оказывается вытеснение нежелательной информации новыми сюжетами, проведение специальных мероприятий по содержательному исправлению ошибок [Додонов, 2010].

Живучесть информационных объектов и систем трудно заметить в нормальных условиях функционирования. Это свойство рельефно проявляется только в случаях потери информации, возникновения нарушений в структуре информационной системы, отказе ее составляющих, отдельных функций, целенаправленных деструктивных влияний. В зависимости от класса систем, их сложности, степени организованности, а также от выбранного уровня анализа свойство живучести может оцениваться как устойчивость, надежность, адаптивность, отказоустойчивость.

Наблюдаемый в настоящее время процесс в области интеллектуализации автоматизированных систем, перехода от простой обработки данных к процессам поддержки принятия решений требует новых подходов. Исначальная парадигма

информационных систем, сформированная несколько десятилетий тому назад, уже не соответствует реальной ситуации – объемам и динамике информационных потоков, сетевой топологии. Необходим поиск новых принципов, в рамках которых оказалось бы возможным проектирование качественно новых систем обработки больших и динамичных массивов информации – информационных систем. Именно поэтому особое место среди задач, занимают задачи, связанные с обеспечением живучести информационных систем и их компонент.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Додонов, 2011] Додонов А.Г., Ландэ Д.В. Живучесть информационных систем. – К.: Наук. думка, 2011. – 256 с.
- [Knight, 2003] Knight J.C., Strunk E.A., Sullivan K.J. Towards a Rigorous Definition of Information System Survivability // Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03), 2003.
- [Григорьев, 2007] Григорьев А.Н., Ландэ Д.В., Бороденков С.А., Мазуркевич Р.В., Пацьора В.Н. InfoStream. Мониторинг новостей из Интернет: технология, система, сервис: научно-методическое пособие. – Киев: ООО "Старт-98", 2007. – 40 с.
- [Li, 2012] Li Y., Miller E.L., Long D.D.E. Understanding Data Survivability in Archival Storage Systems // Proceedings of the 5th Annual International Systems and Storage Conference (SYSTOR 2012), June 4–6, 2012, Haifa, Israel.
- [Додонов, 2010] Додонов А.Г., Ландэ Д.В. Живучесть информационных сюжетов // Материалы XI Международной научно-практической конференции "Информационная безопасность". Ч. 2. – Таганрог: Изд-во ТТИ ЮФИ, 2010. – С. 179-183.

SURVIVABILITY OF INFORMATION OBJECTS IN THE INTERNET

Dodonov A.G. *, Lande D.V. *

**The Institute for Information Recording
NAS Ukraine, Kiev, Ukraine*

dodonov@ipri.kiev.ua

dwlande@gmail.com

In work the concept of survivability of information objects in the Internet is considered. Mechanisms of ensuring survivability of information objects are given. Formal definition and formulas for calculation of survivability of information objects in case of power law distribution of losses of information on servers is given.

Keywords: information object; Internet; survivability; power law distribution



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.855

ТЕХНОЛОГИЯ КОМПОНЕНТНОГО (МОДУЛЬНОГО) ПРОЕКТИРОВАНИЯ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ СИСТЕМАТИЗАЦИИ И КОНТРОЛЯ ЗНАНИЙ

Касторнов А.Ф. *, Касторнова В.А. **

* *Череповецкий государственный университет,
г. Череповец, Россия*
a_kastornov@mail.ru

** *Институт информатизации образования
Российской академии образования, г. Москва, Россия*
kastornova_vasya@mail.ru

В данной работе рассмотрены вопросы построения обучающей экспертной системы (ОЭС), призванной служить одним из педагогических инструментов систематизации и контроля знаний студентов. Предлагаемая авторами структура ОЭС основывается на модульном принципе ее построения и включает в себя модули, позволяющие создавать базу знаний в некоторой предметной области, наполнять ее содержанием, производить ее настройку (обучение), а затем использовать при проведении занятий.

Ключевые слова: обучающая экспертная система; модуль; база знаний; обучение экспертной системы.

ВВЕДЕНИЕ

Целью данной работы является практическая реализация технологии модульного проектирования экспертной системы. В работе описана Паскаль программа, создающая базу знаний, которая включает в себя модули, позволяющие компоновать различные уровни (узлы), содержащие перечень вопросов (переменных), которые ставит ЭС пользователю, чтобы правильно идентифицировать тот или иной объект (исход) в некоторой предметной области.

Экспертные системы (ЭС) возникли как значительный практический результат в применении и развитии методов искусственного интеллекта (ИИ)- совокупности научных дисциплин, изучающих методы решения задач интеллектуального (творческого) характера с использованием ЭВМ. ЭС - это набор программ, выполняющий функции эксперта при решении задач из некоторой предметной области. ЭС выдают советы, проводят анализ, дают консультации, ставят диагноз. Практическое применение ЭС на предприятиях способствует эффективности работы и повышению квалификации специалистов.

Главным достоинством экспертных систем является возможность накопления знаний и

сохранение их длительное время. В отличие от человека к любой информации экспертные системы подходят объективно, что улучшает качество проводимой экспертизы. При решении задач, требующих обработки большого объема знаний, возможность возникновения ошибки при переборе очень мала.

Экспертная система состоит из базы знаний (части системы, в которой содержатся факты), подсистемы вывода (множества правил, по которым осуществляется решение задачи), подсистемы объяснения, подсистемы приобретения знаний и диалогового процессора. При построении подсистем вывода используются методы решения задач искусственного интеллекта.

Основными отличиями ЭС от других программных продуктов являются использование не только данных, но и знаний, а также специального механизма вывода решений и новых знаний на основе имеющихся. Знания в ЭС представляются в такой форме, которая может быть легко обработана на ЭВМ. В ЭС известен алгоритм обработки знаний, а не алгоритм решения задачи. Поэтому применение алгоритма обработки знаний может привести к получению такого результата при решении конкретной задачи, который не был предусмотрен. Более того, алгоритм обработки знаний заранее неизвестен и строится по ходу

решения задачи на основании эвристических правил. Решение задачи в ЭС сопровождается понятными пользователю объяснениями, качество получаемых решений обычно не хуже, а иногда и лучше достигаемого специалистами. В системах, основанных на знаниях, правила (или эвристики), по которым решаются проблемы в конкретной предметной области, хранятся в базе знаний. Проблемы ставятся перед системой в виде совокупности фактов, описывающих некоторую ситуацию, и система с помощью базы знаний пытается вывести заключение из этих фактов.

База знаний - наиболее важная компонента экспертной системы, на которой основаны ее «интеллектуальные способности». В отличие от всех остальных компонент ЭС, база знаний - «переменная» часть системы, которая может пополняться и модифицироваться инженерами знаний и опытом использования ЭС, между консультациями (а в некоторых системах и в процессе консультации). Существует несколько способов представления знаний в ЭС, однако общим для всех них является то, что знания представлены в символической форме (элементарными компонентами представления знаний являются тексты, списки и другие символические структуры). Тем самым, в ЭС реализуется принцип символической природы рассуждений, который заключается в том, что процесс рассуждения представляется как последовательность символических преобразований.

Наиболее распространенный способ представления знаний - в виде конкретных фактов и правил, по которым из имеющихся фактов могут быть выведены новые. Правила в базе знаний служат для представления эвристических знаний (эвристик), т.е. неформальных правил рассуждения, вырабатываемых экспертом на основе опыта его деятельности.

Системы, основанные на знаниях, могут входить составной частью в компьютерные системы обучения. Система получает информацию о деятельности некоторого объекта (например, студента) и анализирует его поведение. База знаний изменяется в соответствии с поведением объекта. Примером этого обучения может служить компьютерная игра, сложность которой увеличивается по мере возрастания степени квалификации играющего.

В последние годы в учебном процессе все большее распространение находят компьютерные технологии проверки знаний, которые принимают, в основном, форму компьютерного тестирования. Данная технология имеет свои достоинства и недостатки, главным из которых является возможность "угадывания" правильного ответа при незнании самого материала. На наш взгляд, сама тестовая форма проверки знаний далека от совершенства и не может качественно оценить знания, особенно при организации итогового контроля знаний по завершению изучения какой-либо крупной темы учебного курса. С помощью

тестирования часто бывает трудно оценить знания учащихся с точки зрения выявления понимания ими и качественных характеристик, и взаимосвязей изучаемых в курсе базовых понятий.

В качестве примера можно привести изучение темы "Программное обеспечение ЭВМ" в курсе информатики. Известно, что здесь принято проводить классификацию ПО на три группы: базовое ПО, системы программирования и прикладное ПО. Аналогичная ситуация имеет место и при классификации типов данных в языке Паскаль. Здесь важно уметь определять классификацию типов по таким характеристикам как: простые, специальные и структурные типы, а в рамках этих классов отличать встроенные, пользовательские, статические и динамические типы данных. С помощью теста проверить эти знания довольно проблематично, так как по своей сути в вопросе теста нужно указать несколько вариантов ответов, один из которых правильный, а все остальные либо заведомо неправильные, либо правдоподобные. Подбор этих вариантов не всегда прост и особенно важно, не всегда удачен со стороны составителя теста. На наш взгляд, эта проблема легко решается, если в качестве инструмента проверки и систематизации знаний использовать экспертную систему учебного назначения.

Экспертные системы создаются для решения разного рода проблем. Основные типы их деятельности можно сгруппировать в следующие категории: интерпретация, прогноз, диагностика, наблюдение, проектирование, отладка, обучение, управление. Экспертные системы, выполняющие обучение, подвергают диагностике, "отладке" и исправлению (коррекции) поведение обучаемого. Обучающие системы создают модель того, что обучающийся знает и как он эти знания применяет к решению проблемы [Касторнова, 2011].

Разработанная нами экспертная система учебного назначения состоит из следующих модулей: *инициализация системы, ввод примеров, тренировка, обучение, запоминание базы знаний на внешнем носителе* (в виде набора массивов фактов и правил), *удаление и добавление переменных и исходов, загрузка базы знаний из ВЗУ в ОЗУ*, а также основной модуль *диалога с пользователем* [Касторнов, 2002]. Первые модули заполняются экспертом (преподавателем), а с последними двумя работают учащиеся. При инициализации системы эксперт определяет структуру базы знаний, где указываются число узлов, переменных и исходов, которые затем вводятся в эту базу. При этом набор переменных для каждого узла задает характерные признаки (атрибуты), присущие объектам (исходам) этого узла. Например, в базе знаний "Программное обеспечение ЭВМ" среди переменных первого узла указываются такие признаки: *создает файлы, управляет работой ЭВМ, создает тексты, решает математические задачи* и пр., которые определяют указанную выше классификацию ПО ЭВМ. Во

втором узле в число переменных входят исходы первого узла и дополнительные признаки, которые позволяют отличить MS DOS от Windows NT, Word от Access, Pascal от Assembler. На следующем этапе система обучается на конкретных примерах, до тех пор, пока она не перестанет ошибаться. Полученная база знаний записывается в виде системы файлов на диске. Готовая система потом может быть использована учащимися после ее загрузки с внешнего носителя. Система позволяет также после работы с ней производить ее модификацию путем удаления или добавления новых переменных и исходов с последующим ее "дообучением".

Рассмотрим более подробно работу ЭС на примере классификации программного обеспечения ЭВМ. На начальном этапе мы задаем системе исходную информацию о числе узлов (их 2), числе переменных и исходов для каждого узла, а также названия переменных и исходов (модуль «инициализация системы»):

УЗЕЛ 1

Переменная 1 – Создает паки
Переменная 2 – Создает директории
Переменная 3 – Загружается автоматически
Переменная 4 – Однозадачный режим
Переменная 5 – Многозадачный режим
Переменная 6 – Графический интерфейс
Переменная 7 – Текстовый интерфейс
Переменная 8 – Создает файлы
Переменная 9 – Создает исполняемые файлы
Переменная 10 – Создает программы
Переменная 11 – Нуждается в трансляции
Переменная 12 – Создает тексты
Переменная 13 – Создает БД
Переменная 14 – Создает ЭТ
Переменная 15 – Создает графику
Переменная 16 – Осуществляет символьные преобразования
Переменная 17 – Управляет работой ЭВМ
Переменная 18 – Создает БЗ
Переменная 19 – Создает анимации
Переменная 20 – Производит импорт-экспорт
Исход 1 – Базовое ПО
Исход 2 – Система программирования
Исход 3 – Прикладное ПО

УЗЕЛ 2

Переменная 1 – Базовое ПО
Переменная 2 – Система программирования
Переменная 3 – Прикладное ПО
Переменная 4 – Процедурный
Переменная 5 – Операторный
Переменная 6 – Низкий уровень
Переменная 7 – Визуальный
Переменная 8 – Поддерживает работу сети
Переменная 9 – Осуществляет вычисления
Переменная 10 – Строит диаграммы
Переменная 11 – Строит БД
Переменная 12 – Строит ЭТ
Переменная 13 – Решает математические задачи

Переменная 14 – Допускает надстройки

Переменная 15 – Создает текст

Исход 1 – MS DOS

Исход 2 – WINDOWS 3.1, 3.11

Исход 3 – WINDOWS 98 и выше

Исход 4 – ASSEMBLER

Исход 5 – BASIC

Исход 6 – PASCAL

Исход 7 – PROLOG

Исход 8 – VISUA_ BASIC

Исход 9 – WORD

Исход 10 – PAINT

Исход 11 – EXCEL

Исход 12 – ACCESS

Исход 13 – MATHCAD

Исход 14 – POWER_POINT

После ввода исходных данных идет задание примера (модуль «ввод примера»), в котором для каждого исхода всех двух узлов указывается наличие его характерных признаков.

За вводом примера наступает этап тренировки системы (модуль «тренировка»), в котором по соответствующему алгоритму [Нейлор, 1991] заполняется массив правил, позволяющих по набору некоторых значений переменных (не обязательно всех) определить соответствующий им исход. Однако этап тренировки, как правило, формирует довольно «сырой» массив правил, который не всегда обеспечивает поиск адекватного переменным (атрибутам) исхода. Поэтому рекомендуется после тренировки провести обучение системы (модуль «обучение»). На этом этапе система, ориентируясь на сформированный ранее массив правил, ставит эксперту уточняющие вопросы относительно наличия тех или свойств выбранному исходу. Обучение заканчивается, как только система перестает ошибаться. Теперь результаты обучения заносятся в массив правил (модуль «запоминание»), при этом на внешнем носителе сохраняются и все ранее созданные массивы (в общей сложности семь файлов). Система готова к работе и ее можно использовать в качестве эксперта.

Работа пользователя (студента) начинается с загрузки ЭС с внешнего носителя в ОЗУ (модуль «загрузка БЗ») и последующего диалога с ней. При диалоге на экране монитора возникают вопросы, на которые надо ответить в форме ввода с клавиатуры значений 0 или 1 в зависимости от наличия или отсутствия указанных свойств у искомого объекта (исхода). Число задаваемых ЭС вопросов и порядок их следования целиком зависит от уровня ее обученности экспертом (преподавателем). Иногда ответ получается за 2-3 хода, порой задается более десятка вопросов (все зависит, конечно, от общего числа введенных в БЗ переменных и искомых исходов). В нашем примере с ПО ЭВМ для идентификации исхода ACCESS система в 1 узле задает вопросы: *создает тексты, создает файлы, создает исполняемые файлы* и принимает решение, что это - *Прикладное ПО*, а потом во 2 узле спрашивает: *создает графику, проводит*

вычисления, создает БД.

Ниже следует протокол работы ЭС по определению исхода *WINDOWS 98* и выше:

УЗЕЛ 1

- (7) – Текстовый интерфейс есть <0/1>? 0
 - (8) – Создает файлы есть <0/1>? 0
 - (10) – Создает программы есть <0/1>? 0
 - (15) – Создает графику есть <0/1>? 0
 - (20) – Производит импорт-экспорт есть <0/1>? 0
 - (1) – Создает папки есть <0/1>? 1
 - (2) – Создает директории есть <0/1>? 0
 - (3) – Загружается автоматически есть <0/1>? 1
 - (4) – Однозадачный режим есть <0/1>? 0
 - (5) – Многозадачный режим есть <0/1>? 1
 - (9) – Создает исполняемые файлы есть <0/1>? 0
 - (13) – Создает БД есть <0/1>? 0
- Предлагается Базовое ПО в качестве исхода.

УЗЕЛ 2

- (11) – Строит БД есть <0/1>? 0
- (4) – Процедурный есть <0/1>? 0
- (6) – Низкий уровень есть <0/1>? 0
- (13) – Решает математические задачи есть <0/1>? 0
- (5) – Операторный есть <0/1>? 0
- (7) – Визуальный есть <0/1>? 1
- (14) – Допускает надстройки <0/1>? 0
- (8) – Поддерживает работу сети есть <0/1>? 1
- (10) – Строит диаграммы есть <0/1>? 0

Предлагается *WINDOWS 98* и выше в качестве возможного исхода.

Анализируя список переменных и исходов БЗ для классификации ПО ЭВМ, можно заметить, что он обладает некоторой повторяемостью: часть переменных первого и второго узлов совпадают (*создает БД, Создает ЭТ, создает текст* и пр.). Кроме того, приходится иногда для надежности работы ЭС исходы 1 узла делать переменными для второго. Такое явление в экспертных системах допускается, но в обучающих ЭС это не допустимо, так вызывает у студентов некоторое недоумение – зачем дважды спрашивать одно и то же. Однако в данной БЗ такое повторение просто необходимо в силу многочисленности исходов 2 узла. Очевидно, что признак «Создает БД» необходим, чтобы отличить прикладное ПО от базового и системы программирования в 1 узле, и ACCESS от EXCEL – во втором. Одним из путей избавления от этого «недостатка» является увеличение числа узлов, что и показано в следующем примере. Первоначально нами была построена двухузловая БЗ «Типизация данных», однако добиться ее безошибочной работы даже после многочисленных сеансов обучения не удалось. Переход на три узла решил все проблемы.

Итак, рассмотрим теперь работу ЭС на примере классификации типов данных языка Паскаль. Здесь уже задаются 3 узла, каждый из которых определяется своим набором переменных:

УЗЕЛ 1

- Переменная 1 – Описывается TYPE-VAR
- Переменная 2 – Содержит другие величины
- Переменная 3 – Вводится одним READ
- Переменная 4 – Выводится одним WRITE
- Переменная 5 – Имеет индексацию своих элементов
- Переменная 6 – Содержит адрес ячейки ОЗУ
- Переменная 7 – Описывается VAR
- Исход 1 – ПРОСТОЙ ТИП
- Исход 2 – СПЕЦИАЛЬНЫЙ ТИП
- Исход 3 – СТРУКТУРНЫЙ ТИП

УЗЕЛ 2

- Переменная 1 – Создается программой
- Переменная 2 – Образует динамические объекты
- Переменная 3 – Создается пользователем
- Переменная 4 – Занимает стандартный объем памяти
- Переменная 5 – Содержит текст
- Переменная 6 – Аналог линейного массива
- Исход 1 – ВСТРОЕННЫЙ ТИП
- Исход 2 – ПОЛЬЗОВАТЕЛЬСКИЙ ТИП
- Исход 3 – СПЕЦИАЛЬНЫЙ ТИП
- Исход 4 – СТАТИЧЕСКИЙ ОБЪЕКТ
- Исход 5 – ДИНАМИЧЕСКИЙ ОБЪЕКТ

УЗЕЛ 3

- Переменная 1 – Ординальный
- Переменная 2 – Числовой
- Переменная 3 – Содержит только два значения
- Переменная 4 – Внутренняя индексация
- Переменная 5 – Внешняя индексация
- Переменная 6 – Фиксированный набор значений
- Переменная 7 – Ограниченный набор значений
- Переменная 8 – Имеет одного наследника
- Переменная 9 – Имеет двух наследников
- Переменная 10 – Упорядоченность элементов
- Переменная 11 – Удаляется из памяти программой
- Переменная 12 – Объединяет разные типы
- Переменная 13 – Располагается в ВЗУ
- Исход 1 – READ
- Исход 2 – INTEGER
- Исход 3 – CHAR
- Исход 4 – BOOLEAN
- Исход 5 – ИНТЕРВАЛЬНЫЙ
- Исход 6 – ПЕРЕЧИСЛИМЫЙ
- Исход 7 – ССЫЛОЧНЫЙ
- Исход 8 – STRING
- Исход 9 – ARRAY
- Исход 10 – SET
- Исход 11 – RECORD
- Исход 12 – FILE
- Исход 13 – ЛИНЕЙНЫЙ СПИСОК
- Исход 14 – СБАЛАНСИРОВАННОЕ ДЕРЕВО
- Исход 15 – ДЕРЕВО ПОИСКА

В базе знаний «Типизация данных» идентификация 15 типов данных осуществляется в три этапа: сначала определяется принадлежность одной из групп типов (простая, специальная, структурная), затем уточняется принадлежность

более узкому классу (*встроенный, пользовательский, специальный, статический, динамический*), а затем определяется окончательный результат.

Определение простого типа данных, соответствующего типам *REAL, INTEGER, CHAR* и *BOOLEAN* (узел 1) системе удается за 5 шагов путем постановки вопросов, соответствующим номерам переменной (1,4,3,6,2), при ответах на них (0,1,0,0,0).

Для *ИНТЕРВАЛЬНОГО* и *ПЕРЕЧИСЛИМОГО* типов – это удается за 6 шагов при чередовании переменных с номерами: (1,3,5,6,7,2) и их значениях (0,0,0,0,1,0).

ВСТРОЕННЫЙ тип (узел 2) определяется за 5 шагов: (1,4,3,6,2) → (0,1,0,0,0), а *ПОЛЬЗОВАТЕЛЬСКИЙ тип* (узел 2) – за 3 шага: (1,4,3) → (0,0,1).

В 3 узле типы *INTEGER, BOOLEAN, ПЕРЕЧИСЛИМЫЙ ТИП* узнаются соответственно по следующей системе вопросов и ответов:
(10,7,1,6,3,11,5,4,2,13,12) → (0,1,1,1,0,0,0,1,0,0)
(10,7,1,6,3,4,9,2,11,13) → (0,1,1,1,1,0,0,0,0,0)
(10,7,2,11,6,1) → (0,0,0,0,1,1).

Представители специального типа *ССЫЛОЧНЫЙ* и *STRING* обладают совсем разными характеристиками, вот почему этот тип имеет совсем разный набор переменных для его определения. По этим же соображениям исход *СПЕЦИАЛЬНЫЙ* фигурирует одновременно в 1 и 2 узлах. Итак, для выхода на специальный тип относительно ссылочного типа необходимо ответить соответственно на следующие вопросы:

(1,3,5,6) → (0,0,0,1)
(1,4,3,6,2) → (0,1,0,0,1).

СПЕЦИАЛЬНЫЙ тип для двух узлов в случае типа *STRING* определяется так:

(1,3,2) → (0,1,1)
(1,4,3,2,6) → (0,0,0,0,1).

Окончательно в узле 3 для идентификации типов *ССЫЛОЧНЫЙ ТИП* и *STRING* система ставит вопросы в следующем порядке:

(10,7,1,6,4,5,2,13) → (0,1,0,1,0,0,0,0)
(10,7,1,6,3,4,13,5,8,2) → (0,1,0,0,0,1,0,0,0,0).

СТРУКТУРНЫЙ тип определяется системой по одному вопросу “*Описывается TYPE-VAR*”. Исход *СТАТИЧЕСКИЕ ОБЪЕКТЫ* для всех ее четырех типов *ARRAY, SET, RECORD, FILE* определяется вопросами:

(1,4,3,2,6,5) → (0,0,0,0,0,1).

Исход *ДИНАМИЧЕСКИЕ ОБЪЕКТЫ* определяется еще быстрее – всего за два хода:
(1,4) → (1,0).

Это объясняется тем, что характерный признак «*Создается программой*» является присущим только этому типу данных. Как правило, такого рода

переменные экспертная система всегда выносит на начало постановки вопросов (согласно заложенному в нее алгоритму [Нейлор, 1991]).

Ниже следуют вопросы и ответы на них, которые определяют поиск статических объектов *ARRAY, SET, RECORD, FILE* в узле 3:

(10,7,2,11,6,1,5,13,12) → (0,0,0,0,1,0,1,0,0)
(10,7,4,11,13,6,2,3,5,9) → (1,1,1,0,0,0,0,0,0)
(10,7,2,11,6,1,5,12,13,4) → (0,0,0,0,1,0,0,1,0,0)
(10,7,2,11,9,8,1,4,13) → (0,0,1,1,0,0,0,1,1).

ЛИНЕЙНЫЙ СПИСОК, СБАЛАНСИРОВАННОЕ ДЕРЕВО и *ДЕРЕВО ПОИСКА* находятся, соответственно, с помощью следующей последовательности вопросов и ответов:

(10,7,2,11,8) → (0,0,0,1,1)
(10,7,2,11,8,12) → (0,0,0,1,0,0)
(10,7,4,11) → (1,0,0,1).

Для целей обучения данная база знаний может быть использована в различных формах.

1. Работа с готовой и обученной базой знаний

Эта форма работы используется для чистого контроля знаний, когда учащемуся дается задание добиться получения на выходе всех окончательных объектов (от MS DOS до PowerPoint, от REAL до ДЕРЕВА ПОИСКА). При этом преподаватель может просмотреть протокол ответов учащихся на вопросы ЭС.

2. Обучение БЗ

Здесь преподаватель вводит в БЗ только переменные и исходы, а учащийся должен обучить ее до получения правильных ответов. Можно также частично обучить БЗ и предложить учащимся добиться получения от нее безошибочных ответов путем ввода достаточного числа обучающих примеров.

3. Модификация БЗ

На этом этапе работы с готовой БЗ предлагается изучить ее поведение, выявить лишние (неактивные) переменные, удалить их из базы и провести последующее “переобучение”.

4. Добавление к БЗ новых объектов

Эта форма работы полезна тем, что учащиеся, которые добавляя новый объект, должны увидеть, достаточно ли существующих в БЗ признаков для идентификации нового объекта путем переобучения системы, или же необходимо дополнить этот список новой (новыми) переменными.

5. Создание новой БЗ

В этом случае учащийся должен предварительно продумать список всех переменных и исходов, а затем ввести их в ЭС и произвести ее обучение.

Отметим также, что созданная нами оболочка ЭС написана на языке Паскаль, она является универсальной и в нее можно поместить любое наполнение из любой предметной области.

Составитель (разработчик) этого наполнения должен решить, по существу, три проблемы:

1. Выбрать правильное (желательно минимальное) число узлов для для правильной идентификации (узнавания) объектов.
2. Подобрать для каждого узла соответствующие признаки (переменные) каждого его объекта (исхода).
3. Обучить путем многократной «прогонки» программы базу знаний ЭС, используя при этом либо удаление неиспользуемых признаков, либо добавления новых, если система допускает ошибки.

ЗАКЛЮЧЕНИЕ

В данной статье был рассмотрен пример построения обучающей экспертной системы на основе модульного проектирования, предусматривающего создания равноуровневых баз знаний. На двух рассмотренных случаях была показана технология выбора узлов, переменных и исходов, которые составляют компоненту «факты» базы знаний. Был также представлен механизм создания «правил» путем ввода в БЗ примера, где каждому объекту (исходу) присваивается набор его характеристик (атрибутов), и последующего обучения системы.

Библиографический список

[Нейлор, 1991] Нейлор К. Как построить свою экспертную систему: Пер. с англ. – М: Энергоатомиздат, 1991. – 286 с.

[Касторнов, 2002] Касторнов А.Ф. Экспертные системы как средства систематизации и контроля знаний. Ученые записки ИИО РАО, выпуск 7, Москва, 2002.- с. 98-114.

[Касторнова, 2011] Касторнова В.А. Современное состояние научных исследований и практико-ориентированных подходов к созданию и функционированию образовательного пространства. Монография. Череповец: ЧГУ, 2011. – 461 с.

TECHNOLOGY OF COMPONENT (MODULAR) DESIGN OF EXPERT SYSTEM FOR SYSTEMATIZATION AND CONTROL OF KNOWLEDGE

Kastornov A.F. *, Kastornova V.A. **

* *Cherepovets State University, Cherepovets,
Russia*

a_kastornov@mail.ru

***Institute of Informatization of Education
Russian Academy of Education, Moscow, Russia*

kastornova_vasya@mail.ru

In this work questions of creation of the training expert system (TES), urged to serve one of pedagogical instruments of systematization and control of knowledge of students are considered. The structure of OES offered by authors is based on the modular principle of its construction and includes the modules, allowing to create the knowledge base in some subject domain, to fill it with the contents, to make its control

(training), and then to use when carrying out occupations.

INTRODUCTION

The purpose of this work is practical realization of technology of modular design of expert system. In work the program creating the knowledge base which includes the modules, allowing to pack the various levels (knots) containing the list of questions (variables) which are raised by ES to the user correctly to identify this or that object (outcome) in some subject domain is described Pascal.

MAIN PART

The expert system consists of the knowledge base, containing the facts and rules, subsystems of a conclusion of an explanation, acquisition of knowledge and the dialogue processor. Rules serve in the knowledge base for submission of informal rules of the reasoning developed by the expert on the basis of experience of its activity. The expert systems which are carrying out training, subject to diagnostics, debugging and correction behavior of the trainee. Training systems create model of that the being trained knows and as it applies this knowledge to a solution. The expert system of educational appointment consists of modules: system initialization, input of examples, training, training, knowledge base storing on the external carrier (in the form of a set of massifs of the facts and rules), removal and addition of variables and outcomes, knowledge base loading from VZU in the RAM, and also the main module of dialogue with the user. Originally for BZ initial information on number of knots, number of variables and outcomes for each knot and the name of variables is set. After input of basic data there is a task of an example in which for each outcome of all knots existence of its characteristic signs is specified. Behind input of an example there comes a stage of training of system in which the array of rules is filled. After training it is trained systems. Results of training are brought in an array of rules. For training this knowledge base can be used in various forms: work with the ready and trained knowledge base, BZ training, BZ modification, addition to BZ of new objects, creation of new BZ.

CONCLUSION

In this article an example of creation of training expert system on the basis of the modular design providing creations of raznourovnevy knowledge bases was reviewed. On two considered cases the technology of a choice of knots, variables and outcomes which make to the facts component of the knowledge base was shown. The mechanism of creation was also presented "governed" a way of input to example BZ where the set of its characteristics (attributes), and the subsequent training of system is appropriated to each object (outcome).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 658.26:004.8

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ТЕСТИРОВАНИЯ ПРОГРАММНЫХ СРЕДСТВ НА ОСНОВЕ МОДИФИЦИРОВАННЫХ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

Бурдо Г.Б.* , Палюх Б.В.* , Мельникова В.В.**.

**Тверской государственной технической университет, г.Тверь, Россия*

gbtms@yandex.ru

pboris@tstu.tver.ru

***ООО «Главный испытательный сертификационный центр программных средств вычислительной техники», г. Тверь, Россия*

viklipse@mail.ru

В статье рассмотрены проблемы проведения сертификационных тестирований программных средств в скомпилированном виде. Предложен подход к созданию тестовой базы с использованием генетических алгоритмов.

Ключевые слова: сертификационное тестирование, автоматизированная тестовая база, генетические алгоритмы, тестовое покрытие.

Введение

При разработке программных средств (ПС), к качеству которых предъявляются повышенные требования, перед вводом системы в эксплуатацию требуется подтверждение со стороны уполномоченного органа (обычно государственного) соответствия ее эксплуатационных характеристик заданным критериям. Такое соответствие определяется в ходе сертификации ПС.

В настоящее время наблюдается достаточно высокий рост программных средств, причем очень часто программные средства разрабатываются в сжатые сроки и при ограниченных бюджетах проектов. Затем они постоянно дорабатываются с целью исправления выявленных ошибок или отслеживания изменений предметной области применения ПС (например, меняется законодательство). В результате ПС становятся «версионными», причем обновления могут происходить довольно часто. Каждая новая версия ПС требует повторной сертификации.

При этом обновления ПС могут происходить быстрее, чем проведение сертификации очередной его версии. Кроме того, затраты на проведение сертификации постоянно обновляющегося ПС могут свести на нет все преимущества от его использования.

В связи с этим остро встает проблема как можно более быстрого прохождения сертификации очередной версии такого ПС. Особенно трудным представляется проведение тестирования ПС без наличия исходных текстов, например, в рамках сертификации ПС по ГОСТ Р.

Сертификация ПС по ГОСТ Р, принятая в России, предусматривает проведение их тестирования, т.е. контролируемое выполнение программы на конечном множестве наборов данных (КМНД) и анализ этого выполнения с целью обнаружения ошибок [Котов, 2010].

Становится понятным, что рациональное ограничение КМНД во многом предопределяет трудоемкость и корректность тестирования, причем неавтоматизированное определение КМНД приводит к дополнительным трудозатратам.

Следовательно, задача формирования оптимального КМНД является актуальной.

Постановка задачи

Известны способы тестирования ПС на основе использования генетических алгоритмов (ГА) [Мельникова, 2011]. Критериями полноты тестового покрытия обычно являются: количество покрытых операторов в исходном коде, количество покрытых ветвей в графе потока управления, количество покрытых путей графа потока управления и т.д.

[Котляров, 2004]. Однако проверка становится возможной при наличии исходных текстов программ, в то время как государственные органы сертификации работают со скомпилированными программными средствами.

Поэтому, логичной является постановка задачи создания автоматизированной системы формирования рационального КМНД для тестирования скомпилированных ПС. В качестве аппарата для создания КМНД авторами предлагается использование модифицированных ГА.

Методика представления (топология) исходных данных

В основу создания КМНД для контролируемых расчетов положен анализ структуры и параметров исходных данных (ИД), используемых в ПС при их выполнении. Их предлагаемое представление позволит при сертификации ПС смоделировать их тестирование с использованием исходных текстов программ.

Топология ИД была рассмотрена и представлена в виде следующих типов переменных:

1. Множества ИД вещественного случайного вида, $V \equiv \{V_k\}, k = \overline{1, m}$ принимающие случайные значения в определенном интервале:

$$V_k = \{b_{kp}\}, V_k^{\min} \leq b_{kp} \leq V_k^{\max}, \rho = \overline{1, \infty} \quad (1)$$

где b_{kp} – случайное значение входной переменной V_k ;

V_k^{\min}, V_k^{\max} – минимальное и максимальное значение входной переменной V_k ; m – их количество.

Множество $V = \cup V_k$ – все массивы ИД случайного типа, определяющие их возможные наборы. Данные 1-го типа, в основном, подчиняются закону нормального распределения.

2. Множества ИД целочисленного вида, $A = \{A_i\}, i = \overline{1, n}$ (2-й тип):

$$A_i = \{A_{ij}\}, A_i^{\min} \leq a_{ij} \leq A_i^{\max}, j = \overline{1, t_i}, \quad (2)$$

$$a_{ij} - \text{int},$$

где n – число целочисленных переменных; t – число значений A_i -ой переменной; A_i^{\min} и A_i^{\max} – минимальное и максимальное значение A_i -ой переменной.

В отличие от данных 1-го типа, используемых непосредственно для получения численного значения результата при использовании ПС, данные 2-го типа используются для определения основного пути графа вычислений. К этим же данным, естественно, относятся и ИД символического типа

(буквы, слова, фразы) из определенного ограниченного набора. Это корректно, т.к. с точки зрения влияния на ход выполнения программы они – переменные одного смыслового содержания.

3. Множества логических (булевых) переменных в составе ИД, $L = \{L_q\}, q = \overline{1, d}, d$ – их количество.

$$L_q = \{l_{qr}\}, r = 1, 2, l_{qr} = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}, \quad (3)$$

Эти исходные данные (3-й тип) определяют разветвление пути графа вычислений. Им так же по смыслу соответствуют и синтаксические переменные, состоящие из наборов по 2 буквы (по 2 слова, фразы).

Методика и алгоритм создания КМНД

Методика создания КМНД состоит в том, чтобы при тестировании скомпилированных программ смоделировать такой вычислительный процесс, который бы косвенно отражал процесс и критерии тестирования ПС в исходных текстах. Такие критерии, как количество покрытых ветвей и путей в графе потока управления, косвенно определяется числом переборов ИД переменных из типов 2 и 3. Корректность работы программы в диапазоне возможных числовых значений исходных данных, используемых для непосредственных расчетов, определяется тем, насколько перекрыта и с какой дискретностью область значений переменных 1-го типа.

Рассмотрим генотип кодирования ИД для тестирования. Топологию исходных данных удобнее представлять в виде генотипа, состоящего из трех хромосом, каждая из хромосом отвечает за ИД соответствующего типа. Структура хромосом показана на рис. 1.

Генами хромосомы являются переменные A_i, V_k, L_q .

1.	V_1	...	V_k	...	V_m
2.	A_1	...	A_i	...	A_n
3.	L_1	...	L_q	...	L_d

Рисунок 1 - Структура хромосом

Алгоритм создания КМНД построен следующим образом.

На первом этапе создаются наборы для обеспечения тестового покрытия ветвей и путей в графе потока информации, т.е. варьируют хромосомами типов 2 и 3 при фиксированном значении хромосомы 1. В этом случае, диапазон всех значений переменных V_k разбивается на нечетное одинаковое число интервалов I (рекомендуется 7, 9 или 11).

Значение каждой величины V_k случайным образом генерируется из срединного интервала.

На первом этапе наборы хромосом 2-го и 3-го типов строятся из предположения, что распределение значений этих ИД подчиняется закону равной вероятности. Учитывая, что мы не знаем места ветвления программы, для полного тестового покрытия необходимы 100% генерация возможных вариантов, но на практике бывает достаточным ограничить перебор, задав для формирования КМНД процент перебора данных в переменных целочисленного 2-го типа и обеспечить полный перебор данных 3-го булева типа.

Исходная популяция хромосом для этапа 1 получается следующим образом. Из хромосомы 1 со значениями V_k из срединного интервала I_{cp} , хромосом 2 и 3 со случайными значениями $\{a_{kj}\}$ и $\{l_{qr}\}$ получается первый набор из трех хромосом.

Следующий, III-й набор из трех хромосом получается следующим образом: а) 1 и 2-я хромосомы остаются неизменными; б) в третьей хромосоме гены l'_{qr} дополняют гены l_{qr} до полного множества $\{l_{qr}\} = L_q$ (т.е. берется второй элемент бинарного множества):

$$l'_{qr} = L_q \setminus l_{qr}. \quad (4)$$

Четвертый и последующие (производные) наборы хромосом на первом этапе строятся по следующему алгоритму:

1 шаг. Берется из исходной популяции хромосом набор I.

2 шаг. Указанный набор фиксируется в новом производном наборе хромосом.

3 шаг. В хромосоме 2 производного набора случайным образом мутируется n -ый ген (не допуская повторов с предыдущими вариантами), получаем новый тестовый набор. Тестовый набор заносится в память.

4 шаг. В полученном на шаге 3 тестовом наборе хромосома 3 заменяется хромосомой из исходного набора II. Заносим тестовый набор в память;

5 шаг. Проверяется условие окончания мутации гена n (по проценту покрытия значений интервала от A_i^{\min} до A_i^{\max}). Если нет, то возврат на шаг 3, иначе – шаг 6.

6 шаг. Проверяется условие окончания мутации предшествующего ($n - 1$) гена. Если нет, то переход к оператору 7, иначе – шаг 8.

7 шаг. В последнем тестовом наборе случайным образом, исключая повторы, в хромосоме 2 мутируется предшествующий ($n - 1$) ген. Переход на шаг 3.

8 шаг. Проверяется условие окончания мутации предшествующего ($n - 2$) гена. Если нет, то переход к оператору 9, иначе – шаг 10;

9 шаг. В последнем тестовом наборе случайным образом, исключая повторы, в хромосоме 2 мутируется предшествующий ($n - 2$) ген. Переход на шаг 3.

10 шаг. Проверяется условие окончания мутации ($n - 3$) гена. Если нет, то переход к шагу 11, иначе к шагу 12;

11 шаг. В последнем тестовом наборе случайным образом, исключая повторы, в хромосоме 2 мутируется ($n - 3$) ген. Переход на шаг 3.

... и т.д.

Предпоследний шаг. Проверить окончание мутации ($n - (n - 1)$) гена. Если нет, то переход к последнему шагу, иначе окончание генерации тестовых наборов.

Последний шаг. В тестовом наборе случайным образом мутируется ($n - (n - 1)$) ген. Переход на шаг 3.

Отдельного разговора заслуживает вопрос о процентном соотношении мутируемых вариантов генов во второй хромосоме. Для гена A_1 , очевидно, должен быть обеспечен полный перебор. Для гена A_2 достаточным будет обеспечить проверку для $K_2 \cdot \frac{100\%}{t_1}$ процента вариантов, где K_2 -

коэффициент запаса, t_1 - количество вариантов гена A_1 . K_2 исходя из наихудшего варианта вероятностного сложения полей рассеивания переменных, рекомендуется брать равным 1,73.

Тогда процент мутации i -го гена, P_{ij} ; в общем случае:

$$P_i = \frac{k^{i-1} \cdot 100\%}{t_1 \cdot t_2 \cdot \dots \cdot t_{i-1}} = \frac{k^{i-1} \cdot 100\%}{\prod_1^i t_i}, \quad (5)$$

где \prod - знак произведения.

На втором этапе выполняется формирование КМНД для тестирования ПС с точки зрения выполнения корректности расчетов в области существования переменных, определяющих числовые данные вычислений. С этой целью строятся варианты наборов из мутируемой хромосомы 1 и постоянного состава генов в хромосомах 2 и 3 из набора I первого этапа.

Исходная популяция хромосом состоит из двух наборов, определяемых хромосомой 1. В первом наборе в хромосоме присутствуют максимальные значения V_k , во втором – минимальные.

Остальные тестовые наборы получают мутацией хромосомы 1, помня, что диапазоны каждого гена V_k разбиты на I число интервалов.

Предположив, что вероятность ввода значений исходных данных подчиняется закону Гаусса, и, учитывая, что крайние значения V_k имеются в базовой (исходной) популяции при числе $I = 7 \div 11$ мутация генов в пределах срединного и двух смежных интервалов обеспечит перекрытие всего диапазона изменения V_k не менее 60%.

Таким образом, алгоритм формирования КМНД на втором этапе следующий.

Шаг 1. Формируем исходную популяцию хромосом.

Шаг 2. Берем из исходной популяции хромосом набор I.

Шаг 3. В наборе I выполняем случайным образом неповторяющуюся мутацию в гене V_m хромосомы I (в пределах срединного и в каждом смежном интервале). Запоминаем тестовый набор.

Шаг 4. Определяем окончание мутации гена V_m (по 1 значению в срединном и по 1-му в каждом смежном интервале). Если да – переход к шагу 5, нет – возврат к шагу 3.

Шаг 5. Проверить окончание мутации V_{m-1} (предшествующего) гена аналогично гену V_m . Если нет, переход на шаг 6, если да – переход на шаг 7.

Шаг 6. Случайным образом осуществляем мутацию гена V_{m-1} (аналогично гену V_m). Возврат на шаг 3.

Шаг 7. Проверяем окончание мутации гена V_{m-2} . Если нет – переход на шаг 8, если да – переход на шаг 9.

и т.д.

Предпоследний шаг. Проверить окончание мутации по $V_{m-(m-1)}$ гену. Если да, то окончание генерации тестовых наборов, нет – переход к последнему шагу.

Последний шаг. Случайным образом (аналогично гену V_m) мутируется $V_{m-(m-1)}$ ген. Возврат на шаг 3.

Методика тестирования ПС

Предлагаемая методика тестирования ПС следующая. Первоначально тестируются ПС с использованием КМНД, полученных на первом этапе. Его успешное завершение подтверждает, как минимум, корректность ветвей и путей в графе потока информации.

Во вторую очередь тестируются программные средства с использованием КМНД, полученных на этапе 2, что позволит, как минимум, утверждать корректность выполнения расчетов на всем диапазоне возможных значений входных данных.

Заключение

Изложенные положения были подтверждены сравнением тестирования ПС в скомпилированном виде с тестированием программ в исходных текстах. Сгенерированные наборы данных обеспечивали необходимую полноту тестового покрытия.

В настоящее время методика и разработанные на ее основе ПС, проходят опытную проверку в ООО «Главный сертификационный испытательный центр программных средств вычислительной техники» (г. Тверь, ул. Ржевская, д.10, 170023).

Библиографический список

[Мельникова, 2011] Мельникова В.В. Тестирование программ с использованием генетических алгоритмов. // В.В. Мельникова, Б.В. Палух, С.Л. Котов, М.А. Проскуряков. - Программные продукты и системы.-2011.-№4.-С.114-117.

[Котляров, 2004] Котляров В.П. Основы современного тестирования программного обеспечения, разработанного на с#:учебное пособие// В.П.Котляров, Т.В. Коликова. - СПб, 2004.- С. 55-57.

[Котов, 2010] Котов С.В. Проведение сертификационных испытаний часто обновляющихся ПС с использованием автоматизированной тестовой базы.// С.В. Котов, А.А. Демирский, В.В. Мельникова.- Вестник ВНИИМАШ, 2010.

AUTOMATED SYSTEM OF TESTING OF SOFTWARE ON THE BASIS OF THE MODIFIED GENETIC ALGORITHMS

Burdo G.B. *, Palyukh B.V. *,
Melnikova V.V. **

*Tver State Technikal University, Tver, Russia
gbtms@yandex.ru

**Chief Center for Software Certification and
Testing (GIC PS VT), Tver, Russia
viklipse@mail.ru

The article considers the problems of carrying out certification tests of software in compiled form. An approach is proposed to create a test database using genetic algorithms

Keywords: certification test, automated test framework, genetic algorithms, test coverage.

СЕКЦИЯ 7.

**ЛОГИКО-СЕМАНТИЧЕСКИЕ МОДЕЛИ ПРИКЛАДНЫХ
ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ**

SECTION 7.

LOGICO-SEMANTIC MODEL OF APPLIED INTELLIGENT SYSTEMS



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 519.816

ИСПОЛЬЗОВАНИЕ ПРОДУКЦИОННО-ФРЕЙМОВОЙ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В АВТОМАТИЗИРОВАННОМ ЛОГИЧЕСКОМ ПРОЕКТИРОВАНИИ ЗАКАЗНЫХ ЦИФРОВЫХ СБИС

Бибило П.Н., Романов В.И.

*Объединенный институт проблем информатики НАН Беларуси
г. Минск, Республика Беларусь*

bibilo@newman.bas-net.by

rom@newman.bas-net.by

Описывается использование продукционно-фреймовой модели представления знаний в системе автоматизированного логического проектирования сложных функциональных блоков заказных цифровых СБИС. Основное внимание уделяется задачам оптимизации функционально-структурных описаний, синтеза и верификации иерархически организованных проектов. Комбинированные маршруты проектирования рассматриваются в виде стратегий, представляющих собой совокупность продукций.

Ключевые слова: продукции; стратегии; фреймы; логические схемы; оптимизация; синтез; верификация.

ВВЕДЕНИЕ

Увеличение степени интеграции сверхбольших интегральных схем (СБИС) и систем на кристаллах приводят к усложнению и соответствующих систем автоматизированного проектирования (САПР). Усложняется и центральный этап логического проектирования, на котором осуществляется переход от алгоритмических (поведенческих) и функциональных описаний к логическим схемам в соответствующих технологических библиотеках. В практике проектирования предлагается достаточно много способов как оптимизации функциональных и структурных описаний, так технологического отображения (собственно синтеза) в той или иной библиотеке проектирования (базисе). Для выбора соответствующих способов и маршрутов автоматизированного проектирования в САПР предлагается использовать экспертные знания, это касается не только САПР электронных цифровых (дискретных) устройств [Паркер, 1987], но и САПР объектов машиностроения [Евгеньев, 2009]. Проведя анализ промышленных систем проектирования цифровых устройств, можно заключить, что областью знаний экспертов являются знания по организации маршрутов проектирования в зависимости от целей проектирования, технологического базиса и характеристик реализуемого проекта. Использование экспертных знаний может значительно повысить эффективность проектирования, однако для этого требуется

разработка соответствующих формализмов представления (описания) знаний [Рыбина, 2010] и средств использования таких описаний в процессе проектирования. В данной работе анализируется область логического проектирования цифровых схем, выясняется назначение и роль экспертных знаний, сообщается о формализме их описания в САПР.

1. Задачи логического проектирования для иерархических описаний проектов

Введение иерархии в описания проекта цифровой системы позволяет уменьшить сложность описания и увеличить размерность решаемых задач проектирования. *Задача синтеза* логической схемы: задано иерархическое описание и базис синтеза, требуется получить логическую схему, реализующую функции, заданные исходным описанием. Подходы к решению задачи синтеза зависят от глобальных направлений организации процесса синтеза и связаны с обработкой иерархии исходного описания и обработкой представлений листовых вершин в дереве иерархии проекта. Перечислим эти основные подходы к синтезу схем для иерархически организованных проектов.

1. Иерархия исходного описания не изменяется, синтез схемы ведется согласно иерархии.

2. Глобальный синтез схемы. Такой подход возможен для проектов меньшей размерности, так он предполагает устранение иерархии и получение проекта в виде одного листа.

3. Введение новой иерархии описания, которое можно осуществить двумя способами: разбиением больших блоков на более мелкие блоки либо укрупнением мелких блоков в более крупные.

4. Локальная модификация иерархии.

Данные подходы к изменению иерархии описания при решении задач синтеза логических схем могут быть применены и для решения других задач логического проектирования: анализа, верификации, моделирования и построения тестов. Например, чтобы решить задачу анализа, можно, различным образом изменяя иерархию описания, вычислять реализуемые схемой подфункции и их суперпозиции и получать функции, реализуемые схемой, зависящие только от входных переменных.

Для задач технологически-независимой оптимизации в литературе предложено большое число самых разнообразных методов [Закревский и др., 2007]. Среди них обычно выделяются методы оптимизации двухуровневых и многоуровневых представлений, а основным критерием оптимизации признан критерий минимальности числа литералов (переменных либо их инверсий) в алгебраических представлениях функций. Минимизация сложности логических описаний достигается за счет выделения одинаковых частей описаний и эффективного использования операций инверсирования.

Различные методы синтеза схем ориентируются на различные формы исходного представления функций. Наибольшее развитие в настоящее время получили методы, оперирующие с алгебраическими представлениями функций и предназначенные для синтеза схем в заданной библиотеке проектирования заказных СБИС и декомпозиционные методы [Sasao, 1996]. Известные в литературе [Darringer, 1984], [Uehara, 1985] экспертные системы логического проектирования ориентированы в основном на компилятивное проектирование на этапе технологического отображения. В таких системах логический вывод (возможность применения правил) связан с комбинаторным поиском по базе данных. Роль эксперта при создании таких систем сводится к оформлению правил замены одних подсхем другими.

2. Экспертные знания

Общий взгляд на задачи логического проектирования для иерархических проектов состоит в том, что при оптимизации, синтезе и верификации проектов могут изменяться гранулы (объемы) листовых описаний, вид деревьев иерархии описаний, способы обработки листовых описаний в зависимости от размерности проекта в целом, от размерностей листовых описаний и от уровня (функционального объема) гранул базисных элементов.

Принятие решений по выбору представлений и выбору уровней гранулярности листовых описаний, изменению дерева иерархии описания, т. е. методам

слияния и разбиения узлов и поддеревьев является областью действий проектировщика – пользователя САПР при работе с конкретным проектом. Проектировщик устанавливает параметры, требуемые для работы программных модулей, принимает решения по выбору технологического базиса для блоков проекта, выполняет проектные операции и процедуры оптимизации и декомпозиции проекта, анализирует данные, оценивает получаемые результаты. В результате действий проектировщика в конкретных ситуациях получается последовательность проектных операций и процедур, приводящая к реализации исходных спецификаций – превращению исходного описания в результирующую логическую схему. Знания проектировщика выражаются в способности принимать рациональные (эффективные) решения при управлении системой проектирования для достижения целей проектирования. При разработке САПР, содержащей средства интеллектуальной поддержки, предварительному анализу подвергается область действий проектировщика. Этот анализ опирается на понятие проектной ситуации, куда включаются цели проектирования, параметры используемых в САПР описаний объектов и базисов проектирования и параметры программных модулей.

3. Продукционно-фреймовая модель представления знаний

Для использования экспертных знаний требуется их формализация. В проблемной области логического проектирования предлагается использовать продукционно-фреймовую модель представления знаний. Фреймы [Минский, 1979] позволяют выделить «рамки» описания проектных ситуаций, продукции дают возможность определения последовательностей проектных действий. Понятие *продукции* формально определяется как запись

$$P^i : P^i \rightarrow B^i,$$

где P^i – условие применимости, B^i – заключение или действие, которое имеет место при истинности P^i [Кузнецов, 1989]. При записи условий применимости продукции используются не предикатные символы, а выражения, представляющие данные символы, условие применимости P^i представляет собой скобочное логическое *выражение*, в котором могут быть использованы три логические операции: конъюнкция, дизъюнкция и отрицание (инверсии) над символами логических переменных $P_1^i, P_2^i, \dots,$

P_q^i , являющихся предикатами. *Предикатом* называется логическая функция, которая при конкретных значениях своих аргументов превращается в высказывание со значением из множества {«истина», «ложь»}. Таким образом, левая часть продукции может принимать только

одно из двух значений – «ложь» либо «истина». Аргументами каждого из предикатов могут быть только однотипные атрибуты. Допустимые типы атрибутов: символьный, числовой, логический. Каждый предикат задается своим выражением. Тождественно истинный предикат задает безусловную продукцию. В качестве правых частей V^i продукций используются некоторые выполняемые проектные действия. Упорядоченные системы пронумерованных продукций понимаются как *стратегии*. Атрибуты задают набор существенных (с точки зрения организации процесса проектирования) свойств проектируемых объектов, технологических ограничений и целей проектирования. Совокупность задействованных в системе атрибутов определяет пространство признаков, необходимых для принятия проектных решений. Атрибуты представляются полным названием, сокращенным названием, типом и текущим значением. Полное название атрибута отражает его целевое назначение. Сокращенное название используется при записи определенных в системе продукций. Атрибуты могут быть трех типов: логический (*bool*), числовой (*integer*) и символьный (*text*). Атрибуты могут быть классифицированы как атрибуты описаний объектов проектирования и базисов проектирования, атрибуты программных модулей, вспомогательные атрибуты для управления выполнением продукций. Атрибуты группируются по слотам, которые, в свою очередь, включаются во фреймы. В результате такого группирования признаковое пространство предлагается описывать четырьмя фреймами: *Блок проекта* (описание иерархически организованных данных), *Параметры* (описание базисов проектирования и параметров), *System* (пути к директориям и имена системных файлов), *Work* (атрибуты для управления стратегиями).

4. Использование модели представления знаний в САПР

На основе предложенной модели представления знаний в области логического проектирования реализованы две программные системы *ESS* и *FLC*. Система *ESS* (*Expert SubSystem*) – предназначена для подготовки и ведения базы знаний интеллектуальной САПР и может быть использована как в области логического проектирования цифровых схем, так и в других проблемных областях и является достаточно универсальным инструментальным комплексом ведения базы знаний (БЗ). В качестве объектов, хранящихся в БЗ, выступают описания: атрибутов, задающих существенные характеристики объектов и условий проведения проектирования; программных модулей, реализующих отдельные проектные операции и некоторые другие действия, связанные с организацией процесса проектирования; динамического меню, для которой разрабатывается БЗ; стратегий проектирования в виде наборов продукций, связывающих текущее

состояние процесса, задаваемое значениями атрибутов, с его дальнейшим развитием в виде исполнения тех или иных проектных действий; фреймов, состоящих из кортежей слотов, задающих группирование атрибутов, используемых для определения объектов проектирования, его целей, истории, технологических ограничений и т. д. Еще одним из объектов БЗ является динамическое меню как отдельный фрагмент меню САПР, использующей данную БЗ. Для описания меню используется структура данных в виде дерева, каждая из вершин которого связана с позицией меню, причем в роли листовой вершины может выступать либо программный модуль, либо стратегия. Продукция, размещаемая в БЗ, определяется общей схемой

«ЕСЛИ <условие>=TRUE, то выполнить
<действие>»

и описывается при помощи множества вычисляемых предикатов и логической формулой условия, а также типом и именем объекта, указанного в правой части продукции и задающего действие, выполняемое системой.

Вторая система *FLC* (*Functional Logic Circuits*), предназначена для автоматизации логического проектирования заказных цифровых СБИС и обеспечивает возможность функционально-структурного описания объектов проектирования на языке высокого уровня; иерархического описания проектов функциональных блоков СБИС; использования набора специализированных проектных процедур на основе средств интеллектуальной поддержки принятия проектных решений; логической верификации проекта после выполнения любой из проектных процедур.

Оптимизация описаний проектов осуществляется на основе применения методов, реализующих различные базовые оптимизационные приемы – минимизацию функций в классе ДНФ, оптимизацию многоуровневых представлений систем функций на основе декомпозиции и разложения Шеннона и т. д. В системе *FLC* реализован производственный подход для управления последовательностями проектных процедур, предназначенных для их использования на этапе логического проектирования сложных функциональных блоков заказных цифровых СБИС.

Экспертными знаниями в системе *FLC* являются последовательности продукций (стратегии), вместе с правилами их компьютерной реализации.

В рамках блока «Оболочка САПР» системы *FLC* сосредоточены все возможности по управлению протекающими в САПР процессами. Блок «Интерфейс пользователя» представляет собой часть оболочки, связанную с организацией человеко-машинного интерфейса данной программной системы.

Проект представляется древовидной структурой, вершины которой соответствуют объектам

проектирования. Собственно процесс проектирования заключается в выборе текущего объекта и проведении над ним некоторого преобразования. Набор возможных преобразований определяется набором включенных в систему программных модулей, реализующих проектные операции. Изменение текущей проектной ситуации, описываемой при помощи атрибутов, связано либо с выбором нового текущего объекта, либо с реализацией выбранной проектной операции над текущим объектом. Каждый проект содержит описание схемы, представленной в виде иерархии объектов проектирования. Это внешнее представление в терминах, которыми оперирует проектировщик в процессе работы. Вместе с тем существует еще и внутреннее представление проекта, отдельные компоненты которого демонстрируются на экране во время сеанса проектирования. В частности, к ним относятся:

- дерево подчиненности объектов проектирования (подсхем), один из которых выделяется цветом и является текущим объектом, над которым будет реализовываться очередная проектная операция;

- множество характеристик (атрибутов) текущего объекта проектирования, существенных с точки зрения проводимого процесса;

- набор параметров текущего объекта проектирования, например множество имен входов и выходов, формат представления и др.;

- описание связей, представленное в текущем объекте, либо его функциональное описание.

Вместе с описанием схемы в проекте представлена некоторая дополнительная информация о текущей конфигурации системы и истории выполненных ранее шагов проектирования. Каждому шагу проектирования соответствует отдельное состояние проекта, автоматически запоминаемое системой в хранилище проектов. Таким способом обеспечивается поддержка сохранения истории проектирования.

Блок «Конвертеры форматов» сосредоточивает в себе средства преобразования способа описания объектов проектирования с целью обеспечения совместимости с внешними программными средствами из той же проблемной области, например системой синтеза *LeonardoSpectrum*, системой моделирования *ModelSim* [Перельройзен, 2004]. С точки зрения главного процесса – проектирования цифровых схем – основным блоком САПР является блок «Проектные операции». Следует особо отметить, что все подчиненные блоки системы *FLC*: «Интерфейс пользователя», «Конвертеры форматов» и «Проектные операции» – не являются жестко определенными и могут менять как свой состав, так и внутреннюю организацию.

ЗАКЛЮЧЕНИЕ

Предложенная модель представления знаний позволяет разрабатывать сложные

комбинированные маршруты оптимизации, синтеза, и верификации логических схем, записываемые в виде иерархически организованных стратегий. Экспериментальные исследования подтвердили высокую эффективность совместного использования *FLC* и промышленных систем проектирования цифровых схем. Одним из направлений дальнейшего развития системы *FLC* является разработка подсистемы накопления статистических данных и средств соответствующей статистической обработки для проведения экспериментов с целью выбора лучших маршрутов проектирования и реализации их в промышленных САПР.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Евгенов, 2009] Евгенов, Г.Б. Интеллектуальные системы проектирования: учеб. пособие / Г.Б. Евгенов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2009. – 334 с.

[Закревский и др., 2007] Закревский, А.Д. Логические основы проектирования дискретных устройств. /А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова //М.: Физматлит, 2007. – 589 с.

[Кузнецов, 1989] Кузнецов В.Е. Представление в ЭВМ неформальных процедур: производственные системы //М.: Наука. Гл. ред. физ.-мат. лит, 1989. – 160 с.

[Минский, 1979] Минский, М. Фреймы и представления знаний. /М. Минский //М.: Энергия, 1979. – 151 с.

[Паркер, 1987] Паркер, Э.С. Использование экспертных систем и кремниевой компиляции для автоматизации процесса проектирования СБИС /Э.С. Паркер, С. Хайяти //ТИИЭР. – 1987, Т. 75, № 6, С. 43–45.

[Перельройзен, 2004] Перельройзен, Е.З. Проектируем на VHDL /Е.З. Перельройзен //М.: СОЛОН-Пресс, 2004. – 448 с.

[Рыбина, 2010] Рыбина, Г.В. Основы построения интеллектуальных систем: учеб. пособие /Г.В. Рыбина //М.: Финансы и статистика; ИНФРА-М, 2010. – 432 с.

[Darringer, 1984] Darringer, J.A. LSS: A System for Production Logic Synthesis /J.A. Darringer //IBM J. Res. and Developm. – 1984, V. 28, № 5, P. 537–545.

[Sasao, 1996] Sasao, T. FPGA design by generalized functional decomposition /T. Sasao // Representations of discrete functions / ed. by Sasao T., Fujita M. – Norwell: Kluwer Academic Publishers, 1996. – P. 233–258.

[Uehara, 1985] Uehara, T. A knowledge-based logic design system /T. Uehara //IEEE Design and Test Comput. – 1985, V. 2, № 5, P. 27–34.

APPLICATION OF THE PRODUCTION RULE MODEL FOR KNOWLEDGE REPRESENTATION IN AUTOMATION LOGICAL DESIGN OF CUSTOM DIGITAL VLSI

Bibilo P.N., Romanov V.I.

United Institute of Informatics Problems
of NAS of Belarus, Minsk, Belarus

bibilo@newman.bas-net.by

rom@newman.bas-net.by

Application of the production rule system for knowledge representation in logical design system of complex functional blocks of custom digital VLSI is described. We focus to solve tasks of optimization of is functional-structural descriptions, synthesis and verification of hierarchically organized projects. The combined routes of design are considered in the form of strategies which represented as the set of products.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.896

ПРИМЕНЕНИЕ RAM-BASED СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ СТАЙНЫХ РОБОТОВ

Дёмин В.В.* , Кабыш А.С*., Дунец И.П.* , Дунец А.П.* , Головки В.А.*

** Брестский государственный технический университет,
г. Брест, Республика Беларусь*

spas@gmail.com

anton.kabush@gmail.com

ipdunets@bstu.by

dunets@gmail.com

gva@bstu.by

В данной работе рассмотрена задача распознавания графической метки ведущего робота в системе ведущий-ведомый роботы. Для решения задачи был разработан метод детектирования на основе RAM-based сетей позволяющий по расположению метки узнать положение и дальность ведущего робота по графическому паттерну. Обученная RAM-based сеть хранит характеристические особенности паттерна в разных секторах относительно ведомого робота. Сработавший дискриминатор сети будет указывать на сектор, в котором находится ведущий робот.

Ключевые слова: RAM-based сети; стайные роботы, распознавание образов, выбор порога.

ВВЕДЕНИЕ

В задаче следования за лидерами от роботов требуется сформировать паттерн формации, при котором каждый предыдущий робот следует за последующим, а ведущий – лидер, либо управляется оператором, либо следует по заранее заданному пути.

Существуют различные подходы к распознаванию роботами друг друга: инфракрасные датчики, световые паттерны, RFID-метки, компьютерное зрение и т.д. В данной работе распознавание ведущего основано на детектировании визуальной графической метки закрепленной на корпусе робота при помощи RAM-based нейронных сетей.

Роботы, оснащенные видеосистемой, могут получить необходимый минимум информации о находящихся поблизости роботах группы, если роботов достаточно легко детектировать по их особенностям, паттернам или световым меткам. Цель работы состоит в разработке методики распознавания ведущего робота, не требующей достаточных вычислительных мощностей, и устойчивой к помехам во внешней среде.

Отсутствие камеры в целом ограничивает область применения swarm роботов. Большинство камер на роботах swarm-масштаба не являются пригодными для глобального восприятия окружающей среды, а используются только для локальных областей, с максимальным радиусом до 1 метра.

В работе [Caprari, 2000] описано стайное поведение группы роботов окрашенных в красный цвет. Эта их характеристика (красный окрас) была использована при детектировании роботов друг другом с помощью с камеры. Если область не менее чем 25 на 25 пикселей по границам имела красный цвет, то область помечалась как вероятный сосед. После фазы распознавания принимались решения о факте наличия робота и дистанции до него.

В работе [Haverinen, 2005], роботы «s-bot» имели светодиодное кольцо по периметру робота формирующее треугольный паттерн из разных цветов, где красный цвет указывает направление движения робота. На роботе установлена всенаправленная камера, с эффективной областью обзора в 60 см. Алгоритм обработки изображения на роботе определяет по «цвету» направление движения соседей в области видимости камеры. Для этого используется вероятностный алгоритм определения направления движения соседей по

изображению, при заранее заданных оценках распределения. Роботы в данном исследовании решали задачу кооперативного транспорта путем создания формации и определяли направление движения путем переговоров.

1. Описание используемых роботов

Ведущий или лидер (leader)- автономный робот, основной задачей которого является движение к цели по некоторому маршруту и, возможно, без столкновений с препятствиями. За ведущим роботом следуют ведомые (followers) роботы, оборудованные видеокамерой для распознавания, ведущего робота. В задаче следования за лидером от ведомых роботов требуется не потерять ведущего – лидера, который либо управляется оператором, либо следует по заранее заданному пути[Gupta, 2009].

В качестве мобильной платформы используется DFRobot Turtle 2WD Mobile platform. Ведущий робот оснащен паттерном на задней части (рис. 2). Размер паттерна 28 на 7 см. Паттерн состоит из трех горизонтальных полос: черной, белой, черной. Высота черных полос 2,5 см, белой – 2 см.

Горизонтальное расположение линий паттерна обусловлены требованием горизонтальной схожести паттерна, т.к. детектирование производится на основе положения робота, а не его ориентации. Порядок черная-белая-черная линий выбран для детектирования промежуточных положений робота.

Ведущий и ведомый робот в связке во время проведения экспериментов изображены на рисунке 1.

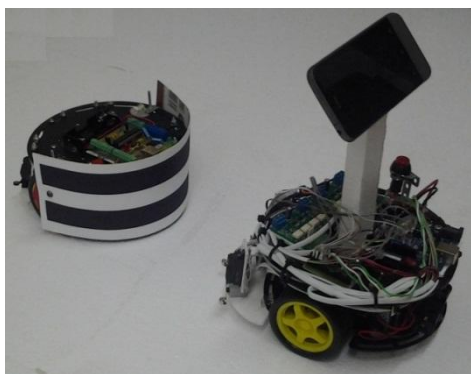


Рисунок 1 – Ведущий и ведомый роботы

2. Система детектирования ведущего робота

Система детектирования ведущего робота основана на использовании RAM-based сетей с предварительными фазами подготовки и обработки изображения (рисунок 2).

Для распознавания области, в которой находится ведущий робот, изображение подвергается обработке с целью выделения графической метки. После предварительных фаз, изображение делится на 6 секторов, в которых выполняется поиск графической метки RAM-based сетью.

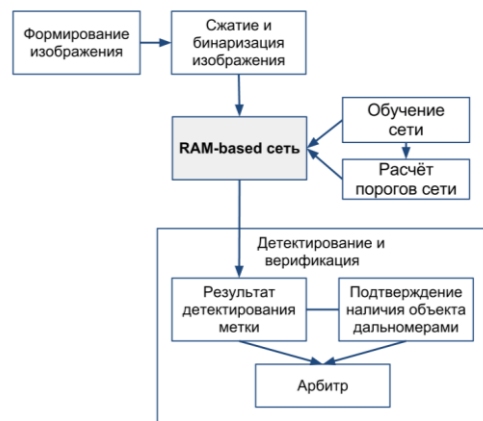


Рисунок 2 – Компоненты системы детектирования графической метки

На основе детектирования арбитр принимает решение о наличии или отсутствии графической метки в заданном секторе. Опционально, может использоваться верификация при помощи инфракрасных дальномеров. На основе результатов распознавания робот принимает решение о дальнейших действиях, согласно алгоритму поведения.

2.1. Подготовка изображения

Расположение и ориентация камеры ведомого позволяет получить область обзора камеры для детектирования от 15 до 70 см. Робот получает цветное изображение с камеры в виде массива размером 320 на 240 пикселей, где каждый пиксель представлен тремя байтами в цветовом пространстве RGB. Полученное изображение конвертируется в цветовое пространство Grayscale, содержащее изображение в градациях серого.

Последующая обработка заключается в обрезке малоинформативных областей изображения и разбиение полученного изображения на сектора(рисунок 3а, 3б).

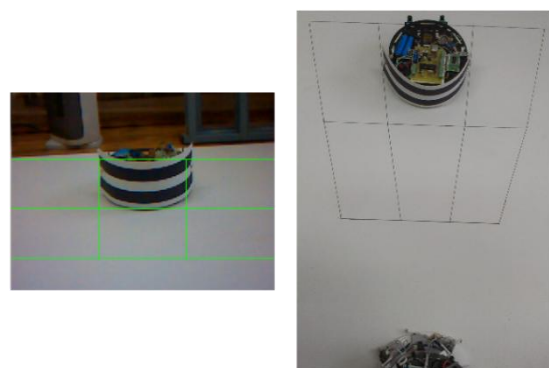


Рисунок 3 – (а) Рабочая область изображения и (б) пространства перед роботом.

Обрезка малоинформативных областей изображения заключается в отсечении 40% верхней части изображения и 10% нижней, которые подобраны экспериментально.

Полученное изображение имеющее размер 320 на 120 пикселей сжимается при помощи линейной интерполяции до размера 80 на 60 пикселей. Затем бинаризуется с помощью адаптивной пороговой бинаризации [Shafaita, 2008] (рисунок 4).



Рисунок 4 – Сжатие и бинаризация изображения

3. RAM-based сети

Низкие вычислительные мощности микроконтроллеров, используемых в мобильной робототехнике, накладывают ограничения на возможности обработки видео данных в реальном режиме. В данной работе представлена архитектура RAM-based нейронной сети, которая является видом ассоциативной памяти с произвольным доступом. Сеть имеет ряд применений для распознавания образов; например для распознавания лиц и букв. В работе [Yao, 2003] реализована система объезда препятствий на базе микроконтроллера 8051 с 512 байтами памяти. Робот обрабатывал данные восьми инфракрасных датчиков и успешно решал поставленную задачу избегания препятствий в режиме реального времени. Отличительной особенностью сети является простота обучения, тестирования, а так же низкие требования к памяти и вычислительным мощностям.

В основе работы сети лежит теория n – кортежей. Кортеж – это блок изображения определенного размера. Бинаризованное изображение шириной I_w и высотой I_h пикселей разбивается на кортежи, которые подаются на вход сети. Введем обозначение кортежа как T , а размер кортежа T_s . Тогда количество кортежей T_c изображения согласно формуле (1).

$$T_c = \frac{I_w * I_h}{T_s}. \quad (1)$$

Изображение, хранимое в памяти сети, называется дискриминатором [Yao, 2003]. Одно RAM-based нейронная сеть может хранить k дискриминаторов. Один дискриминатор состоит из T_c блоков памяти произвольного доступа (RAM), каждый размером T_s . В каждом из блоков хранится образ для сравнения с кортежем, получаемым из изображения, подаваемого на вход сети. Если поступающий образ совпадает с образом RAM – на выходе выдается 1, иначе 0.

Количество совпадений RAM с образами на i -м дискриминаторе, рассчитывается по формуле (2).

$$D_i = \sum_{j=1}^{T_c} (T_j \text{ and } RAM_i^j). \quad (2)$$

где T_j – j -й кортеж подаваемого на сеть изображения, – j -й блок памяти i -го дискриминатора.

Значение на выходе D_i сети является показателем соответствия входного изображения обученному образу. В частном случае, D_i – это количество совпавших кортежей образа и дискриминатора. Сеть не имеет весовых коэффициентов ассоциированных со связями.

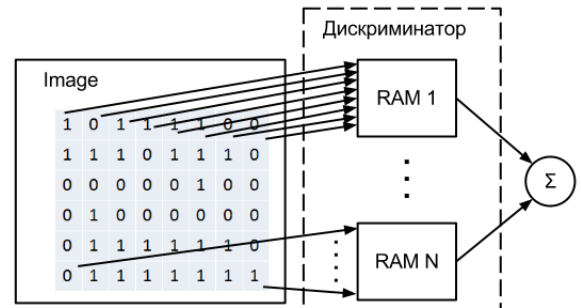


Рисунок 5 – Дискриминатор RAM-based сети, где $N=T_s$

Обучение сети производится методом прямого распространения сигнала. Начальные значения образов дискриминаторов равны нулю. Изображение разбивается на T_c кортежей, размером T_s каждый. Кортежи поступают на входной слой дискриминатора и сохраняются в постоянную память.

Для каждого хранимого образа сети, как правило, требуется меньше памяти (в зависимости от выбранного размера кортежа), чем размер используемого для распознавания. Размер памяти Z , необходимый для одной ячейки памяти RAM, согласно формуле (3).

$$Z = 2^{T_s}. \quad (3)$$

Размер памяти M необходимый для хранения изображения сети:

$$M = T_c * Z. \quad (4)$$

3.1. Архитектура RAM-based сети для детектирования паттерна

Архитектура сети применяемой для классификации изображения с целью определения сектора, в котором находится ведущий робот, приведена на рисунке 7. Сеть имеет $d=6$ дискриминаторов, каждому из которых соответствует свой сектор, в котором может находиться ведущий робот.

Распознавание метки заключается в определении класса дискриминатора соответствующего текущему изображению. Подаваемое на вход сети изображение подвергается обработке, разбивается на кортежи, сворачивается в дискриминатор и последовательно сопоставляется с каждым дискриминатором сети. Предварительно, сеть обучается, настраивая каждый дискриминатор для определения положения робота в отдельном секторе

изображения. Образ последовательно сравнивается с каждым из дискриминаторов, для каждого из которых рассчитывается количество срабатываний совпадений D_i по (6).

$$D_i = D_i^{\text{черные}} + D_i^{\text{остальные}}. \quad (5)$$

Где, $D_i^{\text{черные}}$ и $D_i^{\text{остальные}}$ - это количество совпадений по чёрному и остальным кортежам соответственно, рассчитываемые по формуле (3) с учётом цвета кортежа.

Выходы дискриминаторов D_i подаются на вход классификатора, который оценивает значение D_i относительно функции порога дискриминатора $T_i(D_i)$. Все дискриминаторы, количество срабатываний которых попадает в диапазон порога, становятся кандидатами в победители.

Итоговый дискриминатор победитель определяется по максимуму количества срабатываний. Выходом классификатора является номер дискриминатора-победителя, которому соответствует сектор, в котором находится ведущий робот.



Рисунок 6 – Архитектура RAM-сети для определения сектора.

Описанный подход распознавания ведущего робота на основе RAM-based нейронной сети позволяет определить ориентировочно расстояние: близко или далеко и направление лево-центр-право. Расстояние близко означает, что ведущий может располагаться на расстоянии от 25 до 45 см, далеко – от 45 до 70 см.

4. Обучение RAM-based сети

4.1. Обучение Ram-based сети

Обучение сети заключается в создании дискриминаторов для каждого из секторов. Размер сектора равен усредненному размеру паттерна ведущего робота. Количество секторов, где может находиться робот, определяется относительными размерами паттерна и рабочей областью изображения. В данной работе один дискриминатор содержит один образ паттерна. Обучающие образы подаются на вход сети, которая запоминает их в соответствующих дискриминаторах. Изображения, используемые в качестве основы для обучающих образов, приведены на рисунке 7. На рисунке 8 показаны обучающие образы после предобработки. Обучающие образы отражают эталонное положение ведущего робота относительно ведомого.

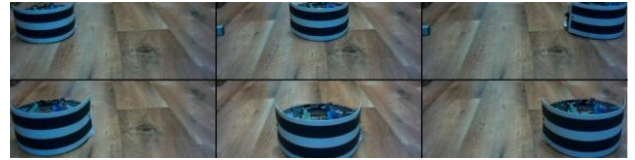


Рисунок 7 - Паттерны для обучения сети



Рисунок 8 – Паттерны после предобработки

Следовательно, если это же изображение будет подано на вход сети, количество совпадений на i -м дискриминаторе будет максимальным. Поэтому, во время обучения на дискриминаторе запоминается $D_i^{\text{чернmax}}$ - количество совпадений по чёрному цвету, и $D_i^{\text{остальнымmax}}$ - количество срабатываний по остальным. Конкретные значения этих величин зависят от размера кортежа T_s .

4.2. Расчет порога Ram-based сети

Предыдущие работы авторов [Yao, 2003] основывают выбор дискриминатора-победителя по количеству совпадений образов RAM. Простое определение сектора положения лидера на основе дискриминатора победителя в данной работе не представляется возможным из-за неравнозначности размера графической метки робота в дальних и ближних секторах. Ближние сектора содержат паттерн большего размера, и количество срабатываний в ближних секторах будет выше по фону. Следовательно, требуются разные пороги для разных секторов.

Для устранения сбоев введены пороги для каждого из дискриминаторов сети. Значение порогов получены исходя из средневзвешенной суммы результатов срабатывания дискриминатора соответствующего класса (формула 5).

Допустим, на вход сети подается произвольное изображение с роботом в позиции, попадающей в один из секторов. Если изображение близко к эталонному, то количество срабатываний будет стремиться к соответствующим значениям $D_i^{\text{чернmax}}$ и $D_i^{\text{остальнымmax}}$.

Если положение робота отличается, то количество срабатываний будет соответственно уменьшаться. Суть задания пороговой функции в том, что бы определить, при каких значениях $D_i^{\text{черные}}$ и $D_i^{\text{остальные}}$. Метка считается распознанной, если значение количества срабатываний соответствующего дискриминатора попало в допустимый порог.

Для определения порога дискриминатора использовалась выборка положений робота по

секторам. Ведущий робот $P_i=20$ раз ставился в каждый сектор.

По каждому изображению, для каждого из дискриминаторов высчитывалось срабатываний по полностью чёрным паттернам, и по всем остальным. Нижний порог i -го дискриминатора для чёрных и белых совпадений определяется, метом k -минимальных ближайших соседей.

$$T_i^{c\min} = \frac{1}{k} \sum_{t=1}^k D_i^c(t). \quad (6)$$

где, $T_i^{c\min}$ - это нижняя граница порога дискриминатора для цвета $c \in$ (чёрный, остальные). k - количество нижних соседей участвующих в расчете нижней границы порога.

Пороговая функция определяется следующим образом:

$$F_i(D_i) = \begin{cases} 1, & \text{если } T_i^{c\min} \leq D_i^c \leq T_i^{c\max} \\ 0, & \text{иначе} \end{cases} \quad (7)$$

После определения порогов сеть считается рабочей и готова к детектированию. В таблице 1 отображены внешние параметры сети, используемые в данной работе при настройке и обучении сети.

Таблица 1 - Внешние параметры сети

Параметр	Обозначение	Значение
Размер изображений.	ширина I_w высотой I_h	Подбирается эмпирически
Количество обученных дискриминаторов в сети.	D	6
Размер кортежа.	T_s	Подбирается эмпирически
Размер дискриминатора	T_c	Рассчитывается по (2)
Порог бинаризации.	T_b	Подбирается эмпирически
Нижняя граница дискриминатора D_i^c по цвету c .	$T_i^{c\min}$	Обучение, формула (7)
Верхняя граница дискриминатора D_i^c по цвету c .	$T_i^{c\max}$	
Количество значений в обучающей выборке для расчета порога.	K	Определяется эмпирически $0 < i \leq P_i$

5. Детектирование

Проведем M экспериментов детектирования с истинными и ложными целями. Целью экспериментов ставится определение качества детектирования при рассчитанных значениях порогов при заданном размере кортежа, а так же определение оптимального размера кортежа для детектирования метки.

Будем обозначать количество образов с ведущим роботом как P (от *positive*), и количество образов с ложными целями как N (от *negative*). Тестовая выборка состоит $M=60$ образов, случайно подаваемых на вход сети. Количество образов с ведущим роботом $P=30$, где на каждом образе ведущий робот случайным образом расположен в некотором секторе. Ложные образы делятся на 3 группы по 10 штук ($N=30$), где на каждом из образов в случайной позиции находится ложный предмет прямоугольной, квадратной или круглой формы.

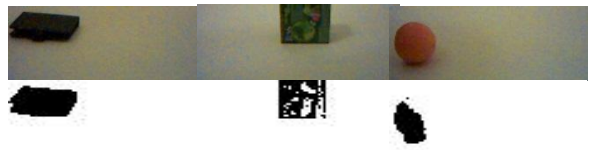


Рисунок 9 – Оригинальные и бинаризованные примеры ложных образов разных категорий.

Целью разработки алгоритма детектирования паттерна робота является получение классификатора с наибольшей вероятностью обнаружения атак TPR (*true positive rate*) при приемлемом уровне ложных срабатываний FPR (*false positive rate*). Результаты тестирования сведены в таблицу 2 и 3.

Таблица 2 - Таблица оценок срабатываний для распознавания графической метки робота в зависимости от размера кортежа.

Выход сети	Эталонный выход сети	Размер кортежа				
		2	4	6	8	10
True	Positive	19	21	21	29	30
True	Negative	15	17	13	2	0
False	Positive	15	13	17	28	30
False	Negative	11	9	9	1	0

Таблица 3 - Характеристики и результаты детектирования

Описание	Размер кортежа,				
	2	4	6	8	10
ACC	0.57	0.63	0.56	0.52	0.5
TPR	1	0.7	0.7	0.97	1
FPR	0.5	0.43	0.56	0.93	1
FNR	0.37	0.3	0.3	0.33	0
TNR	0.5	0.57	0.43	0.67	0

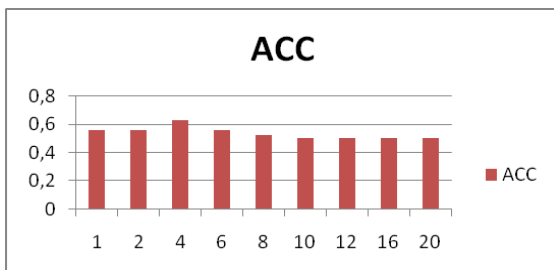


Рисунок 10 – точность классификации для всех тестируемых размеров кортежей

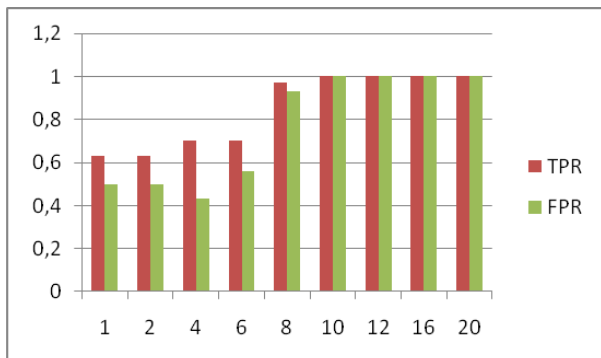


Рисунок 11 - Зависимость TPR/FNR от размера кортежа

Исходя из рассчитанных параметров, оптимальный размер кортежа для детектирования равен 4 (рисунок 10 и 11).

Заключение

В данной работе было показано применение RAM-based сетей для распознавания положения и дальности ведущего робота по графическому паттерну. Обученная RAM-based сеть хранит характеристические особенности паттерна в разных секторах относительно ведомого робота. Предобработанное изображение подается на вход RAM-based сети, которая по количеству совпадений в дискриминаторах определяет в каком секторе находится паттерн. Порог срабатывания дискриминаторов сети индивидуальным образом характеризует необходимую степень схожести для распознавания паттерна в данном секторе.

Основным преимуществом применения RAM-based сетей для распознавания графических меток является снижение вычислительных ресурсов, что позволяет применять данное решение на платформах с низкой производительностью. Эффективность работы системы обусловлена предобработкой изображения, выбором размера кортежа. Основные отличия данного подхода от других, это малые вычислительные ресурсы, простота обучения сети и простота тестирования сети.

Библиографический список

- [Caprari, 2000] G. Caprari, K. O. Arras и R. Siegwart, «The autonomous miniature robot alice: from prototypes to applications» в In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00) - 2000.
- [Haverinen, 2005] J. Haverinen, M. Parpala и J. Rning, «A miniature mobile robot with a color stereo camera system for swarm

robotics research.» в IEEE International Conference on Robotics and Automation (ICRA2005), Barcelona, Spain - 2005.

[Yao, 2003] Q. Yao, D. Beetner, D. Wunsch II и B. Osterloh, «A RAM-Based Neural Network for Collision Avoidance in a Mobile Robot.» Neural Networks, Proc. of IEEE International Joint conference – 2003 , pp. 3157-3160 vol. 4.

[Gupta, 2009] M. Gupta, J. Das, M. Vieira, H. Heidarsson, H. Vathsangam и G. Sukhatme, «Collective Transport of Robots: Emergent Flocking from Minimalist Multi-robot Leader-following» в IEEE/RSJ International Conference on Intelligent Robots and Systems - 2009

[Shafaita, 2008] F. Shafaita и др., Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images // Document Recognition and Retrieval XV San Jose, CA | January 27, 2008

APPLICATION RAM-BASED NETWORK FOR SWARM ROBOTICS DETECTION

Demin V.V.*, Kabush A.S.*, Dunets I.P.*,
Dunets A.P.*, Golovko V.A*.

*Brest State Technical University, Brest, Republic of Belarus

spas@gmail.com

anton.kabush@gmail.com

ipdunets@bstu.by

dunets@gmail.com

gva@bstu.by

In this paper we consider the problem of detection the graphic mark of leading robot in a leader following task. The detection method based on RAM-based is purposed. Trained RAM-based network keeps the features of a visual mark in different sectors regarding the slave robot. The network winner-discriminator will indicate the sector in which the leading robot present.

INTRODUCTION

In the task of following the leaders of the robot is required to form a pattern formation, in which each robot follows the previous followed, and the leader - a leader, or controlled by the operator or follows a predetermined path.

There are different approaches to the recognition of each other's robots: infrared sensors, light patterns, RFID-tags, computer vision, etc. In this paper, the recognition is based on the detection of the leading visual graphic tags attached to the body using a robot RAM-based neural networks.

The robots are equipped with a video system can obtain the minimum information about a group of nearby robots if robots is easy to detect by their features, patterns or light marks. The purpose of this paper is to develop recognition techniques leading robot that does not require enough computing power and immunity to interference in the environment.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 681.513.6, 004.896

НЕЙРОСЕТЕВАЯ МОДЕЛЬ ДЛЯ РЕАЛИЗАЦИИ ПОИСКОВЫХ ДВИЖЕНИЙ МОБИЛЬНОГО РОБОТА

Прокопович Г.А.

*Объединённый институт проблем информатики НАН Беларуси,
г. Минск, Республика Беларусь*

rprakapovich@robotics.by

Предлагается модель нейросетевого процессора, разработанного на основе принципов функционирования гетероассоциативной памяти, который способен реализовать ветвящиеся алгоритмы. В качестве примера, приводится описание реализации алгоритма поиска пути в лабиринте. Предложенную модель можно рассматривать как нейросетевой компонент более общей нейросетевой иерархической системы управления, предложенной автором ранее.

Ключевые слова: гетероассоциативная память; система управления; поисковые движения; алгоритмы с ветвлением.

ВВЕДЕНИЕ

Среди основных тенденций в области разработок интеллектуальных систем управления робототехническими аппаратами выделяется проблема обеспечения их полной либо частичной (с привлечением человека в контур управления) автономности. Таким образом, типовой задачей для автономных мобильных роботов является ограниченный по времени и запасу энергии процесс поиска в неизвестной местности искомого объекта и возвращение в точку старта вместе с ним, либо с некоторой информацией о нем. Причём, как правило, искомый объект описан неполными или противоречивыми начальными данными.

Несмотря на свою сложность, описанная задача в естественной природе является типичной и имеет массу наглядных примеров своего успешного решения. Поэтому среди перспективных методов управления интеллектуальными системами можно выделить бионический. В данном подходе большое внимание уделяется изучению фундаментальных принципов функционирования естественных систем управления и механизмов их эволюции, а не отдельным аспектам его проявления [Жданов, 2009].

Большинство бионических систем управления в той или иной степени основаны по аналогии с управляющими системами живых организмов – их нервной системой [Самарин, 2002; Жданов, 2009]. Естественные системы управления реализуют принципы адаптивного распознавания входной информации и универсальные поисковые

алгоритмы для соответствующего реагирования. Причём в качестве практической реализации подобных систем чаще всего выступают различные модели искусственных нейронных сетей [Головкин, 2001].

Искусственные нейронные сети представляют собой распределённые и параллельные вычислительные системы, способные к адаптивному обучению путём анализа положительных и отрицательных воздействий [Уоссермен, 1990]. Применительно к робототехнике они дают возможность замены традиционного программирования на процесс обучения с помощью примеров, что позволяет использовать режим постоянного дообучения при работе в изменяющихся средах [Амосов и др., 1991]

1. Постановка задачи

Как правило, в бионических методах аналитическая модель объекта управления априори неизвестна. Необходимые знания добываются эмпирически, в процессе взаимодействия со средой и самим объектом управления [Гаазе-Рапопорт и др., 1987; Самарин, 2002; Жданов, 2009]. Таким образом, приспособление системы управления к среде функционирования достигается путём последовательного извлечения информации об особенностях данной среды, что выступает как своеобразный аналог ее познания [Камшилов, 1979].

Известные модели искусственных нейронных сетей чаще всего используются в контурах

управления как отдельные управляющие элементы, обладающие некоторой необходимой переходной характеристикой, либо выполняют функции предобработки и классификации данных. Причём, такие нейронные сети не способны полноценно реализовать систему управления. Однако это противоречит биологическим аналогам, которые представляют собой набор взаимодействующих однородных нейронных структур.

В связи с этим, целью данной работы является продолжение исследований автора по развитию нейросетевой модели иерархической системы управления с распределённой обработкой информации [Прокопович, 2010а]. Указанная модель представляет собой набор семантически связанных функциональных компонентов, состоящих из гетероассоциативных нейроподобных сетей. Тогда задача будет заключаться в попытке разработать новый нейросетевой компонент, предназначенный для реализации поисковых движений мобильного робота.

2. Нейросетевая модель иерархической системы управления

Одной из главных особенностей нервной системы живых организмов, позволяющей им приспосабливаться к изменениям среды, является выработка новых форм реагирования на внешние и внутренние воздействия [Крылов, 2007]. Такие формы поведения, основная цель которых заключается в приобретении новой информации, называются обучением [Гаазе-Рапопорт и др., 1987]. Эту способность можно определить как совокупность процессов, обеспечивающих выработку и закрепление форм реагирования, адекватных физиологическим, биологическим и социальным потребностям организма.

Как правило, выделяют три различные группы способов организации поведения живых организмов: реактивное, оперантное и когнитивное [Крылов, 2007]. Под реактивным поведением подразумевается пассивное взаимодействие, когда организм на длительный или короткий промежуток времени запоминает свои индивидуальные реакции на определённые стимулы, при котором происходит трансформация нейронных цепей и формирование новых следов памяти. Среди разновидностей реактивного поведения обычно выделяют безусловные и условные рефлексy.

Оперантное (экстремальное) поведение представляет собой закрепление таких действий, последствия которых для организма желательны, и отказ от тех действий, которые приводят к нежелательным последствиям. Для реализации новых типов поведения выделяют три метода обучения: проб и ошибок, формирование автоматизированных реакций (последовательности простейших действий) и подражание [Гаазе-Рапопорт и др., 1987].

Когнитивное поведение в эволюционном отношении является наиболее поздним и наиболее эффективным типом обучения. В полном объёме такое поведение присуще только людям, хотя какие-то его эволюционные предшественники или отдельные элементы наблюдаются и у высших животных [Симонов, 1981; Гаазе-Рапопорт и др., 1987; Крылов, 2007]. Если такое поведение присуще искусственным системам, то оно называется интеллектуальным.

Опираясь на методологию общей теории поведения естественных и искусственных систем [Гаазе-Рапопорт и др., 1987] и метод автономного адаптивного управления [Жданов, 2009] автором данной работы была предложена модель (рисунок 1) нейросетевой иерархической системы управления с распределённой обработкой информации [Прокопович, 2010а] для управления робототехническими аппаратами, в которой реализуются три контура управления:

1. $P \rightarrow \text{ФРО} \rightarrow \text{РА} \rightarrow \text{ПР} \rightarrow \text{Э}$ – реактивный,
2. $P \rightarrow \text{ФРО} \rightarrow \text{БЗ} \leftrightarrow \text{ПР} \rightarrow \text{Э}$ – экстремальный,
3. $P \rightarrow \text{ФРО} \rightarrow \text{БЗ} \leftrightarrow \text{ЭМ} \rightarrow \text{ПР} \rightarrow \text{Э}$ – когнитивный,

где P – рецепторы, ФРО – блок формирования и распознавания образов, РА – рефлексивный аппарат, БЗ – база знаний, ЭМ – блок эмоций и мотиваций, ПР – блок принятия решений и Э – эффекторы.

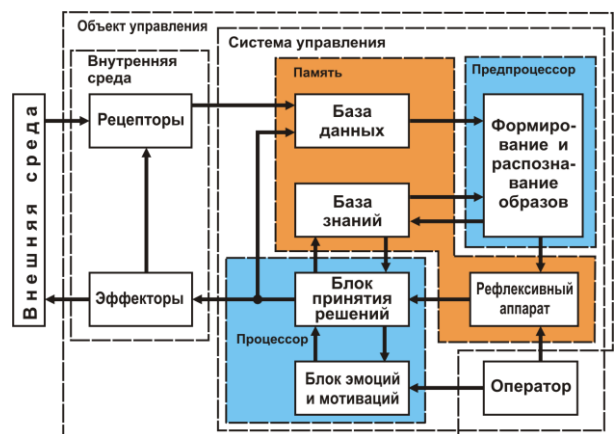


Рисунок 1 – Модель нейросетевой системы управления

В предложенной модели системы управления реактивное поведение предназначено для контроля за ответственными параметрами всего робота, которые не должны выходить за критические значения. К ним относятся напряжение и сила тока в двигателях, максимально допустимая скорость движения, а также расстояния до внешних преград. Поисковое, или экстремальное, управление предназначено для минимизации целевых функций робота путём случайных или направленных опытом действий. Причём в системе управления должны запоминаться только наиболее удачные попытки. На верхнем уровне управления находится интеллектуальный (когнитивный) контур, предназначенный для прогнозирования изменений ситуации на несколько шагов вперёд и принятия соответствующих управляющих решений.

Указанную систему управления можно рассматривать как универсальную масштабируемую нейросетевую вычислительную среду, состоящую из однотипных многофункциональных блоков, выполненных на базе гетероассоциативных нейронных сетей типа «двунаправленная ассоциативная память» [Kosko, 1988; Уоссермен, 1990; Прокопович, 2009]. Безусловные и условные рефлексы, задействованные в первом контуре управления, реализованы в блоке РА в виде двух различных нейроподобных сетей, которые при поступлении определённого стимула вызывают строго фиксированную реакцию [Прокопович, 2010b] или последовательность реакций [Прокопович, 2012b].

Контур экстремального управления предназначен для реализации более сложного типа поведения робототехнического аппарата, система управления которого способна не только распознавать ранее обученные ситуации и реагировать на них соответствующими действиями эффекторов, но также идентифицировать и запоминать новые. Наиболее ответственным звеном в данном контуре является БЗ, реализованная на основе системы классификаторов, которые можно рассматривать как специальный случай теории обучения с подкреплением [Джонс, 2004]. В качестве прототипа в работе [Прокопович, 2010b] был взят адаптивный нейросетевой классификатор, который благодаря наличию гетероассоциативного механизма позволяет быстро извлекать из памяти эталонные образы и обрабатывать входную информацию.

Когнитивный контур управления предназначен для отображения закономерностей реального мира в виде неформального опыта с целью достижения целей, поставленных оператором и необходимых для самосохранения самого робота [Самарин, 2002; Прокопович, 2010b]. Когнитивный контур управления предназначен для решения таких ответственных задач, как декомпозиция целевых требований на подзадачи (функция мотивации), прогнозирование и оценка состояний внешней и внутренней сред (функция эмоциональной окраски) и разработка алгоритмов принятия решений (функции планирования).

Декомпозицию целевых требований можно рассматривать как процесс кластеризации набора элементарных действий, необходимых для достижения эффекторами конечных состояний, которые указаны в мотивациях. Как положительные, так и отрицательные эмоции, определяющие величину побуждения к достижению целей, возникают в процессе рассогласования между прогнозируемыми и действительными состояниями объекта управления [Симонов, 1981]. В отличие от двух предшествующих тактических контуров управления (реактивного и экстремального), третий является стратегическим. Поэтому, процесс проектирования всегда сопряжен с выбором

оптимального по определенным критериям решения из нескольких альтернативных.

Для решения всех трех задач когнитивного управления в БЗ требуется наличие уже готовых, апробированных образов-шаблонов, которые, по-видимому, должны иметь большую размерность. В связи с этим, указанные шаблоны предлагается записывать в БЗ путем обучения в виде ассоциативных последовательностей [Прокопович, 2012a; Прокопович, 2012b].

3. Реализация поискового алгоритма с помощью нейросетевого процессора

В работе [Прокопович и др., 2010] описывается алгоритм децентрализованного управления группой роботов, который заключается в выполнении единого набора рекомендаций по выбору направлений движения для каждого члена группы. Указанный алгоритм реализуется в каждом роботе в виде жёстко выполняемой компьютерной программы, для которой требуется микроконтроллер, либо микропроцессорное устройство. Однако реализация даже несложного алгоритма с помощью классических нейронных сетей на данный момент является затруднительным. Это связано с тем, что классические нейронные сети выполняют, в основном, функцию классификатора, либо аппроксиматора. В связи с этим, в данной работе предлагается реализовать описанный алгоритм управления групповыми роботами на основе предложенного выше компонентного подхода для построения нейросетевой системы управления.

В работе [Прокопович, 2011] описывается нейросетевая ассоциативная модель достижения цели, на основе которой в работе [Прокопович, 2012b] был предложен метод гетероассоциативного управления манипулятором промышленного робота. Указанная система управления, в отличие от известных аналогов, не содержит в явном виде массивы значений углов поворота каждого из звеньев, а последовательно извлекает их из блока гетероассоциативной памяти.

Однако с помощью предложенной модели достижения цели можно реализовать гораздо больше практических приложений. Так, например, в качестве информационных блоков-образов, запоминаемых во время обучения, можно представить список рекомендаций, которым должна придерживаться система управления мобильным роботом при прохождении некоторого помещения по заранее известной местности (т.е. имеется точная карта). Таким образом, информационными блоками могут служить характерные точки на карте: повороты, лестничные проёмы, коридоры, двери и т.д. Другими словами, их можно представить как результат декомпозиции рассматриваемого пути на промежуточные участки, характеризующие некоторые точки в пространстве.

В данной работе предлагается использовать

указанную нейросетевую модель достижения цели как имитационный процессор, реализующий выполнение некоторых не сильно ветвящихся алгоритмов. В качестве примера можно рассмотреть, так называемый, алгоритм «правой руки», который предназначен для прохождения заранее неизвестного лабиринта [MYROBOT.RU, 2007]. Суть данного алгоритма заключается в том, чтобы робот всё время двигался вдоль стены, находящейся по правый борт мобильного робота. Двигаясь вдоль стены, робот должен следить, есть ли проход справа. Если проход есть, то робот должен идти по нему, чтобы не оторваться от стены справа. Если прохода нет – впереди стена, то робот должен повернуть налево. Основная часть алгоритма, не учитывающая нахождение стены и завершение самого поиска (нахождение выхода), изображена на следующей блок-схеме (рисунок 2).

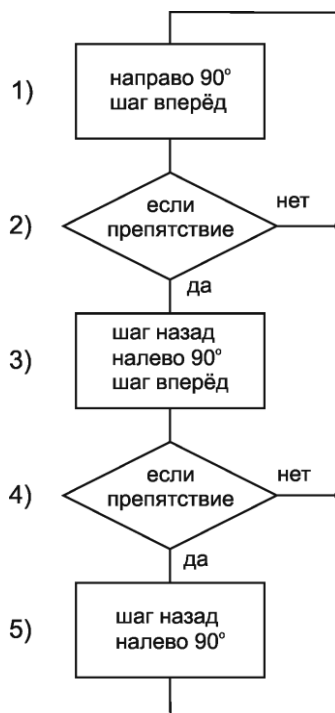


Рисунок 2 – Алгоритм движения по лабиринту по правилу «правой руки»

Анализируя приведённую выше блок-схему, можно заметить, что если логические блоки 2 и 4 будут выдавать положительные результаты, то вычислительные блоки 1, 3 и 5 (которые представляют собой наборы управляющих команд для моторов робота) будут непрерывно следовать друг за другом в строгой последовательности. Причём, после блока 5 обязательно должен следовать блок 1, т.е. должен получиться замкнутый цикл. Данный сценарий можно реализовать по примеру указанного метода управления манипулятором на основе модифицированной сети двунаправленная ассоциативная память [Прокопович, 2012b]. Однако если один из логических блоков 2 и 4 выдаст отрицательный результат, то полученная последовательность активизации блоков 1-3-5 будет нарушена. Таким образом, перед нами стала задача преобразовать

указанную модель достижения цели, чтобы она позволила реализовывать алгоритмы с ветвлениями.

В результате была получена модифицированная схема (рисунок 3) достижения цели, позволяющая реализовать приведённый выше алгоритм. Первое нововведение заключается в том, что информационные блоки должны содержать не только управляющие команды для приводов, но и значения сенсоров, соответствующие конкретным ситуациям. Второе нововведение заключается в использовании особого блока управления (БУ), включённого в контур обратной связи и выполняющего функцию коммутатора. Его задачей является селективное переключение информационных потоков. Если соответствующие участки извлечённого из ассоциативной памяти информационного блока совпадут с показаниями сенсоров, то логический элемент И выдаст положительный результат и блок управления соединит второй слой нейронной сети с первым через матрицу синаптических связей V , которая представляет собой единичную матрицу размером $N \times N$. При этом, i -й информационный блок извлечёт из гетероассоциативной памяти на втором слое нейронов $(i+1)$ -й блок.

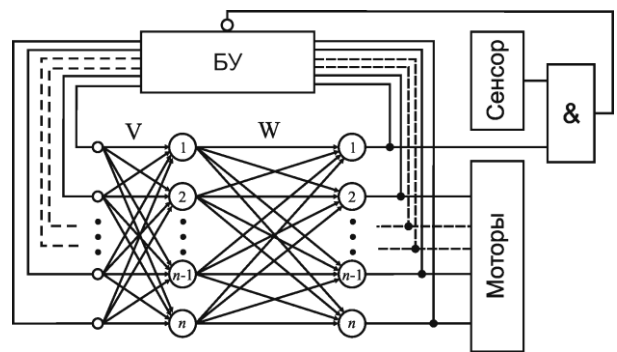


Рисунок 3 – Схема нейросетевого процессора

Если соответствующие участки вновь извлечённого информационного блока не совпадут со значениями текущих сенсорных данных, то блок управления принудительно подаст на первый слой нейронов значения k -й информационный блок, где k – число всех информационных блоков в гетероассоциативной памяти (причём, в нашем случае, $k=3$). В результате, на выходах второго слоя нейронов будет сгенерирован первый информационный блок и на входы блока моторов поступят соответствующие управляющие сигналы. Другими словами, будет реализован процесс альтернативного перехода управления от любого логического блока к 1-му вычислительному блоку (см. схему на рисунке 2). Таким образом, предложенная нейросетевая модель способна имитировать выполнение приведённого выше поискового алгоритма.

В связи с тем, что предложенный нейросетевой процессор базируется на основе модели гетероассоциативной памяти, подробно описанной в работе [Прокопович, 2012a], то на него накладываются соответствующие ограничения:

- длина всех информационных блоков должна быть одинаковой;
- информационные блоки не должны повторяться (уникальность образов);
- последний информационный блок должен быть ассоциирован с первым.

Благодаря первому ограничению первый и второй слои гетероассоциативной памяти должны иметь одинаковое число нейронов.

Покажем на конкретном примере, каким образом можно создать информационные блоки-образы. Так как робот имеет только один сенсор, установленный на переднем бампере, то его значения могут быть закодированы в бинарном виде только как 0 или 1. Поэтому для кодирования сенсорной информации достаточно одного бита. Однако, в связи с тем, что в 3-м вычислительном блоке имеется три действия, выполняемых роботом, а в 1-м и 5-м – только два, то дополним 1-й и 5-й блоки новым неактивным действием – «стоять на месте».

Таким образом, имеется уже пять типов движений: «вперёд», «назад», «поворот на право», «поворот на лево» и «стоять на месте». Тогда минимальным числом бинарных символов, с помощью которых их можно закодировать пять состояний робота является число три. Допустим, что 111 – движение вперёд, 101 – движение назад, 011 – движение налево, 110 – движение на право и 000 – стоять на месте. Тогда мы получим следующие три информационных блока:

1. **1** 110 111 000,
2. **1** 101 011 111,
3. **0** 101 011 000,

где жирным шрифтом указаны сенсорные, а остальные – моторные данные. Причём, судя по приведённой выше блок-схеме (рисунок 2), в третьем информационном блоке сенсорный бит может быть как 1, так и 0. Однако это не приводит к коллизии, так как после очередной итерации либо с помощью гетероассоциативной памяти, либо с помощью блока управления на выходе второго слоя нейронов так или иначе появится первый информационный блок.

Предложенная модель реализации поисковых движений была проверена с помощью имитационного компьютерного моделирования. Полученные результаты подтвердили работоспособность предложенной модели. В результате, на виртуальном полигоне (рисунок 4) функционировало шесть поисковых роботов, система управления каждого из которых состояла из предложенного нейросетевого процессора. На приведённом рисунке изображён фрагмент виртуального полигона с шестью роботами, изображёнными направленными треугольниками, и одним кругом – искомым объектом. Причём, при использовании указанной выше кодировки для управления виртуальным роботом понадобилась матрица синаптических связей W размером всего 10×10 элементов и один логический элемент И.

Таким образом, была доказана возможность реализации алгоритмов.

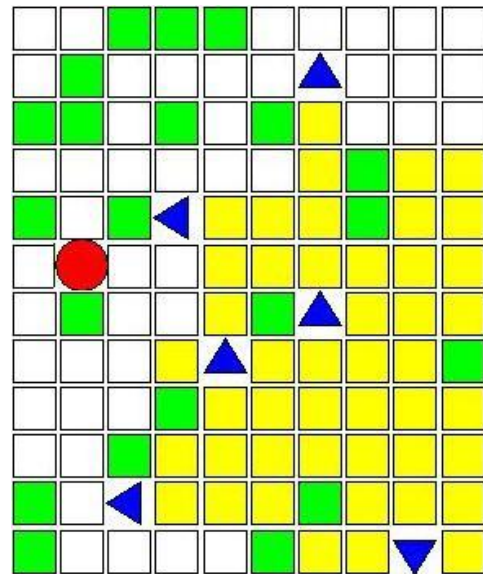


Рисунок 4 – Моделирование поискового движения

Полученная нейросетевая модель реализации поисковых движений представляет собой законченный компонент, который по своей структуре и выполняемым функциям полностью соответствует контуру экстремального управления описанной выше иерархической системы управления [Прокопович, 2010a]. Другим словами, блок моторов соответствует блоку эффекторов, блок управления и логический элемент – блоку принятия решений, блок сенсоров – блоку рецепторов, а гетероассоциативная память – базе знаний. Причём, движение информационных потоков полностью совпадает.

4. Выводы

В работе была предложена нейросетевая модель реализации поисковых движений на примере прохождения лабиринта мобильным роботом. Отличительной особенностью предложенной модели является то, что она способна реализовать выполнение ветвящихся алгоритмов и тем самым имитировать процесс управления техническими устройствами. Причём, предложенную модель можно рассматривать как нейросетевой процессор, который программируется методом обучения. Обучающая выборка представляет собой набор информационных блоков, интегрирующих в себе не только управляющие, но и сенсорные данные, необходимые для корректного выполнения заданного алгоритма.

Данная модель может интегрироваться практически без изменений в более общую, предложенную ранее автором, нейросетевую иерархическую систему управления робототехническими аппаратами, которая характеризуется наличием распределённой обработкой информации. Работоспособность предложенной модели была подтверждена

компьютерным моделированием.

Работа выполнена при частичной поддержке гранта БРФФИ-ГФФИУ Ф11К-169 «Разработка и исследование методов и алгоритмов скрытой и защищённой передачи информации в задачах группового управления мобильными роботами и подвижными системами».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Kosko, 1988] Kosko B. Bidirectional associative memories / В Kosko // IEEE Transactions on Systems, Man, and Cybernetics, 1988, vol. 18, no. 1. – P. 49–60.
- [MYROBOT.RU, 2007] Прохождение лабиринта: правила и алгоритмы [Электронный ресурс]. – 2007. – Режим доступа : http://myrobot.ru/articles/logo_mazesolving.php. – Дата доступа : 03.12.2012.
- [Амосов и др., 1991] Амосов, Н.М. Нейрокомпьютеры и интеллектуальные роботы / Н. М. Амосов [и др.] ; под общ. ред. Н.М. Амосова. – Киев : Наукова думка, 1991. – 272 с.
- [Гаазе-Рапопорт и др., 1987] Гаазе-Рапопорт, М.Г. От амёбы до робота: модели поведения / М. Г. Гаазе-Рапопорт, Д. А. Поспелов. – М. : Наука, 1987. – 288 с.
- [Головко, 2001] Головко, В.А. Нейронные сети: обучение, организация и применение. Кн. 4: учеб. пособие для вузов / В. А. Головко. – Под общ. ред. Л.И. Галушкина. – М: ИПРЖР, 2001. – 256 с.
- [Джонс, 2004] Джонс, М.Т. Программирование искусственного интеллекта в приложениях / М. Т. Джонс // Пер. с англ. А.И. Осипов – М.: ДМК Пресс, 2004. – 312 с.
- [Жданов, 2009] Жданов, А.А. Автономный искусственный интеллект / А. А. Жданов. – 2-е изд. – М. : БИНОМ, 2009. – 359 с.
- [Камшилов, 1979] Камшилов, М.М. Эволюция биосферы / М. М. Камшилов. – М. : Наука, 1979. – 256 с.
- [Крылов, 2007] Психология: учебник / под общ. ред. А.А. Крылова. – М. : Проспект, 2007. – 752 с.
- [Прокопович и др., 2010] Прокопович, Г.А. Моделирование коллективного поведения роботов для поисково-исследовательских задач / Г. А. Прокопович, В. А. Сычёв // Экстремальная робототехника: материалы междунар. науч.-техн. конференции (Москва, 18-20 мая 2010 г.). – СПб : Политехника-сервис, 2010. – С. 237-243.
- [Прокопович, 2009] Прокопович Г.А. Адаптивный нейросетевой классификатор / Г. А. Прокопович // Информатика, 2009, №3(23). – С. 68-81.
- [Прокопович, 2010а] Прокопович, Г.А. Иерархическая система управления с распределённой обработкой информации для интеллектуальных автономных систем / Г. А. Прокопович // Робототехника. Взгляд в будущее: материалы международного семинара. – СПб : Политехника-сервис, 2010. – С. 195-198.
- [Прокопович, 2010б] Прокопович, Г.А. Нейросетевой блок памяти для адаптивной работы сложных технических систем в динамической среде / Г. А. Прокопович // Информатика, 2010, №2(26). – С. 54-65.
- [Прокопович, 2011] Прокопович, Г.А. Нейросетевая ассоциативная модель достижения цели / Г. А. Прокопович // Сборник тезисов XVIII Международной научной конференции студентов, аспирантов и молодых учёных «Ломоносов-2011», секция «Вычислительная математика и кибернетика». – М. : МАКС Пресс, 2011. – С. 91-93.
- [Прокопович, 2012а] Прокопович, Г.А. Применение гетероассоциативных нейронных сетей для записи и восстановления информации / Г. А. Прокопович // Информатика, 2012, №2(34). – С. 38-49.
- [Прокопович, 2012б] Прокопович, Г.А. Модель нейросетевой ассоциативной памяти для управления манипулятором / Г. А. Прокопович // Информатика, 2012, №3(35). – С. 16–25.
- [Самарин, 2002] Самарин, А.И. Модель адаптивного поведения мобильного робота, реализованная с использованием идей самоорганизации нейронных структур / А. И. Самарин // IV Всероссийская научно-техническая конференция "Нейроинформатика-2002". Мат-лы дискуссии "Проблемы интеллектуального управления – общесистемные, эволюционные и нейросетевые аспекты". – М.: МИФИ, 2003. – С. 106-120.
- [Симонов, 1981] Симонов, П.В. Эмоциональный мозг /

П. В. Симонов. – М. : Наука, 1981. – 140 с.

[Уоссермен, 1990] Уоссермен, Ф. Нейрокомпьютерная техника: теория и практика / Ф. Уоссермен. – М.: Мир, 1990. – 240 с.

NEURAL NETWORK MODEL FOR MOBILE ROBOT SEARCH MOVEMENTS

Prakapovich R.A.

United Institute of Informatics Problems of NAS of Belarus

rprakapovich@robotics.by

A model of a neural network processor, which is based on the principles of functioning of heteroassociative memory and which is able to implement branching algorithms, is suggested here. The description of a search algorithm for maze solving is taken as an example. The offered model can be considered as a neural network component within a more general hierarchical neural network control system, suggested by the author earlier.

INTRODUCTION

Among the main trends in the intelligent robot control systems development is the problem of maintenance of their full or partial autonomy. Among advanced methods for management of intelligent systems an important role belongs to the bionic method. Such an approach pays much attention to studying of fundamental principles of functioning of natural control systems. It is to note, artificial neural networks are most often implementations of various models of practical bionic systems.

MAIN PART

In the earlier paper [Прокопович, 2010а] a neural network model of a hierarchical control system with distributed information processing for autonomous mobile robots has been presented. This system includes three control circuit as follows: reactive (safety function), optimal (search function) and cognitive (planning function). A model of a neural network processor, which is able to implement branching algorithms represents in this paper. Proposed neural mobile robot control system on the base of «the right-hand rule» is able to path find of maze.

CONCLUSION

The neural network model of realization of search movements on the example of maze solving for the mobile robot is described in this paper. A distinctive feature of this model is that it can implement branching algorithms and, as a result, imitate the process of control and management of technical devices. The presented model can be integrated almost without any changes into a more general hierarchical neural network control system, suggested by the author earlier. Functionality of the suggested model is proved by the computer simulation.



УДК 004.896,81'322;004.934;004.912

КОМПОНЕНТЫ НАТУРАЛЬНО-РЕЧЕВОГО КОМПЛЕКСА ОПРЕДЕЛЕНИЯ ФОРМЫ ПРЕПЯТСТВИЯ ДЛЯ МОБИЛЬНЫХ РОБОТОВ

Сычѳв В.А. *, Гецевич Ю.С. *, Прокопович Г.А. *

**Объединѳнный институт проблем информатики НАН Беларуси,
г. Минск, Республика Беларусь*

vsychyov@robotics.by

yury.hetsevich@gmail.com

rprakapovich@robotics.by

Предлагается экспериментальный многокомпонентный комплекс для управления мобильным роботом, предназначенным для функционирования в человеческой среде в автоматическом режиме или под управлением оператора. Компонент распознавания образов работает на основе алгоритма, интерпретирующего данные от двумерного сканирующего дальномера. Компонент диалога подготавливает данные для передачи на озвучивание. Компонент синтеза речи по тексту озвучивает идентифицированные данные и сформированные на их основе сообщения для пользователя. Предложенные алгоритмы реализованы программно и протестированы на действующем мобильном роботе.

Ключевые слова: дальномер; система управления; автономный мобильный робот; компонент распознавания образов; компонент синтеза речи потексту.

ВВЕДЕНИЕ

Для решения задачи ориентирования автономного мобильного робота в пространстве применяются разнообразие сенсоры, в числе которых энкодеры, акселерометры, гироскопы, дальномеры, видеосенсоры [Каляев, 2007]. Особое место занимают сканирующие дальномеры. Приборы данного типа позволяют получить двумерное или трёхмерное изображение пространства перед роботом на основе данных измерения расстояний до окружающих предметов. Сканирующие дальномеры дают возможность получить важную для ориентирования робота информацию, однако для этого требуется применение соответствующих алгоритмов обработки данных, получаемых в процессе сканирования пространства перед роботом.

Предлагаемый алгоритм позволяет построить систему распознавания геометрических форм объектов, которые можно описать такими примитивами, как окружность или угол, полученные с помощью двумерного сканирующего дальномера (рисунок 1). Предлагаемый алгоритм ориентирован на применение в компонентах системы управления сервисных и бытовых роботов и может быть реализован на базе маломощных микроконтроллеров. Основное назначение подобных роботов – автоматическая уборка и мониторинг

жилых и производственных помещений.

Так как подобные роботы чаще всего функционируют в окружении людей, полезным дополнением к их системе управления является блок синтеза речи, с помощью которого робот может озвучивать свои действия.

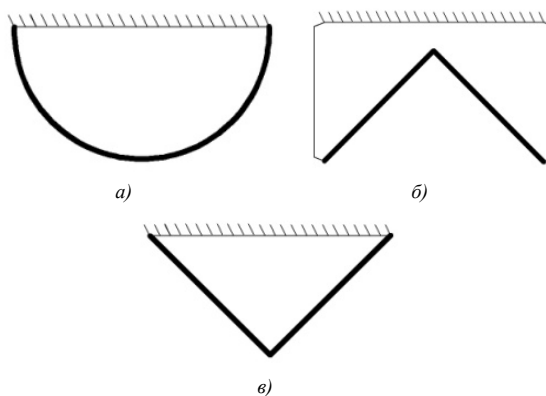


Рисунок 1 – Геометрические примитивы: а) – внешняя окружность, б) – внутренний угол, в) – внешний угол

Блок-схема автономного мобильного робота, построенного с использованием данного алгоритма и с применением блока синтеза речи, приведена на рисунке 2. Для решения задачи распознавания в сенсорных данных, полученных при сканировании пространства перед роботом, приведѳнных выше

примитивов могут быть использованы различные методы и алгоритмы.



Рисунок 2 – Блок-схема системы управления автономного мобильного робота

Среди них выделяются корреляционные методы, аппарат искусственных нейронных сетей [Хайкин, 2006], метод опорных векторов [Vapnik, 1998], использование систем с детерминированным хаосом [Кузнецов, 2006; Дмитриев, 1991; Дмитриев, 2001].

1. Применение систем с детерминированным хаосом для фильтрации и распознавания сигналов

Существует ряд работ, посвящённых использованию систем, демонстрирующих хаотические режимы работы, для фильтрации и распознавания сигналов [Кузнецов, 2006; Дмитриев, 1991; Дмитриев, 2001]. В их основу положен принцип стабилизации динамической системы с помощью обратной связи с задержкой по времени [Schöll, 2008]. В англоязычной литературе для данного метода принято название *DelayedFeedbackControl* (сокращённо *DFC*).

Блок-схема, иллюстрирующая стабилизацию динамической системы с помощью метода *DFC*, приводится на рисунке 3.

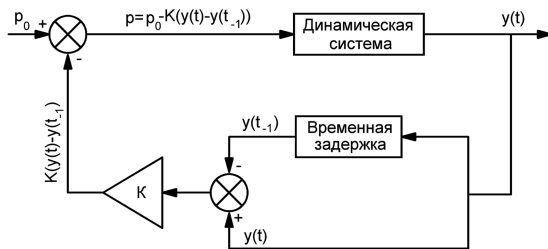


Рисунок 3– Принцип стабилизации динамической системы с помощью отрицательной обратной связи с задержкой по времени (*DelayedFeedbackControl*)

На рисунке 3 приняты следующие обозначения: P_0 – входной сигнал, P – результирующее входное воздействие на динамическую систему, K – линейное усиление, $y(t)$ – выходное состояние динамической системы в текущий момент времени,

$y(t_{-1})$ – выходное состояние динамической системы в предыдущий момент времени.

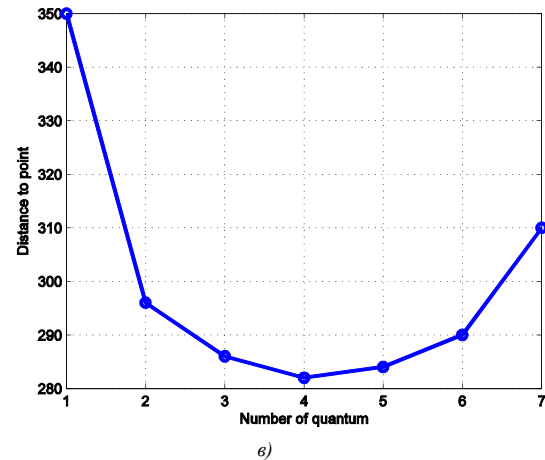
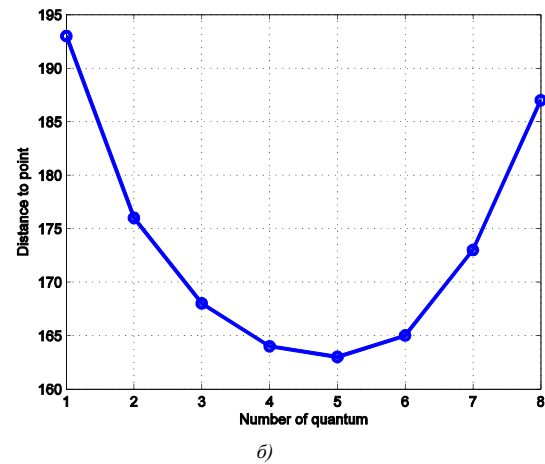
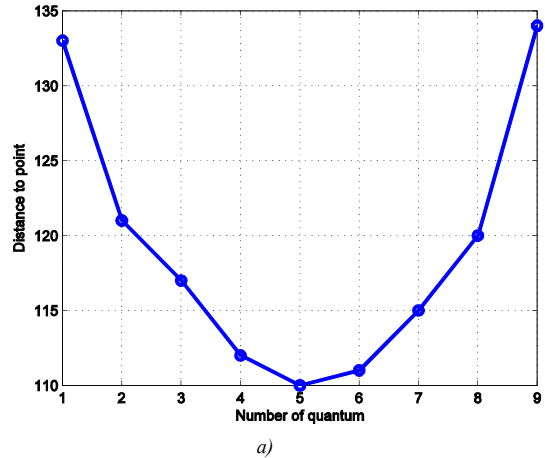
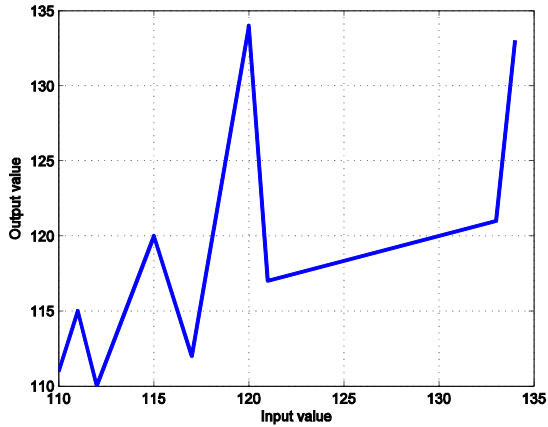


Рисунок 4– Изображение округлой поверхности, полученное с помощью двумерного сканирующего дальномера с расстояния 110 мм (а), 163 мм (б) и 281 мм (в)

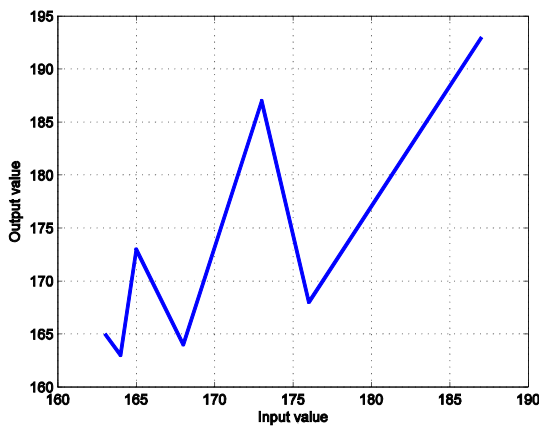
Функционирование блоков суммирования проиллюстрировано знаками «+» и «-».

Результирующее входное воздействие на динамическую систему вычисляется по формуле (1):

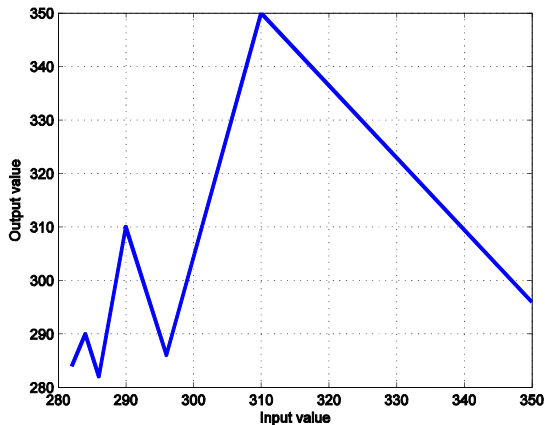
$$p = p_0 - K(y(t) - y(t_{-1})). \quad (1)$$



a)



б)



в)

Рисунок 5 – Варианты передаточной функции динамической системы

Критерием стабильности системы является выполнение условия (2):

$$K(y(t) - y(t_{-1})) = 0. \quad (2)$$

Для того чтобы применить метод *DFC* при распознавании сигналов, необходимо использовать

динамическую систему с заранее определённой передаточной функцией. Передаточная функция должна подбираться таким образом, чтобы при последовательном вводе значений эталонного сигнала p_0 система стабилизировалась, а при вводе неизвестного сигнала входила в хаотический режим работы. Следовательно, стабилизация системы при распознавании служит признаком того, что множество входных сигналов p_0 относится к некоторому известному множеству.

При практической реализации устройств распознавания или фильтрации сигналов на основе метода *DFC* используется динамическая система с дискретным временем, переходная характеристика которой реализована с помощью нелинейного усилителя или цифрового сигнального процессора.

Такое устройство может применяться для того, чтобы, к примеру, на сигналах, полученных от двумерного сканирующего дальномера, выделить последовательность сигналов, относящуюся к округлому телу (рисунок 4) или телам другой искомой формы (рисунок 8, а-в).

2. Применение метода *DFC* для распознавания сигналов двумерного дальномера

Если предположить, что передаточные функции для объектов одного класса будут схожи, и притом значительно отличаться от передаточных функций объектов других классов, то становится возможным реализовать новый способ распознавания. Данный способ основан на анализе передаточной функции нелинейного элемента, при которой динамическая система (рисунок 3) стабилизируется. Таким образом, становится возможным достижение большей универсальности системы распознавания, выраженной в инвариантности к взаимному расположению дальномера и сканируемого тела.

Однако, как видно из рисунка 5, передаточные функции нелинейного элемента, построенные на основе формул 1 и 2, значительно отличаются друг от друга даже для одного и того же объекта – округлого тела, отсканированного дальномером с разным удалением. Вызвано это тем, что при указанном способе формирования передаточной функции система оперирует расстояниями между дальномером и рядом точек исследуемого предмета. Очевидно, что при таком подходе достижение инвариантности к взаимному положению робота и препятствия затруднительно.

Следовательно, необходимо изменить способ формирования передаточной функции динамической системы.

К примеру, для формирования передаточной функции можно установить выходное значение нелинейного элемента, равное $y(t) = p_0$, если $p > 0$, и $y(t) = y(t-1) + p_0$, если $p = 0$.

Блок-схема алгоритма формирования передаточной функции динамической системы на основе описанного выше принципа приведена на рисунке 6.

На блок-схеме (рисунок 6) N – число элементов входного множества p_0 .

Варианты передаточной функции динамической системы, полученной с помощью предложенного алгоритма, показаны на рисунке 7 синим цветом и соответствуют входным сенсорным данным, приведённым на рисунке 4.

При этом передаточные функции нелинейных элементов, полученные для схожих образов, также будут близки по форме. Это даёт возможность заранее выработать и сохранить в памяти системы управления роботом шаблоны передаточных функций нелинейного элемента для каждого из примитивов, показанных на рисунке 1.

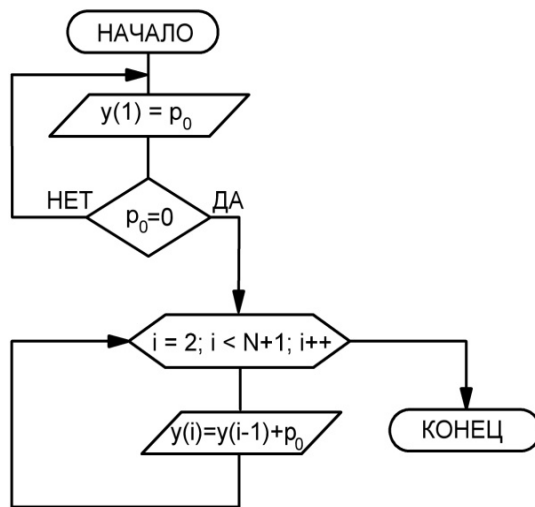
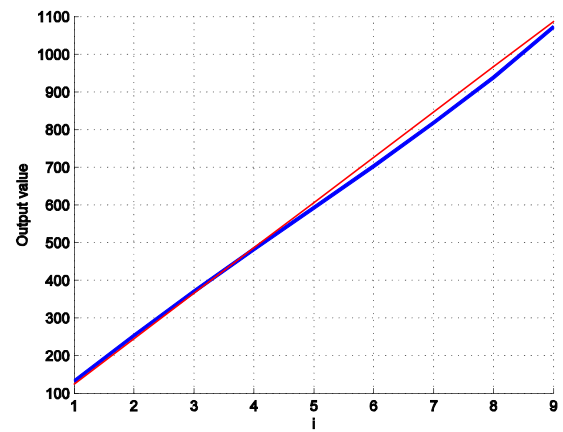


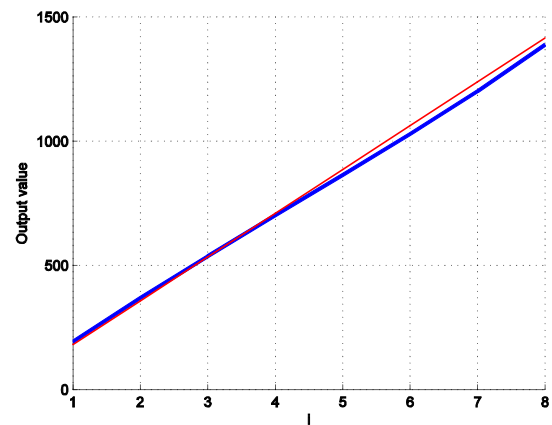
Рисунок 6 – Блок-схема алгоритма формирования передаточной функции динамической системы

После сравнения полученной передаточной функции с имеющимися у нас шаблонными образами становится возможным отнести предмет на изображении к одному из трёх классов (рисунок 1) либо сделать вывод о том, что пространство перед роботом свободно.

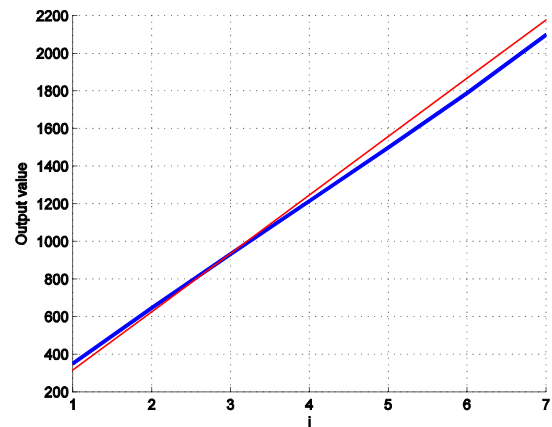
Через графики на рисунке 7 проведены прямые линии, обозначенные красным цветом, чтобы показать, что форма передаточной функции хотя и близка к прямой, однако прямой не является.



a)



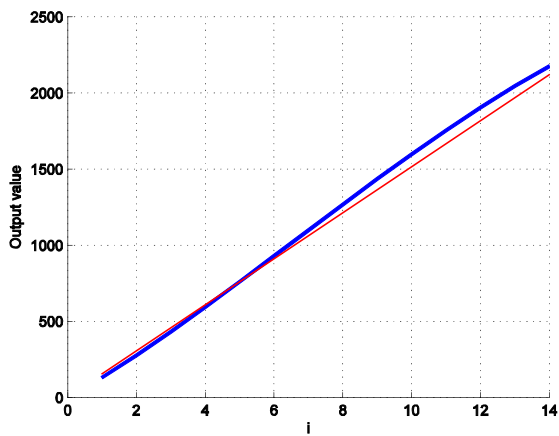
б)



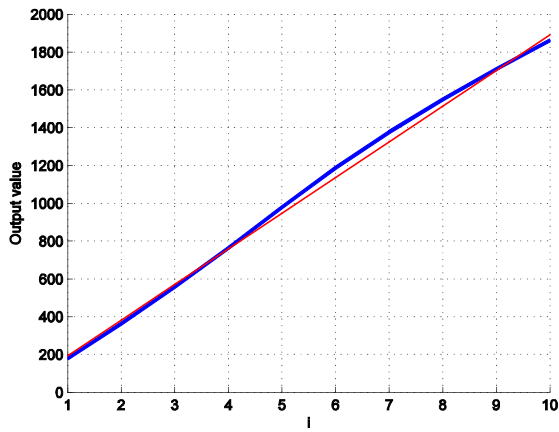
в)

Рисунок 7 – Варианты переходной характеристики динамической системы

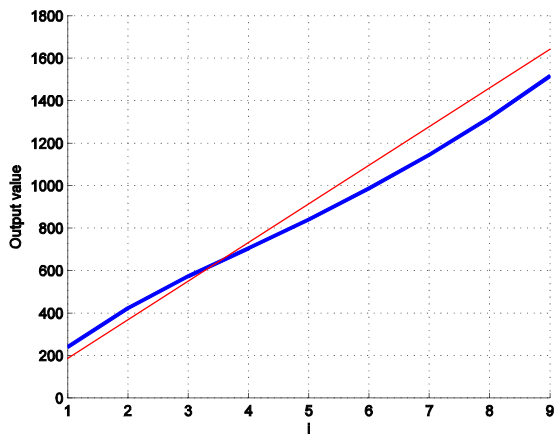
Таким образом, для каждого нового входного образа А следует построить такую передаточную функцию нелинейного элемента, при которой система будет стабилизирована только при поступлении на вход данного образа А (рисунок 3). В данном случае критерием стабильности является соблюдение равенства $p = 0$.



a)



б)



в)

Рисунок 8 – Изображения, полученные с помощью сканирующего дальномера: а) – внутренней окружности, б) – внутреннего угла, в) – внешнего угла, и форма соответствующей передаточной функции динамической системы г)-е)

Следовательно, при программно-аппаратной реализации описанного алгоритма распознавания входного образа каждый раз необходимо строить новую передаточную функцию нелинейного элемента, позволяющую стабилизировать систему (рисунок 3).

3. Экспериментальное исследование результатов

Для того чтобы проверить работоспособность предлагаемого многокомпонентного комплекса реальных данных и выявить границы его применимости, комплекс был программно реализован на языке программирования Си и протестирован с помощью экспериментального мобильного робота (рисунок 9).

На борту робота был установлен инфракрасный дальномер Sharp 2Y0A21 и сервопривод, вращающий дальномер в горизонтальной плоскости в диапазоне от -39 до 39 градусов с дискретностью $1,5$ градуса.

Робот оснащён одноплатной системой управления на основе восьмимбитного микроконтроллера AVR. Система управления включает в себя также драйвер двигателей постоянного тока для управления ходовой платформой и радиомодем типа XBeePro для связи с персональным компьютером.

В результате проведения эксперимента было установлено, что при соблюдении указанных выше условий (расстояние до препятствий $18-23$ см) примитивы определялись верно в 70% случаев. Повысить точность и рабочий диапазон дальностей возможно применением более совершенных дальномеров. Диапазон расстояний, измеряемых дальномером, – от 10 до 80 сантиметров, однако было введено программное ограничение верхней границы измеряемого расстояния на уровне 40 см, так как на больших расстояниях дискретность сканирования в $1,5$ градуса не позволяет получить достаточно точное изображение предмета, находящегося перед роботом.

Увеличение же дискретности нежелательно, так как сопряжено с увеличением времени сканирования, которое при описанных параметрах составляет 10 секунд. Для защиты от помех применён программно реализованный медианный фильтр. При предъявлении округлого тела робот загорался оранжевый светодиод, внутреннего угла – синий, а внешнего угла – зелёный.

Кроме того, с помощью блока диалога осуществлялось формирование пакета данных, включающих MAC-адрес робота, результаты измерений расстояний до препятствия и название класса, к которому было отнесено препятствие в соответствии с рисунком 1.

Ниже приводится пример пакета данных, получаемых от робота в процессе функционирования.

```
*Robot ON*
MAC:$4090B9F8#
```

```
=====
distance
```

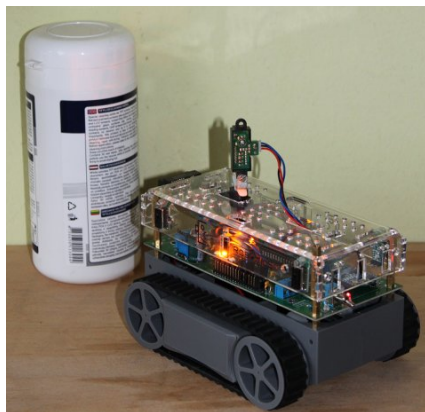
```
...
```

```
number_of_quatnt = 18
```

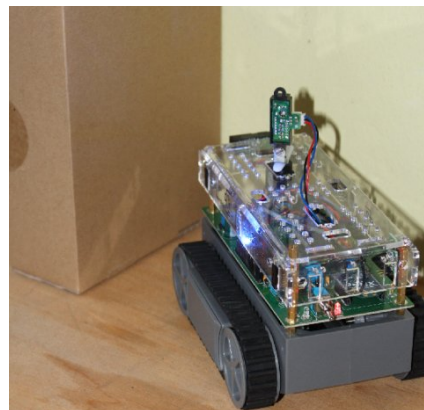
```
326  
267  
243  
243  
236  
233  
233  
233  
233  
233  
235  
238  
243  
247  
251  
259  
299
```

```
*object_type: circle*  
*Robot OFF*
```

Ряд дистанций до препятствия, следующий за словом *distance*, опущен. Приводятся только те расстояния, которые не превышают 40 см, так как именно они служат входными данными для блока распознавания формы препятствия. Фрагмент, обозначенный с помощью «*», идентифицируется и обрабатывается в необходимые орфографические слова подпроцессором текстового компонента и озвучивается с помощью двуязычного синтезатора речи по тексту.



а)



б)



в)

Рисунок 9 – Классификация препятствий роботом: а) – круглое тело, б) – внутренний угол, в) – внешний угол

4. Компонент речевого вывода информации о работе синтезатора белорусского и русского языков

Для перевода информации, полученной от робота в текстовой форме в речевую, используется два компонента, основанных на синтезаторах речи по тексту на белорусском [Гецавічы інш., 2011] и русском языках [Лобановидр., 2008]. Данные компоненты реализованы в качестве сторонних подключаемых модулей. В результате такого подключения к системе двуязычного синтезатора речевую информацию пользователь получает озвученную информацию от робота о конкретном распознанном объекте.

Синтез устной речи по тексту для каждого языка осуществляется на основе лексико-грамматического анализа входного текста путём моделирования процессов речеобразования с учётом правил произношения звуков и интонирования для каждого языка.

Текст поступает на вход синтезатора и далее подвергается

ся последовательной обработкой рядом специализированных процессоров в соответствии с общей структурой синтезатора речевой потоку, представленной на рисунке 10. Синтезатор включает следующие компоненты: текстовый процессор, просодический процессор текста и сигнала, фонетический процессор и акустический процессор. Каждый из этих модулей поддерживается набором ответов в соответствии с базой данных правил.

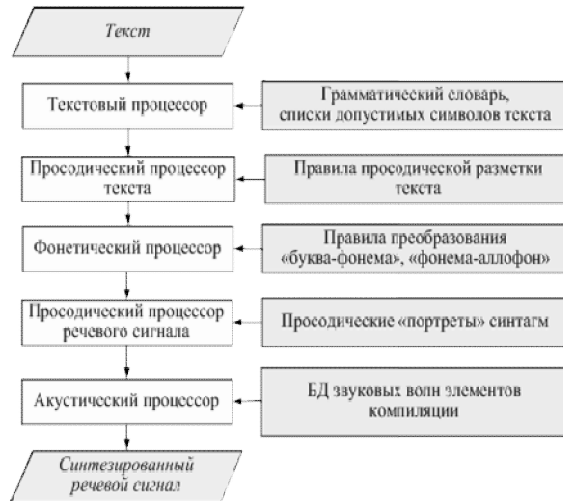


Рисунок 10 – Структура системы синтеза речи по тексту

Наибольшей модификации для озвучивания сообщений пользователю от мобильного робота подвергся текстовый процессор. Для него были доработаны плагины, позволяющие обрабатывать текст в следующей последовательности: очистка текста, идентификация и обработка сообщений от мобильного робота, преобразование знаков (аббревиатур, сокращений, чисел и др.), расстановки слесовых ударений и грамматических признаков словоформ.

В самом простом случае плагин идентификации сообщений реагирует на начальные «*Robot ON*» и конечные «*Robot OFF*» последовательности символов. В идентифицированном фрагменте плагин-обработчик находит последовательность символов «*object_type:». Тип объекта он переводит по встроенному словарю возможных сообщений с внутреннего машинного языка робота на естественный язык пользователя. Например, синтезатор белорусской речи по тексту по входной строке «*object_type: circle*» будет сгенерирован орфографический текст «*перада мною знаходзіцца прадмет у выглядзе акружнасці*».

Далее преобразованный текст поступает на вход просодического, а затем фонетического процессоров. В результате работы просодического процессора текст разделяется на синтагмы, акцентные единицы, далее он разделяется на элементы акцентных единиц: интонационно-предъядро, ядро и изядро. И наконец, просодический процессор, в соответствии с базой данных просодических «портретов» синтагм,

для языкового сигнала устанавливает значимость амплитуды, продолжительность фоновых частот основного тона для каждого элемента акцентных единиц.

Акустический процессор на основе информации о том, как и насколько необходимо синтезировать аллофоны, а также как и насколько просодические характеристики должны быть приписаны каждому аллофону, генерирует речевой сигнал путем компиляции отрезков натуральных звуковых волн соответствующих аллофонов мультифонов. Таким образом, входной текст преобразуется в речевой сигнал.

Заключение

Таким образом, был разработан экспериментальный многокомпонентный комплекс для управления мобильным роботом, предназначенным для функционирования в человеческой среде в автоматическом режиме или под управлением оператора. Мобильный робот способен идентифицировать объекты-препятствия трёх типов и озвучивать их естественным языком для пользователя.

Важно отметить, что спроектированная ранее архитектура двуязычного синтезатора речи по тексту позволила быстро масштабировать этот синтезатор для организации взаимодействия робота и пользователя.

Целью дальнейших исследований является повышение рабочего диапазона дальностей и добавление новых примитивов в набор распознаваемых изображений. Кроме этого, в будущем планируется идентифицировать и обрабатывать расстояния от мобильного робота до объектов, а также генерировать вопросы от робота пользователю типа «з якого боку мне аб'язджаць гэты аб'ект-перашкоду?» и принимать на них ответы.

Работа выполнена в лаборатории моделирования самоорганизующихся систем при поддержке гранта БРФФИ-ГФФИУФ11К-169 «Разработка исследований методов алгоритмов скрытой защищённой передачи информации в задачах группового управления мобильными роботами и подвижными системами» совместно с лабораторией распознавания и синтеза речи ОИПИ НАН Беларуси.

Библиографический список

- [Каляев, 2007] Интеллектуальные роботы: учебное пособие для вузов / И.А. Каляев [и др.] ; под общ. ред. Е.И. Юревича. – М. : Машиностроение, 2007. – 360 с.
- [Хайкин, 2006] Хайкин, С. Нейронные сети: полный курс, 2-е изд. / С. Хайкин. – М. : Издательский дом «Вильямс», 2006. – 1104 с.
- [Vapnik, 1998] Vapnik, V.N. Statistical learning theory / V.N. Vapnik. – New York : A Wiley-interscience publication, 1998. – 732 p.
- [Кузнецов, 2006] Кузнецов, С.П. Динамический хаос / С.П. Кузнецов. – 2 изд. – М. : Физматлит, 2006. – 356 с.

[Дмитриев, 1991]Дмитриев, А.С. Запись и восстановление информации в одномерных динамических системах / А.С. Дмитриев // Радиотехника и электроника. –1991. –Т.36. –№.1, с.101-108.

[Дмитриев, 2001]Новое в синергетике: Взгляд в третье тысячелетие. Динамический хаос как носитель информации / А.С. Дмитриев ; под общ.ред. Г.Г.Малинецкого. – М. :Наука, 2001. – 480 с.

[Schöll, 2008] Handbook of Chaos Control / edited by E. Schöll and H. G. Schuster – 2nd ed. – Wiley-VCH Verlag GmbH & Co. KGaA, 2008. – 819 p.

[Геце́вич і інш., 2011]Геце́вич, Ю.С. Синтэза тарбеларускага і рускага маўлення патакспецыяльна для стаячых і рухоўных платформаў / Ю.С. Геце́вич, Д.А. Пакладок // Сб. матэрыялаў І Молодзёжнага інавацыйнага форуму «Навука і бізнэс – 2011» (Мінск, 14–18 нояб. 2011 г.). – Мінск : Беларускае навука, 2011. – С. 14–15.

[Лобанов и др., 2008]Лобанов, Б.М. Компьютерный синтез и клонирование речи / Б.М. Лобанов, Л.И. Цирульник // Минск : Белорусская наука, 2008. – 344 с.

COMPONENTS OF THE NATURAL SPEECH COMPLEX FOR IDENTIFICATION OF DIFFERENT TYPES OF OBSTACLES FOR MOBILE ROBOTS

Sychou U. A., Hetsevich Yu. S., Prakupovich R. A.

*United Institute of Informatics Problems
of NAS of Belarus,
Minsk, Republic of Belarus*
vsychyov@robotics.by
yury.hetsevich@gmail.com
rprakupovich@robotics.by

INTRODUCTION

Among robots designed to operate in human environments, there are robots for monitoring of premises, automatic cleaning, etc. Such robots need to be able to move around their environment automatically. They also need built-in means of interaction with users, so the last could control a robot and change its route. Most often such robots operate in the human environment. That is why an extremely usable robot will be the one, that can voice its operations or facts, which it establishes, with the help of internal algorithms and hardware.

In order to implement functions of identification and voicing of characteristics of obstacle objects in front of mobile robots a multi-component control system has been developed. It consists of a component of identification of obstacles and a component of bilingual text-to-speech synthesis.

MAIN PART

The component of obstacles identification is founded on the algorithm that analyzes data obtained from a two-dimensional scanning range finder, and recognizes geometric primitives such as a circle, outer or inner angle.

The component of bilingual text-to-speech synthesis is based on the algorithm of lexical and

grammatical analysis of an input text by modeling processes of speech production according to pronunciation and intonation rules for the Belarusian and Russian languages.

In order for this speech synthesizer to voice messages from a robot to a user by understandable lexical means, some necessary plugins of the text component have been developed, and dictionaries for translation of machine commands into Belarusian or Russian have been added. As a result, now the plugins can identify and analyze commands of mobile robots in an input symbol stream (or text) of the synthesizer. Thus, when a mobile robot approaches to an obstacle, a user hears a message such as *«there is an obstacle object in front of me»* (in the original: *«перадамнаю знаходзіцца на прадмету вы глядзеакружнасці»*).

The proposed components have been implemented as software and tested with an experimental mobile robot.

CONCLUSION

Thus, the authors have developed the experimental multi-component complex for mobile robot control, which is able to operate in the human environment, either automatically or under user's control. The mobile robot is able to distinguish obstacle objects of three types and voice data for the user with the help of the natural language.

It is important to note that the architecture of bilingual text-to-speech synthesis, which has been designed earlier, allows to scale quickly this synthesizer for human-robot interaction.

Our next research goal is to increase the working distance range and to add new primitives to the set of recognizable images. Also in future it is planned to identify and analyze the distance from a robot to an object, as well as to generate questions which a robot will address to a user, such as *«from which side should I bypass this object obstacle?»* (in the original: *«з якога боку мне аб'язджаць гэты аб'ект-перашкоду?»*), and, of course, accept answers.

The work has been accomplished in the Laboratory of self-organization system modeling with the support of the BRFFR-SFFRU grant F11K-169 «Development and research of methods and algorithms for hidden and secure information transmission in the problems of group control of mobile robots and active systems» in cooperation with the Laboratory of Speech Recognition and Synthesis of the UIIP of the NAS of Belarus.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.81:159.942.52

МОДЕЛИРОВАНИЕ ДВИЖЕНИЙ ЧЕЛОВЕКА НА ОСНОВЕ НЕЧЕТКИХ ТЕМПОРАЛЬНЫХ ВЫКАЗЫВАНИЙ

Заболеева-Зотова А.В. *, Бобков А.С. *, Петровский А.Б. **

** Волгоградский государственный технический университет,*

г. Волгоград, Россия

zabzot@vstu.ru

bobkovart@gmail.com

*** Институт системного анализа Российской академии наук,*

г. Москва, Россия

pab@isa.ru

В статье рассматривается подход к формализованному описанию движений человека с использованием последовательных нечетких темпоральных высказываний.

Ключевые слова: движения человека, нечеткое темпоральное событие, нечеткое последовательное темпоральное высказывание.

ВВЕДЕНИЕ

Автоматизированная обработка информации об эмоциональном состоянии человека является актуальной задачей, решение которой позволило бы решить ряд экономических, социальных и бытовых проблем [Бобков и др., 2011]. Программные комплексы, осуществляющие интеллектуальную обработку потока видеоданных, могут оказаться полезными везде, где требуется анализ поведения человека, в том числе и его эмоциональной составляющей – на транспортных узлах, в крупных магазинах и других местах.

Отсутствие моделей и методов, обеспечивающих адекватную идентификацию эмоциональных реакций по телодвижениям человека, не позволяет пока эффективно автоматизировать этот процесс. Остается актуальной и проблема формального описания проявления эмоций через телодвижения.

В данной работе предложен подход к моделированию движений человека с целью идентификации его эмоционального состояния. Подход основан на использовании понятий нечеткого темпорального высказывания и нечеткого темпорального события [Ковалев и др., 2011], методах определения адекватности моделей по критерию истинности нечеткого темпорального высказывания [Bernshtein et al, 2009].

1. Входные данные модели

Для представления характерных движений человека разработана векторная модель скелета, состоящая из 22 точек, которые соответствуют основным анатомическим частям (суставами и отделами позвоночника) человеческого тела. Входными данными модели служит информация об изменении углов поворота анатомических узлов при движении человека.

Идентификация движений человека в видеопотоке проводилась с использованием безмаркерной технологии. Она обеспечивает анализ состояния человека, не сковывая его движения, и как следствие, не ограничивает его эмоциональные реакции [Bobkov et al, 2012]. Сопоставление известных коммерческих систем, осуществляющих безмаркерное распознавание людей на видео, позволило выбрать систему Brekel Kinect как наиболее подходящую для отображения движений человека и дальнейшего анализа изображения. Система Brekel Kinect, работающая с сенсором от компании Microsoft, дает возможность самостоятельно выбирать положение сенсора. Если не требуется работа системы в режиме реального времени, допустимо использование IpiStudo совместно с несколькими видекамерами, в том числе и USB веб-камерами.

Векторная модель человека в системе определяется форматом описывающего её файла. Были проанализированы преимущества и недостатки различных форматов для анимации, в том числе, Biovision Hierarchy (bvh), COLLADA (dae), VALVE Source Engine Animator SMD (smd). Учитывая возможности систем (в частности, Brekel Kinect и IriStudo) для захвата движения и экспорта анимации, был выбран формат bvh.

2. Формализованное описание движений человека

Продолжительность движения анатомического узла в векторной модели скелета измеряется в кадрах. Для описания длительности движения на ограниченном естественном языке введена нечеткая темпоральная переменная «Продолжительность», определенная на следующем множестве термов: «Нулевая», «Очень короткая», «Короткая», «Умеренная», «Длительная», «Очень длительная». Функции принадлежности термов представлены функцией Гаусса.

Формализация движений анатомических узлов осуществляется с помощью семейства лингвистических переменных (ЛП) «Скорость изменения угла». Множество термов ЛП «Скорость изменения угла» состоит из следующих элементов: «Очень быстрое уменьшение», «Быстрое уменьшение», «Среднее уменьшение», «Медленное уменьшение», «Очень медленное уменьшение», «Нулевая», «Очень медленное увеличение», «Медленное увеличение», «Среднее увеличение», «Быстрое увеличение», «Очень быстрое увеличение». Графики функций принадлежности термов переменной «Скорость изменения угла» представлены на рисунке 1:

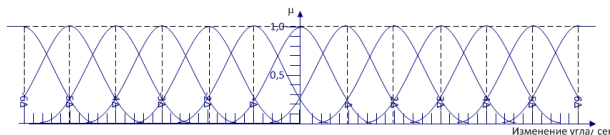


Рисунок 1 – График функций принадлежности термов переменной «Скорость изменения угла».

Каждая лингвистическая переменная характеризует движение определённого сустава (точки сгиба) в модели скелета человека. В предлагаемой модели точки сгиба сгруппированы в зависимости от значений максимальной подвижности (МП) суставов. Число лингвистических значений (ЛЗ) в лингвистических переменных постоянное, но сами значения могут различаться. Это зависит от максимальной подвижности и типа сустава (с тремя, с двумя, с одной осью вращения) или другой части тела, в которой происходит изменение угла поворота.

Для определения ЛП, описывающих 22 точки векторной модели скелета, найдено соответствие между узлами и анатомическими местами изменения углов при движении человека. С этой

целью был проведен анализ нормальной подвижности суставов, которые не считаются отклонениями, у спортсменов (пловцов, гимнастов, легкоатлетов) и людей, не занимающихся спортом.

Каждая группа узлов со схожими величинами МП описывается при помощи ЛП, максимум функции принадлежности которой достигается каждые Δ градусов за секунду. Данный набор ЛП может настраиваться в зависимости от того, какой вид движения человека описывается. Например, для малых периодических колебаний (постукивание по столу пальцами руки, покачивание, «переминание» с ноги на ногу) было экспериментально установлено, что наиболее подходящее значение Δ равно 1/18 от максимальной подвижности сустава.

Используя введенные выше лингвистические переменные, представим движение сустава вокруг одной из осей в виде нечеткого темпорального события (НТС). Так как события расположены на временной оси последовательно друг за другом, то движение можно описать последовательным нечетким темпоральным высказыванием (ПНТВ).

Пример: «Наблюдается среднее уменьшение угла очень короткой продолжительности, за которым следует практически нулевая стабилизация изменения угла, за которой следует среднее увеличение угла очень короткой продолжительностью». Модель данного описания движения представляет собой ПНТВ вида

$$W = (B_{-3} \text{ rtf } D_1) \text{ rtsn}(B_0 \text{ rtf } D_0) \text{ rtsn}(B_{+3} \text{ rtf } D_1).$$

Здесь rtf – нечеткое темпоральное отношение, rtsn – темпоральное отношение непосредственного следования, ЛЗ нечеткой переменной B «Скорость изменения угла»: B_{-3} – Среднее уменьшение, B_0 – Стабилизация, B_{+3} – Среднее увеличение; ЛЗ нечеткой темпоральной переменной D «Продолжительность»: D_1 – Очень короткая, D_0 – Нулевая.

3. Интерпретация последовательных нечетких темпоральных высказываний, описывающих движения человека

Для поиска определённого движения человека по запросу, представленному на ограниченном естественном языке, разработана интерпретирующая модель последовательных нечетких темпоральных высказываний, описывающих движения. Алгоритмическая схема модели дана на рисунке 2.

Из векторной модели скелета, располагая границами диапазонов поиска определённого движения и именем анализируемой части тела, можно получить массив значений углов вращения узла относительно одной из осей X , Y или Z . Далее формируются дочерний массив углов ρ_i , в котором находятся только те значения, которые попадают в анализируемый промежуток, и матрица разностей

углов, принадлежащих разным кадрам для одного узла. Элементы матрицы образуются по следующему правилу: $\rho_{ij}=\rho_j-\rho_i$ при $j>i$, $\rho_{ij}=0$ при $j\leq i$.



Рисунок 2 – Алгоритмическая схема интерпретирующей модели ПНТВ, описывающих движения человека

Для вычисления значения функции принадлежности лингвистической переменной «Скорость изменения угла» матрица разностей углов преобразуется к виду, в котором абсолютные изменения координат выражаются через шаг Δ максимальной подвижности, зависящей от вида анализируемой части тела.

В основе модели интерпретации последовательного нечеткого темпорального высказывания лежит критерий истинности, характеризующий возможность семантического сопоставления исходного фрагмента динамического процесса δ_q и признака q . Семантическое сопоставление задается формулой

$$J(q/\delta_q) = F_q(\delta_q) \& \mu_{Lq}(\delta_q).$$

Здесь $F_q(\delta_q)$ – характеристическая функция, устанавливающая семантическое отношение между нечеткими значениями переменных динамического процесса; $\mu_{Lq}(\delta_q)$ – функция принадлежности терма L_q лингвистической переменной L .

Критерий истинности $J(W/S)$ ПНТВ W относительно динамического процесса S записывается в виде

$$J(W/S) = \max_{I \in V} (J(W/S)_I),$$

где V – множество всех возможных интерпретаций I . В частности, критерий истинности ПНТВ W относительно динамического процесса S для множества нечетких темпоральных событий, которые выражаются признаками a, b, c , определяется выражением

$$J(W/S)_I = J(a/\delta_a) \& J(b/\delta_b) \& J(c/\delta_c) = (F_a(\delta_a) \& \mu_{L_a}(\delta_a)) \& (F_b(\delta_b) \& \mu_{L_b}(\delta_b)) \& (F_c(\delta_c) \& \mu_{L_c}(\delta_c)).$$

В нашей случае динамический процесс – это последовательность кадров в векторной модели человека, представляющая повороты одного из узлов вокруг осей X, Y или Z , а в качестве критерия истинности рассматривается критерий сходства элементарных движений с эталонными.

Рассмотрим в качестве примера вращение узла «Правый голеностопный сустав». Вычислим значение критерия сходства фрагмента динамического процесса относительно следующего НТС: «Для правого голеностопного сустава в течение короткого промежутка времени наблюдается очень медленное уменьшение угла».

По графику функции принадлежности терма «Нулевая» лингвистической переменной «Продолжительность», приведенному на рисунке 3, определяется значение $F_r(\delta_r)=0,7$ при $\delta_r=t_2-t_0=2$ кадрам.

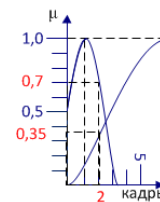


Рисунок 3 – Функция принадлежности терма «Нулевая» ЛП «Продолжительность»

Рассчитаем значение функции принадлежности $\mu_{Lq}(\rho_{ij})$ терма L_q «Очень медленное уменьшение» лингвистической переменной «Скорость изменения угла» при условии, что максимальная подвижность голеностопного сустава у спортсменов равна 79 градусам, а у людей, не занимающихся спортом, – 65 градусам. Выберем значение МП равное 72 градусам. Пусть в момент времени t_0 угол поворота ρ_0 составлял 10,00 градусов, а в момент времени t_2 угол поворота ρ_2 равнялся 6,13 градусов. Тогда $\Delta=72/18=4,00$; $\rho_{ij}= 6,13-10,00 = -3,87$; $\mu_{Lq}(\rho_{ij})=0,74$. Рассчитанное значение функции принадлежности изображено на рисунке 4.



Рисунок 4 – Значение функции принадлежности терма «Очень медленное уменьшение» ЛП «Скорость изменение угла»

Для полученных значений функций принадлежности критерий истинности равен 0,7. Имея набор НТС, выраженных через три последовательно проявляющихся признака, например, поворота вокруг оси X , Y или Z , можно рассчитать критерий истинности ПНТВ W относительно динамического процесса S .

При анализе выходных данных интерпретирующей модели необходимо учесть установленный порог достоверности полученных результатов, то есть значение, которое будет считаться границей корректно распознанных характерных телодвижений. Совокупности разбиений, для которых критерий истинности относительно анализируемого последовательного нечеткого темпорального высказывания окажется выше, чем порог достоверности, считается найденными корректно.

Описанная интерпретирующая модель способна работать в режиме реального времени [Розалиев и др., 2010]. При этом осуществляется динамическое формирование треугольных матриц разностей углов, принадлежащих разным кадрам в пределах одного узла. Через определённый квант времени, который зависит от скорости записи изображения, поступают новые векторные значения параметров, получаемых от Microsoft Kinect. В эти же кванты времени в системе осуществляется пересчет вторичных нечетких признаков, описывающих движение.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 11-07-00398, 12-07-00266, 12-07-00270).

Библиографический список

- [**Бобков и др., 2011**] Развитие системы автоматизированного определения эмоций и возможные сферы применения / Бобков А.С., Заболеева-Зотова А.В., Орлова Ю.А., Розалиев В.Л. // Открытое образование. - 2011. - № 2. - С. 59-62.
- [**Ковалев и др., 2011**] Идентификация дискретно-динамической системы с изменяющейся структурой в стохастической среде / Ковалев С.М., Соколов С.В. // Обзорные прикладной и промышленной математики. - Москва, 2011. - Т. 18, вып. 4. - С. 545-549.
- [**Розалиев и др., 2010**] Моделирование эмоционального состояния человека на основе гибридных методов / Розалиев В.Л., Заболеева-Зотова А.В. // Программные продукты и системы: международный науч.-практ. журнал. – Тверь, 2010 – Вып.2 (90). – С.141-146.
- [**Bobkov et al, 2012**] Fuzzy Approach to Identification of Human Emotions based on Recognition and Analysis of Body Movements / Bobkov A.S., Rosaliev V.L., Zaboloeva-Zotova A.V., Petrovsky A.B. // The 2nd World Conference on Soft Computing (December 3-5, 2012, Baku, Azerbaijan): proceedings / Ministry of Communications and Information Technologies of the Republic of Azerbaijan. – Baku, 2012. – P. 117-120. – Eng.
- [**Bernshtein et al, 2009**] Models of representation of fuzzy temporal knowledge in databases of temporal series. / Bernshtein L.S., Kovalev S.M., and Muravskii A.V. // Journal of Computer and Systems Sciences International, 2009, 48, 625-636.

MODELING HUMAN MOVEMENTS BASED ON THE FUZZY TEMPORAL STATEMENTS

Zaboloeva-Zotova A.V.* , Bobkov A.S.* ,
Petrovsky A.B.**

* *Volgograd State Technical University,
Volgograd, Russia*

***Institute for Systems Analysis,
Russian Academy of Sciences,
Moscow, Russia,*

zabzot@vstu.ru
bobkovart@gmail.com
pab@isa.ru

The paper considers an approach to the formalized description of human movements by using sequential fuzzy temporal statements.

In order to define body movements, first of all, we recognize a person on video images using a technology for markerless motion capture with the digital video camera Brekel Kinect. Video pictures are presented in the special animation format – the BVH-file, which describes poses of body skeleton and contains motion data. Such technology allows visualizing and analyzing different movements of a person, determining areas of static or dynamic postures of micro and macro movements.

We also use the vector model of skeleton, which consists of 22 nodes correspondent to different anatomical parts of body and joints with one, two or three axes of rotation. In the vector model of skeleton, the typical movements of human body are described with the linguistic variables, which characterize duration of event, and variation of rotation angle. These variables allow to determinate the direction and speed of movement of every joint.

The movement of the joint around an axis has been described in the form of fuzzy temporal events. Since the events are located one after another on the time axis, the motion can be represented as a sequential fuzzy temporal sentence. In the model of sequential fuzzy temporal sentence, an adequacy of the analyzed fragment of a dynamic process and the corresponding attribute are determined by the validity criterion.

This work was partially supported by the Russian Foundation for Basic Research (projects 11-07-00398, 12-07-00266, 12-07-00270).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

РАСПОЗНАВАНИЕ ПОЗЫ ТЕЛА ЧЕЛОВЕКА С ПОМОЩЬЮ 3D КАМЕРЫ С СЕНСОРОМ ГЛУБИНЫ

Шпирко А.А., Дорофеев Н.С., Бобков А.С.

*Волгоградский государственный технический университет,
г. Волгоград, Российская Федерация*

alex555@yandex.ru

nikita.dorofei@gmail.com

bobkovart@gmail.com

В работе описан метод контроля правильности выполнения физических упражнений лечебной физкультуры с помощью технологии Microsoft Kinect. Формализован процесс идентификации позы тела человека по координатам некоторых «характерных» узлов тела человека с использованием продукционных правил. С помощью нейронной сети – многослойного персептрона происходит классификация упражнений на правильные и неправильные. Обучение нейронной сети происходило на выборке, составленной экспертом в области ЛФК.

Ключевые слова: упражнения ЛФК; продукционные правила; нейронная сеть; контроль правильности выполнения упражнений.

ВВЕДЕНИЕ

Лечебная физическая культура (ЛФК) – это метод лечения, состоящий в применении физических упражнений и естественных факторов природы к больному человеку с лечебно-профилактическими целями. В основе этого метода лежит использование биологической функции организма – движения [Medicinform, 2012].

До недавнего времени обеспечить контроль правильности выполнения физических упражнений мог только персональный тренер или врач-специалист по ЛФК. Самостоятельное выполнение упражнений возможно с использованием специализированной литературы или мультимедийных дисков, с записанными примерами выполнения упражнений. Но в обоих случаях, невозможно проконтролировать правильность выполнения человеком упражнений.

Применение сенсора Microsoft Kinect [Орлова и др., 2011], в разрабатываемой системе, позволит сделать процесс занятий удобным и безопасным. Также разработанные модели и методы применяются в работах по распознаванию эмоциональных реакций человека [Заболеева-Зотова и др., 2011], [Розалиев и др., 2011a], [Розалиев и др., 2011b].

В статье описан метод контроля правильности выполнения физических упражнений (ФУ) лечебной физкультуры с помощью технологии Microsoft Kinect.

1. Распознавание движения

Любое физическое упражнения – это движение. Для контроля правильности выполнения ФУ необходимо распознать движение человека.

Выполняемое ФУ снимается безмаркерным способом с помощью сенсора Kinect (3-D камера от Microsoft с сенсором глубины), далее используется система Brekel Kinect, предоставляющая распознанное движение в формате. bvh – файла.

BVH обозначает данные Bio Vision Hierarchical. Этот формат предоставляет возможность представления информации об иерархии каркаса тела человека в добавление к данным о движении [Kinecthacks, 2012].

```

HIERARCHY
ROOT_HIPS
{
  OFFSET 0.0000 0.0000 0.0000
  CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
  JOINT Chest1
  {
    OFFSET 0.0415 6.2451 0.0203
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT Chest2
    {
      OFFSET 0.0118 22.5192 -0.7792
      CHANNELS 3 Zrotation Xrotation Yrotation
      JOINT Leftcollar
      {
        OFFSET -0.3300 25.0202 0.8579
        CHANNELS 3 Zrotation Xrotation Yrotation
        JOINT Leftshoulder
        {
          OFFSET 15.6466 -3.5590 0.4633
          CHANNELS 3 Zrotation Xrotation Yrotation
          JOINT Leftelbow
          {
            OFFSET 6.7754 -28.9612 2.1049
            CHANNELS 3 Zrotation Xrotation Yrotation
            JOINT Leftwrist
            {
              OFFSET 5.9027 -23.6186 1.7304
              CHANNELS 3 Zrotation Xrotation Yrotation
              End Site
              {
                OFFSET 3.1558 -12.6376 1.4914
              }
            }
          }
        }
      }
    }
  }
}
JOINT Rightcollar
{
  OFFSET -0.3300 25.0202 0.8579
  CHANNELS 3 Zrotation Xrotation Yrotation
  JOINT Rightshoulder
}

```

Рисунок 1- Структура BVH-файла

2. Идентификация позы человека и физического упражнения

Каждое ФУ характеризуется набором поз тела человека. Каждая поза характеризуется набором координат узлов (суставов) относительно центра тела.

ФУ ЛФК характеризуется набором т.н. «характерных» узлов человеческого тела. Это узлы, на которые направлено его воздействие и взаиморасположение которых в процессе выполнения определяет правильность выполнения упражнения

Идентификация ФУ и характерных узлов человеческого тела, реализована с помощью продукционной модели вида:

$$pm = \langle S, L, A \rightarrow B, C, Q \rangle,$$

Где $S = \{S_1, S_2, S_3\}$ – множество зон человеческого тела (верхняя, нижняя, центральная); L - условие применимости ядра продукции. Сигнал о том, что было выбрано упражнение.

$A \rightarrow B, C$ - ядро продукции, интерпретируется фразой:

«Если A , то B и C »,

где $A = \{a_1, a_2, a_3, \dots, a_n\}$ – множество физических упражнений;

$B = \{\{b_1, b_2, \dots, b_n\}_1, \{b_1, b_2, \dots, b_n\}_2, \dots, \{b_1, b_2, \dots, b_n\}_m\}$ – множество, включающие себя подмножества из набора характерных узлов (суставов) человеческого тела b , где b это вектор – строка вида (b_x, b_y, b_z) .

$C = \{c_1, c_2, \dots, c_n\}$ – множество нейронных сетей, каждая из которых настроена для определения правильности конкретного упражнения.

3. Метод проверки правильности выполнения физических упражнений

Для контроля правильности и эффективности выполнения физического упражнения, необходимо определить, какие действия при выполнении упражнения будут считаться правильными, а какие нет. Эта задача ложится на эксперта в области ЛФК. На основании мнения эксперта и основных правил лечебной физкультуры, с помощью системы распознавания движения Brekel Kinect были

записаны упражнения, которые будут считаться правильными. Также были записаны движения, представляющие собой наиболее распространенное неправильное выполнение упражнений.

Под упражнением в данном контексте понимается набор поз которые последовательно принимает человек и перед системой стоит задача определить к какому классу относится в текущий момент времени поза, которую принимает человек: к классу «правильное выполнение» или к классу «неправильное выполнение».

Обучающая выборка состояла из 2000 поз на каждый класс, где каждая поза характеризовалась набором «характерных» узлов для каждого физического упражнения.

Было протестировано десять различных методов, способных решать задачи классификации, на одних и тех же входных данных. По окончании, метод, показавший наилучший результат, был использован для решения задачи классификации правильности выполнения ФУ.

Таблица 1 - Сравнение эффективности методов решения задачи идентификации

Метод	Правильно Идентифицированные образы(%)
Дерево решений: CHAID алгоритм (автоматический обнаружитель взаимодействий)	62,58
Дерево решений: C&RT алгоритм	69,99
Дерево решений: boosting trees алгоритм («растущие деревья»)	81,74
Дерево решений: random forest (алгоритм «случайные леса»)	76,18
Метод опорных векторов (Support Vector Machine)	76,25
Метод К-ближайших соседей (K-Nearest neighbors)	82,11
Байесовский классификатор (Naive Bayes Classifiers)	87,36
Нейронная сеть: вероятностная сеть	94,12
Нейронная сеть: Многослойный перцептрон	96,02
Нейронная сеть: радиально-базисных функций	68,20

3. Характеристики нейронной сети

Наилучший результат показала нейронная сеть многослойный персептрон со следующими параметрами:

Тип НС	%успешной классификации и всей выборки	%успешной классификации и тестовой выборки	%успешной классификации и контрольная выборки	Алгоритм обучения	Функция ошибки	Функция активации скрытого слоя	Функция активации внешнего слоя
МРП18-12-2	96,0231	96,0001	95,9887	BFGS5	Кросс-энтропия	Экспоненциальная	Софтмакс

Рисунок 2 – Параметры нейронной сети

Алгоритм обучения – BFGS. В данном алгоритме используется BFGS-схема обновления, названная по первым буквам имен Broyden-Fletcher-Goldfarb-Shanno (если точнее, то эта формула строит не сам гессиан, а обратную к нему матрицу; таким образом, не надо тратить время на её обращение). В случае больших размерностей объем памяти порядка N^2 , требуемый для хранения гессиана, оказывается слишком большой нагрузкой, также как и затраты машинного времени на его обработку. Поэтому вместо использования N значений градиента для построения гессиана можно обойтись меньшим числом значений, позволяющим использовать объем памяти порядка $N \cdot M$. Обычно на практике M выбирают в промежутке от 3 до 7, в сложных случаях можно увеличить эту константу до 20. В результате такой экономии получается не сам гессиан, а лишь его аппроксимация. Изначально этот метод разрабатывался для оптимизации функций очень большого числа аргументов (сотни и тысячи), поскольку именно в этом случае увеличение числа итераций из-за пониженной точности аппроксимации гессиана полностью окуается снижением накладных расходов на обновление гессиана, однако нет причин, по котором этот метод нельзя применять для задач малой размерности. Основным его достоинством является масштабируемость, поскольку он обеспечивает высокое быстродействие на задачах большой размерности, при этом позволяя решать и задачи малой размерности [Осовский, 2007].

Функция ошибки – кросс-энтропия. Функция, основанная на теоретико-информационных характеристиках. Особенно хорошо подходит для задач классификации. Имеется два варианта: для сетей с одним выходом и для сетей с несколькими выходами; в первом варианте используются логистические функции активации, во втором – функции софтмакс [Боровиков, 2007].

Функция активации внешнего слоя Софтмакс. Это функция активации, специально предназначенная для классификационных сетей с кодированием по методу один-из- N . Вычисляет нормированную экспоненту (т.е. сумма выходов равна единице). В сочетании с кросс-энтропийной функцией ошибок позволяет модифицировать многослойный персептрон для оценки вероятностей принадлежности классам [Хайкин, 2006].

4. Пример работы системы

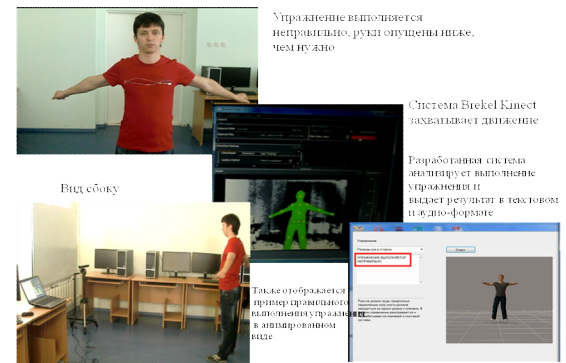


Рисунок 3 – Пример работы системы. Упражнение выполняется неправильно

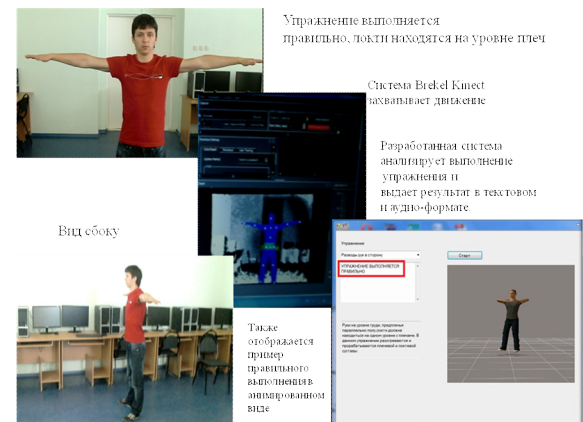


Рисунок 4 – Пример работы системы. Упражнение выполняется неправильно

На рисунках представлены примеры выполнения правильного и неправильного выполнения ФУ. Распознанное движение сохраняется в BVH - формат, подается на вход нейронной сети, которая уже возвращает результат – заключение правильно или неправильно выполняется то или иное ФУ. Одновременно пользователю показывается анимированный пример правильного выполнения и звуковое сопровождение.

Заключение

В результате работы была автоматизирована проверка правильности выполнения ФУ, разработаны правила идентификации позы человека и ФУ, построена нейронная сеть для проверки правильности выполнения ФУ.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 12-07-00266, 12-07-00270).

Библиографический список

[Орлова и др., 2011] Орлова, Ю.А. Обзор современных автоматизированных систем распознавания эмоциональных реакций человека / Ю.А. Орлова, В.Л. Розалиев // Изв. ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах". Вып. 10 : межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2011. - № 3. - С. 68-72.

[Medicinform, 2012] Основы ЛФК / Медицинская информационная сеть. – [2012]. – Электронный ресурс. Режим доступа: <http://www.medicinform.net/fizio/lfk/>

[Kinethacks, 2012] Применение Kinect на PC /Kinethacks. – [2012]. – Электронный ресурс. Режим доступа: <http://kinethacks.net/>

[Заболеева-Зотова и др., 2011] Развитие системы автоматизированного определения эмоций и возможные сферы применения / А.В. Заболеева-Зотова, Ю.А. Орлова, В.Л. Розалиев, А.С. Бобков // Открытое образование. - 2011. - № 2. - С. 59-62.

[Розалиев и др., 2011a] Розалиев, В.Л. Моделирование эмоционального состояния человека на основе гибридных методов / В.Л. Розалиев, А.В. Заболеева-Зотова // Программные продукты и системы: международный науч.-практ. журнал. – Тверь, 2010 – Вып.2 (90). – С.141-146.

[Розалиев и др., 2011b] Розалиев, В.Л. Применение нейронных сетей и грануляции при построении автоматизированной системы определения эмоциональных реакций человека / В.Л. Розалиев, А.С. Бобков, О.С. Федоров // Изв. ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах»: межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2010. - Вып. 9, № 11. - С. 63-68.

[Хайкин, 2006] Хайкин, С. Нейронные сети: полный курс / С. Хайкин. - М.: Филинь, 2006. - 122 с

[Осовский, 2007] Осовский, С. Нейронные сети для обработки информации /С. Осовский. - М.: Финансы и статистика, 2007.- 344 с.

[Боровиков, 2007] Боровиков, В. П. Нейронные сети STATISTICA Neural Networks: Методология и технология современного анализа данных / В. П. Боровиков. - М. : Статсофт, 2007. - 632 с.

RECOGNITION OF BODY POSTURES USING A 3D CAMERA WITH THE DEPTH SENSOR

Shpirko A.A., Dorofeev N.S., Bobkov A.S.

*Volgograd State Technical University, Volgograd,
Russian Federation*

alexs55@yandex.ru

nikita.dorofei@gmail.com

bobkovart@gmail.com

A new method of controlling correctness of performing physical exercises of physical therapy with the help of 3D camera with depth sensor is described. Process of identifying human body postures by coordinates of some "specific" nodes of the human body using production rules was formalized. With the help of neural networks - multilayer perceptron exercises are classified on right and wrong. Neural network was trained using samples prepared by the experts in the field of physical therapy.

INTRODUCTION

Relevance of the work by the fact that to date the only method by which to correctly perform the person exercising physical therapy can be controlled - it is a

personal trainer or a doctor - a specialist gymnastics. With self-perform these exercises, it is likely that people will not carry them out correctly, because not all people who need exercise therapy sessions have sufficient knowledge in this area. While automating the control of correctness, the system checks the correctness of the exercises, provides tips on technique performance visually and in text form.

MAIN PART

Any physical exercise represents motion. In order to work with any motion, first of all, this motion should be captured. 3D sensor from Microsoft, named Kinect, captures the motion, then motion data is recognized and transmitted into BVH file. BVH means "Bio Vision Hierarchical" and represents a special format where human body joints and their positions are described in human-readable numbers.

Also any physical exercise can be characterized by a set of human's poses, which, in their turn represent a set of "specified" body joints. And the correctness of performing an exercise depends on these joints interposition. Information about joint positions is gathered from BVH file, which is parsed in real-time. Also in real-time information from sensor is written in it. Besides Cartesian coordinates, this file contains angles of rotations of joints relative to each other.

A neural network, trained with the help of medical experts classifies poses, which a person take while performing an exercise, and makes a decision while he is doing wrong or right. The training set for neural network contained several thousands of samples of exercises both in correct and in incorrect type. Several methods, which can solve classification task where tested and the best results showed multilayer perceptron.

CONCLUSION

As a result of work, the rules identifying human posture and exercise have been developed, neural network to test the correctness of the exercise, was constructed and thus, control of correctness of performing physical exercises was automated.

This work was partially supported by the Russian Foundation for Basic Research (projects 12-07-00266, 12-07-00270).



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 621.391

СПРАВОЧНАЯ СИСТЕМА С РЕЧЕВЫМ ИНТЕРФЕЙСОМ

Житко В.А.*, Гецевич Ю.С.***, Лобанов Б.М.**

* *Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

zhitko.vladimir@gmail.com

** *Объединённый институт проблем информатики НАН Беларуси, Минск*

Yury.Hetsevich@gmail.com

lobanov@newman.bas-net.by

Описывается модель справочной системы по правилам дорожного движения РБ с речевым пользовательским интерфейсом. Для распознавания речи используется интернет сервис Google по распознаванию речи. Разработан дополнительный компонент распознавания речи. Внесенные доработки позволили распознавать поток непрерывной речи. Анализ вопросов и генерация ответов осуществляется с помощью настраиваемого лингвистического процессора NooJ. Синтез речи по тексту осуществляется с помощью разработанной системы «МультиФон» на белорусском или русском языках.

Ключевые слова: речевой интерфейс; синтез речи; распознавание речи, справочные системы; анализ текстов.

ВВЕДЕНИЕ

В связи с бурным ростом рынка мобильных и встроенных приложений разработкам в сфере упрощения ввода запроса информации уделяется большое внимание. Наиболее перспективными в этой ситуации являются разработки, связанные с распознаванием и синтезом речи, – современной альтернативе ручному вводу с клавиатуры и графическому интерфейсу. Использование речевого ввода команд и вопросов позволяет запрашивать справочную информацию одновременно с выполнением основной деятельности (например, при управлении автомобилем), а использование при этом речевого синтеза позволяет также разгрузить зрительный канал восприятия информации.

Следует отметить, что для естественно-языкового интерфейса конкретной прикладной системы возможно использование ограниченной лексики и грамматики языка без серьёзного ущерба функциональности вопросно-ответной системы. Ограниченный естественный язык – это подмножество естественного языка, текст на котором без каких-либо усилий воспринимается носителем исходного естественного языка, а также не требует длительного изучения для приобретения навыков составления текстов на этом языке. Это позволяет снизить время обработки естественно-

языковых конструкций в системе и частично избежать лингвистических неоднозначностей.

1. Справочная система по правилам дорожного движения

В настоящее время существует ряд справочных систем обладающих речевым интерфейсом (Siri, Voice Actions и др.), однако все они являются закрытыми коммерческими проектами и использовать их в других проектах не представляется возможным.

Целью данного проекта является разработка моделей и средств построения естественно-языкового интерфейса конкретной прикладной системы. Выбранный компонентный подход к построению пользовательского интерфейса позволит в дальнейшем свести разработку новых прикладных систем к настройке и адаптации готовых компонентов к конкретной предметной области.

В качестве предметной области для проекта были выбраны реальные правила дорожного движения (ПДД), действующие на территории РБ. Отметим, что набор ПДД характеризуются структурированностью исходных данных, что существенно облегчает решение поставленной задачи. В процессе взаимодействия с пользователем справочная система в качестве речевого ответа

производит выбор текста наиболее релевантного ПДД или задает уточняющие вопросы).

1.1. Обработка данных

Комплекс ПДД представлен в виде набора отдельных правил с дополнительным указанием ключевых слов для каждого правила. Цикл работы справочной системы начинается с ввода пользователем голосового (речевого) запроса на естественном языке. Используемый в системе компонент распознавания речи преобразует произнесённый запрос в набор наиболее вероятных текстовых сообщений. По введённому в систему запросу строится его формальное описание в памяти системы. Все предшествующие результаты анализа используются при анализе последующих запросов, что позволяет системе сохранять ход диалога с пользователем и задавать уточняющие вопросы.

Первым этапом анализа запроса пользователя является его морфологический анализ. На данном этапе для каждого слова входной фразы ставится в соответствие ее начальная форма. Для данного шага используется лингвистическая система NooJ [Silberztein, 2003].

NooJ является свободно распространяемым продуктом для формализации лингвистических данных. Система включает в себя морфологический и синтаксический анализатор, а также удобные средства для разметки корпуса вручную. В NooJ встроена система визуального написания грамматик, которая позволяет создавать различные системы анализа текста (например, для задачи фактографического поиска).

В настоящее время системой NooJ поддерживается ряд языков: арабский, армянский, болгарский, каталанский, китайский, хорватский, английский, французский, немецкий, иврит, венгерский, итальянский, польский, португальский, испанский язык, русский и белорусский языки. Ведутся работы по поддержке и других языков.

Лингвистический процессор NooJ включает несколько вычислительных средств, используемых для формализации лингвистических данных и разбора текстов.

Finite-State Transducer (FST) - граф, описывающий ряд последовательностей языковых единиц, в котором каждая такая последовательность связана некоторым аналитическим результатом. Последовательности языковых единиц описаны во входной части FST; соответствующие результаты описаны в части продукции FST.

Как правило, синтаксические FST представляют собой последовательности слов, и производят добавление лингвистической информации (такой как синтаксическая структура). Морфологические FST представляют последовательности букв, которые описывают словоформу, и затем дополняют лексическую информацию (такую как часть речи,

ряд морфологических, синтаксических и семантических меток).

В NooJ конечные автоматы (FSA), как правило, используются, чтобы определить местонахождение морфо-синтаксических шаблонов в корпусах и извлечь последовательности соответствий, чтобы построить индексы, соответствия.

Рекурсивные Сети Перехода (RTNs) - грамматики, которые содержат больше чем один граф: графы могут быть FST или FSA, а также включать ссылки на другие, включенные графы, эти графы могут в свою очередь содержать другие ссылки к тому же самому, или к другим графам. Вообще, RTNs используются в NooJ, чтобы построить библиотеки повторно используемых грамматик: разработанные простые грамматики могут быть снова использованы в более общих грамматиках, в свою очередь эти грамматики снова могут быть использованы и т.д.

Расширенные Рекурсивные Сети Перехода (ERTNs) - RTNs, которые содержат переменные, эти переменные как правило хранят части последовательностей, и затем используются в некоторых операциях (например, преобразуют их содержимое в множественное число, и т.д.), и затем уже используется при построении результатов.

NooJ включает флективный/деривационный модуль, который связан со словарями, так, чтобы он мог автоматически связывать словарные статьи с соответствующими формами, которые используются в грамматиках (эта функциональность позволяет избавляться от полных словарей формы INTEX, таких как DELAF и DELACFs).

Словари NooJ связывают каждую лексическую статью с флективной и/или деривационной парадигмой. Например, все глаголы, которые спрягаются как "aimer", связаны с парадигмой "+FLX=AIMER"; все глаголы, которые принимают "-able" суффикс, связаны с парадигмой "+DRV=ABLE" и т.д.

Парадигмы, такие как "AIMER" или "ABBLE" описаны или графически в RTNs или в текстовых файлах.

Все это позволяет использовать NooJ как на первом этапе морфологического разбора, так и на последующем при поиске релевантного правила ПДД.

Перспективным так же выглядит использование грамматик NooJ для анализа пользовательских запросов с целью их классификации и выделения предмета вопроса. Классификация вопросов позволит учитывать особенности классов вопросов при поиске релевантных ответов.

Пример грамматики описывающий один из классов вопросов приведен на рисунке 1. Такой класс вопросов может быть использован при следующем диалоге:

- Что такое автомобиль? (выделяются следующие ключевые слова: автомобиль)

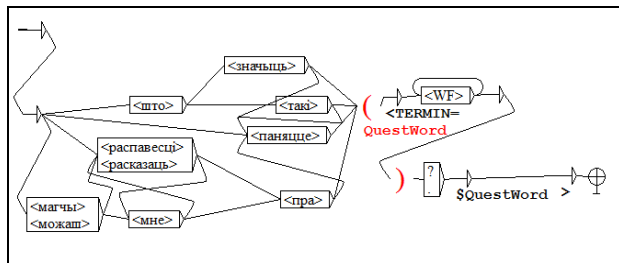
- Автомобиль — механическое транспортное средство, имеющее не менее четырех колес, расположенных не менее чем на двух осях, за исключением колесных тракторов и самоходных машин. К автомобилям приравниваются квадрициклы (мотоколяски) — четырехколесные механические транспортные средства, имеющие приводы управления автомобильного типа и массу в снаряженном состоянии не более 550 килограммов;

- А есть ли двухколесное транспортное средство? (выделяются следующие ключевые слова: транспортное средство)

- Мотоцикл — двухколесное механическое транспортное средство с боковым прицепом или без него, приводимое в движение двигателем с рабочим объемом более 50 куб. сантиметров. К мотоциклам приравниваются трехколесные механические транспортные средства, имеющие массу в снаряженном состоянии не более 400 килограммов, а также механические транспортные средства, оборудованные двигателем с рабочим объемом до 50 куб. сантиметров, имеющие максимальную конструктивную скорость движения, определенную их технической характеристикой, более 50 км/ч;

- А что такое прицеп? (выделяются следующие ключевые слова: прицеп)

- Прицеп — транспортное средство, предназначенное для движения в составе с механическим транспортным средством;



а)

Seq.
Раскажы мне пра аўтапоезд./<TERMIN=аўтапоезд>
Што такое аўтамабіль?/<TERMIN=аўтамабіль>
Што такое паняцце мапед?/<TERMIN=мапед>
Што такое паняцце мапед?/<TERMIN=паняцце мапед>
Што значыць уступіць дарогу?/<TERMIN=уступіць дарогу>
можаш расказаць мне пра жылую зону?/<TERMIN=жылую зону>

б)

Рисунок 1 – Пример анализа возможных вопросов к справочной системе: а) синтаксическая грамматика для поиска предмета вопроса; б) результат поиска предмета вопроса.

На последнем этапе – этапе поиска релевантной группы ПДД – рассчитываются коэффициенты соотношения входной фразы и группы ПДД. Далее операция повторяется на множестве ПДД из выбранных групп. Если по результату поиска было найдено множество ПДД с равными коэффициентами, система строит множество

ключевых слов позволяющих сделать окончательный выбор правила и просит пользователя уточнить вопрос или выбрать ключевое слово из списка. Ответ пользователя анализируется аналогичным образом, но уже с использованием прошлого результата.

Пример диалога пользователя и системы с уточняющим вопросом:

- Должен ли я уступать дорогу на перекрестке? (выделяются следующие ключевые слова: дорога, уступать, перекресток)

- Уточните вопрос или выберете одно из следующих ключевых слов: неравнозначные дороги, меняет направление, равнозначные дороги.

- Перекресток с равнозначными дорогами. (выделяются дополнительные ключевые слова: равнозначные, дорога)

- На перекрестке равнозначных дорог водитель транспортного средства, кроме трамвая, обязан уступить дорогу транспортным средствам, приближающимся справа. Этим же правилом должны руководствоваться между собой водители трамваев. На таких перекрестках водитель трамвая имеет преимущество перед другими транспортными средствами (кроме трамваев) независимо от направления его движения.

Пример диалога пользователя и системы с использованием словаря синонимов:

- Как мне развернуться? (выделяются следующие ключевые слова: развернуться (вводится синоним "разворот"))

- При повороте налево или развороте водитель транспортного средства обязан уступить дорогу встречным транспортным средствам, движущимся по равнозначной дороге прямо или направо, и попутному трамваю. Этим же правилом должны руководствоваться между собой водители трамваев. При одновременном праве на движение водители трамваев имеют преимущество перед другими транспортными средствами (кроме трамваев) независимо от направления их движения.

- Как мне развернуться на светофоре? (выделяются следующие ключевые слова: развернуться (вводится синоним "разворот"), светофор)

- При повороте налево или развороте по зеленому сигналу светофора водитель транспортного средства, за исключением трамвая, обязан уступить дорогу встречным транспортным средствам, движущимся прямо или направо, и попутному трамваю. Таким же правилом должны руководствоваться между собой водители трамваев.

1.2. Синтез речи

Для перевода текстовой информации в речевую используется два компонента, основанных на

синтезаторах речи по тексту на белорусском [Гецевич и др., 2010] и русском [Лобанов и др., 2008] языках.

Синтез устной речи по тексту осуществляется на основе лексико-грамматического анализа входного текста путём моделирования процессов речеобразования с учётом правил произношения звуков и интонирования для каждого языка. Орфографический текст поступает на вход синтезатора и далее подвергается последовательной обработке рядом специализированных процессоров в соответствии с общей структурой синтезатора речи по тексту, представленной на рисунке 2. Синтезатор включает модули: текстовый процессор, просодический процессор текста и сигнала, фонетический процессор и акустический процессор. Каждый из этих модулей поддерживается наборами соответствующих баз данных и правил.

Главный модуль лингвистического процессора – текстовый процессор – управляет другими его модулями и контролирует процесс преобразования в них текста в последовательность синтагм. Процессор слов определяет возможные лексико-грамматические характеристики слова (последовательности символов, отделенных пробелами и знаками препинания).

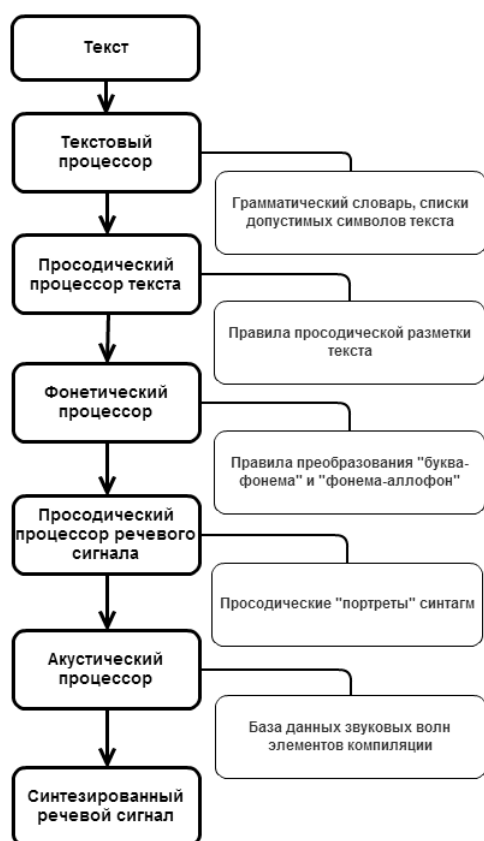


Рисунок 2 – Общая структура синтезатора

Лексико-грамматический процессор определяет лексико-грамматические характеристики слова на основе вариантов, предложенных предыдущим процессором и лексико-грамматических характеристик других слов в тексте. Дополнительные плагины производят обработку специальных выражений (например, чисел,

сокращений, дат, времени и др.) в орфографические слова. Процессор выражений расставляет ударения в словах, создаёт фонетические слова через присоединение к словам предлогов и частиц, заменяет конкретные выражения результатами обработки из плагинов. Процессор сборки словосочетаний соединяет отдельные слова в группы, исходя из их лексико-грамматических характеристик. Процессор создания синтагм разделяет полученные словосочетания на возможные синтагмы, с указанием интонационного типа синтагмы в зависимости от знаков препинания и лексико-грамматических характеристик слов.

Фонетический процессор производит преобразование последовательности букв, из которых состоит синтагма в последовательность фонем.

Просодический процессор производит определение просодических характеристик (частоты основного тона, длительности, амплитуды сигнала) для каждой фонемы в последовательности, исходя из интонационного контура, определяемого типом синтагмы.

Акустический процессор соединяет аллофоны, определяемые фонемами, изменяет просодические параметры аллофонов, формирует звуковой сигнал. Контроллер преобразования звуковых форматов управляет плагинами, преобразующими звук. Такие плагины производят преобразование звука в различные форматы (изменение частоты дискретизации, точности передачи звука, упаковка в формат MP3).

Таким образом, спроектированная архитектура позволяет разработать качественно новый синтезатор речи по тексту для русской и белорусской речи с высокой степенью «лингвистического понимания» входного текста и генерацией речи для самого широкого круга потребителей.

1.3. Распознавание речи

Для распознавания речевого сигнала в системе используется компонент, в основе которого лежит сервис распознавания речи, разработанный компанией Google.

В настоящее время компания Google является лидером по предоставлению облачных технологий распознавания речи [Manjoo F., 2011]. В течение последних пяти лет активно развивалась облачная технология распознавания речи Google Voice, и к настоящему времени существуют технологии распознавания речи для большинства европейских языков, включая русский, японский и китайский языки. Одним из немаловажных компонентов системы распознавания речи Google Voice является обучающая выборка звукозаписей человеческого голоса. Для системы Google Voice источником таких записей являются различные сервисы, предоставляемые Google, использующие речевые технологии, к ним относятся система распознавания

речи и команд в системе Android, сервис диктовки писем Google Mail, телефонная справочная система Goog411 и др. [Singhal, A., 2011]. Таким образом обучающая выборка постоянно пополняется новыми образцами голосов с их особенностями как произношения, так и эффектами, вносимыми техническими особенностями записи и передачи голоса на различных устройствах. К примеру, в 2011 г. обучающая выборка для английского языка составляла примерно 230 млрд записей слов извлеченных из реальных запросов [Enge, E. 2011]. Для обработки таких объемов информации требуется около 70 лет процессорного времени, однако с использованием облачной технологии Google время сокращается до одного дня [Singhal, A., 2011].

Для того, чтобы использовать систему распознавания речи Google Voice в рассматриваемом прикладном проекте, разработан дополнительный компонент распознавания речи. В разработанном компоненте распознавания речи выделены следующие программные блоки:

- детектор речи;
- блок сжатия и обработки речевого сигнала;
- блок сопряжения с системой Google Voice;

Детектор речи является необходимым дополнительным компонентом, обусловленным используемой архитектурой удаленного распознавания речи. В этом случае канал передачи данных (в нашем случае интернет-соединение) является «бутылочным горлышком», ограничивающим максимально возможный объем передачи данных в заданный промежуток времени. При передаче по каналу слишком длинного отрезка речевого сигнала происходит недопустимо длительная задержка ответной реакции системы распознавания. Кроме того, при этом возникает значительный риск получения ошибочных результатов распознавания. Ввод коротких отрезков речи возможен при использовании ручного стартстопного режима, предоставляемого системой Google Voice. Однако для решения поставленной здесь задачи стенографирования устной речи этот режим оказывается крайне неудобным. Обычно диктовка осуществляется короткими фразами, заканчивающимися паузами. После произнесения каждой из них пользователь, как правило, желает убедиться в правильности распознавания и при необходимости повторить ее более четко.

Задача детектора речи заключается в автоматической локализации полезного сигнала, т. е. в определении начала и конца произнесенной фразы. Это обеспечивает автоматический пофразовый ввод речи. В экспериментальной программной системе использован относительно простой, но достаточно эффективный алгоритм определения полезного потока речи, основанный на расчете усредненной текущей энергии звуковой волны входного сигнала. Перед началом работы в течение одной секунды производится замер

фонового шума для определения порога срабатывания детектора речи. Порог вычисляется как произведение средней энергии фонового шума на коэффициент, заданный в настройках программы (по умолчанию равный 2,0).

Далее ведется запись в кольцевой буфер задержки сигнала (по умолчанию задержка равна 1,0 с) и одновременно вычисляется текущая средняя энергия звуковой волны входного сигнала (по умолчанию время усреднения равно 0,4 с). Если порог будет превышен, задержанный входной сигнал передается на вход компонента сжатия и обработки сигнала, а передача сигнала будет вестись до тех пор, пока значение средней энергии не станет меньше значения порога.

Полученный полезный сигнал в модуле сжатия и обработки сигнала подготавливается для отправки на удаленный сервер распознавания речи Google Voice. Для этого сигнал кодируется в открытом формате FLAC (Free Lossless Audio Codec) с заданными характеристиками (частота дискретизации 16кГц, моно). При этом происходит сжатие сигнала кодеком, что уменьшает объем передаваемых данных и, как следствие, сокращает время ожидания результата распознавания. Использование open-source-кодека имеет и другие преимущества: простота использования сторонними разработчиками, единообразие получаемого сервером формата данных. Каждый компонент программы работает асинхронно и имеет свои пулы данных для нивелирования эффектов, связанных с задержками в работе каждого компонента. Такие задержки появляются как на этапе записи и кодирования речевого сигнала в файл, так и отправки его на удаленный сервер.

На рисунке 3 описанный алгоритм более подробно представлен в графическом виде.

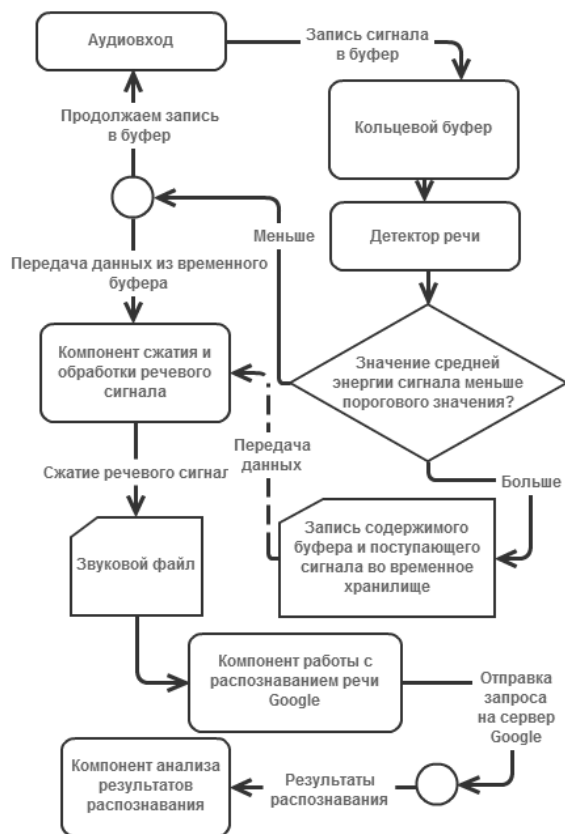


Рисунок 3 – Схема работы компонента распознавания речи

Заключение

Представленная модель справочной системы с разбиением на отдельные компоненты со строго заданным функционалом, позволяет упростить разработку, а в дальнейшем – сопровождение, различных систем с речевым интерфейсом.

Разбиение системы на отдельные независимые компоненты дает возможность интеграции сторонних разработок и проектов и производить интеграцию различных подходов и методов в рамках одного проекта, что позволяет эффективно использовать их лучшие стороны.

Компоненты системы распознавания и синтеза речи по тексту предоставляют конечному пользователю возможность устно задавать вопрос и слышать ответ на него от системы, а не просто вводить вопрос через клавиатуру и читать ответ с экрана компьютера. Это делает языковой интерфейс еще более естественным для пользователя.

Библиографический список

- [Гецевич и др., 2010] Гецевич, Ю.С. Система синтеза белорусской речи по тексту / Ю.С. Гецевич, Б.М. Лобанов. Речевые технологии. – 2010. – № 1. – С. 91-100.
- [Лобанов и др., 2008] Лобанов Б.М., Компьютерный синтез и клонирование речи / Лобанов Б.М., Цирульчик Л.И. Минск: Белорусская наука, 2008. – 344 с.: ил.
- [Manjoo F., 2011] Manjoo, F. Now You're Talking / Farhad Manjoo // The Slate Group. – Washington Post Compa-ny, 2012. – [Electronic resource] – Mode of access : http://www.slate.com/articles/technology/technology/2011/04/now_youre_talking.single.html. – Date of access : 01.08.2012.

[Singhal, A., 2011] Singhal, A. Knocking down barriers to knowledge / Amit Singhal // Google Official Blog [Electronic resource]. – 2011. – Mode of access : <http://googleblog.blogspot.com/2011/06/knocking-down-barriers-to-knowledge.html>. – Date of access : 01.08.2012.

[Enge, E. 2011] Enge, E. Search Algorithms with Google Director of Research Peter Norvig / E. Enge // Stone Temple Consulting [Electronic resource]. – 2011. – Mode of access : <http://www.stonetemple.com/search-algorithms-with-google-director-of-research-peter-norvig>. – Date of access : 01.08.2012.

[Silberztein, 2003] Silberztein, M. NooJ Manual [Electronic resource]. – 2003. – Mode of access : <http://www.nooj4nlp.net/NooJManual.pdf>. – Date of access : 01.07.2012.

HELP SYSTEM WITH SPEECH USER INTERFACE

V.A. Zhitko*, Y.S. Hetsevich**,
B.M. Lobanov**

* Belarusian State University of Informatics and
Radioelectronics, Minsk, Republic of Belarus

zhitko.vladimir@gmail.com

** United Institute of Informatics Problems of the
National Academy of Sciences of Belarus, Minsk,
Republic of Belarus

Yury.Hetsevich@gmail.com

lobanov@newman.bas-net.by

This work describes a model of help system with speech user interface for highway regulations. Also describes components for text-to-speech, question-answering and voice recognition. Primary goal of this work is making easy building speech user interfaces.

MAIN PART

For analyzing user questions and generating answers used linguistic processor NooJ.

Another important component is voice output. This component is top required for making natural language interface friendlier for users. Also, as previous component, it built in a traditional technology. But some tasks are hard to solve in traditional ways, for example correct detect the stress in words, sometimes for this need understand meaning of text content, and this could be solve by semantic approaches.

Additional third-party component is natural Russian voice input. For this component used Google voice recognition service with made some changes. So component can recognize non-stop user speech.

CONCLUSION

In the given paper short description of model and methods for designing and prototyping natural language interfaces with voice input, question-answering processing and output Belarusian or Russian languages.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

КОМПОНЕНТНОЕ ПРОЕКТИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ ДЛЯ ПОДГОТОВКИ ШКОЛЬНИКОВ К ЦЕНТРАЛИЗОВАННОМУ ТЕСТИРОВАНИЮ ПО ИНОСТРАННЫМ ЯЗЫКАМ

Елисеева О.Е., Русецкий К.В.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

olae@open.by

rusetski.k@gmail.com

В работе представлено компонентное проектирование интеллектуальной обучающей системы для подготовки к централизованному тестированию по иностранному языку. Выделенные компоненты ИОС позволяют выполнять разработку отдельных ее частей почти автономно с участием специалистов различного профиля.

Ключевые слова: централизованное тестирование; интеллектуальная обучающая система; ИОС; база знаний.

ВВЕДЕНИЕ

Основной формой контроля знаний абитуриентов при поступлении в высшие и средние специальные учебные заведения Республики Беларусь в настоящее время является централизованное тестирование (ЦТ). Чтобы успешно сдать ЦТ, абитуриентам необходимо владеть комплексом знаний по соответствующей дисциплине в объеме средней школы.

Для поступления на популярные сегодня специальности экономического профиля, специальность «Международные отношения», а также ряд гуманитарных специальностей, требуется сдавать ЦТ по иностранному языку. По данным Республиканского института контроля знаний, число сдававших тесты по иностранному языку в 2012 году составило более 34 тысяч человек. Это свидетельствует о том, что общее количество школьников, готовящихся к централизованному тестированию по иностранному языку достаточно велико. Многие из них прибегают к помощи частных репетиторов, некоторые готовятся к испытаниям самостоятельно. Однако и тем, и другим необходимы постоянные занятия по выполнению тестов, подобных тем, которые они будут проходить в процессе централизованного тестирования. Очевидно, что процесс такой «тренировки» предполагает наличие достаточно большого количества разнообразных упражнений-тестов, а также быструю обратную связь, т.е. информацию о правильности/неправильности прохождения теста. Такая обратная связь в

большинстве случаев представлена в виде заранее заготовленного списка ответов к тестам, сверившись с которыми учащиеся узнают, правильно ли они выполнили тест. Однако простой констатации факта об успешности/неуспешности прохождения теста недостаточно. Для качественной подготовки к тестированию абитуриенту крайне важно сообщить, какая допущена ошибка и какие правила необходимо повторить, чтобы ее исправить при прохождении нового теста. Именно эту задачу в большинстве своем решают преподаватели и репетиторы в процессе подготовки учащихся к ЦТ.

1. Цель работы и постановка задачи проектирования

Резюмируя сказанное выше, сформулируем две основные задачи, которые встают в процессе подготовки к ЦТ по иностранному языку:

- подготовка большого количества тестов;
- анализ и исправление допущенных в процессе тренировочного тестирования ошибок.

Решение обеих указанных задач можно автоматизировать путем использованием технологий искусственного интеллекта. Будем при этом учитывать, что подобного рода задачи вполне успешно решаются в рамках интеллектуальных обучающих систем (ИОС) [Chad Lane, 2006], [Рыбина, 2008]. В связи с этим **целью** данной работы ставится проектирование ИОС для подготовки к ЦТ. Так как указанная система является достаточно сложной и требует привлечения специалистов из различных

предметных областей (см. ниже), то становится очевидным, что при ее проектировании необходимо выделить компоненты, разработка которых возможна автономным образом. Фактически речь идет о выделении вполне самостоятельных компонентов, создание которых возможно с привлечением специалистов различных направлений и квалификации (различных компетенций), а также с учетом особенностей обработки информации различного типа и требуемых для этого методов и технологий.

2. Компетенции

Создание любой компьютерной системы обучения предполагает привлечение специалистов различного профиля. Когда речь идет о создании ИОС, следует учитывать, что ее разработка носит так называемый междисциплинарный характер.

Учитывая сказанное, в контексте данной работы выделим основные необходимые для проектирования и реализации рассматриваемой ИОС компетенции:

- глубокие знания иностранного языка (на лингвистическом уровне);
- владение методикой составления тестов различного вида для централизованного тестирования по иностранным языкам;
- знание технологии организации централизованного тестирования;
- владение современными технологиями программирования;
- владение технологиями представления данных и знаний;
- владение технологиями представления, хранения и обработки сложно структурированной информации;
- владение технологиями проектирования и реализации интерфейсов обучающих систем;
- владение технологиями защиты информации и организации авторизованного доступа к данным.

Каждая из перечисленных выше компетенций требует учета особенностей организации информации и знаний соответствующего типа. В процессе компонентного проектирования ИОС для подготовки к ЦТ будем ориентироваться на различные виды подлежащих формализации и обработке знаний, с одной стороны, и различные целевые группы пользователей – с другой.

3. Целевая аудитория (пользователи) ИОС для подготовки к ЦТ по иностранному языку

Основной целевой аудиторией (конечными пользователями) проектируемой ИОС являются школьники старших классов и абитуриенты, готовящиеся к прохождению испытаний на централизованном тестировании. Будем исходить из предположения, что обучаемые положительно

мотивированы к обучению. К другой категории пользователей ИОС можно отнести лиц старшего возраста, которые занимаются управлением и подготовкой к сдаче тестов. В большинстве случаев они будут принимать решение о целесообразности использования предлагаемого программного продукта в качестве инструмента для такой подготовки. Речь идет о родителях, преподавателях подготовительных курсов, репетиторах. Очевидно, что каждая из обозначенных целевых групп (категорий) пользователей будет использовать ИОС для достижения различных целей (рис.1).

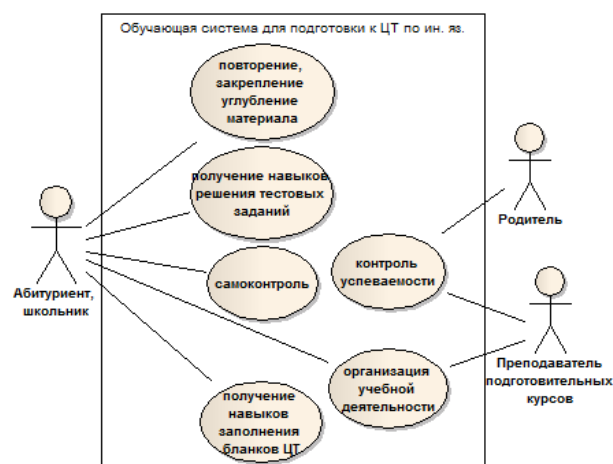


Рисунок 1 – Цели пользователей ИОС по подготовке к ЦТ по иностранному языку

Из рисунка1 видно, что:

- Школьники и абитуриенты могут использовать ИОС для повторения, углубления и отработки знаний иностранного языка, приобретенных ими в школе, а также получения практических навыков решения тестовых заданий и заполнения тестовых бланков. При этом особое значение приобретает формирование навыков самоконтроля, что в рамках школы, как правило, не происходит в силу специфики организации учебного процесса, предполагающего постоянный контроль со стороны учителя.
- Преподаватели подготовительных курсов и репетиторы могут воспользоваться ИОС для организации учебной деятельности, а также в качестве источника многообразных тестовых упражнений, которые в данной системе генерируются в автоматизированном режиме.
- Родители школьников и абитуриентов с помощью ИОС получают возможность наблюдать за успеваемостью и общим уровнем подготовки своих детей к предстоящему тестированию.

4. Концепция и компоненты ИОС

Далее для наглядности в качестве примера будем рассматривать ИОС, рассчитанную на подготовку школьников к тестированию по конкретному иностранному языку. В качестве такового избран английский язык, так как он является наиболее распространенным. Создание в качестве прототипа ИОС по подготовке к ЦТ по одному языку позволит

выявить ее достоинства и недостатки, отладить методику ее практического использования, а также создать технологию тиражирования ИОС на случай других иностранных языков.

Исходя из выше сказанного, база знаний ИОС в идеале должна содержать информацию, соответствующую курсу английского языка в объеме средней школы, который разбивается на лексические и грамматические темы. При проектировании и реализации рассматриваемой ИОС будем учитывать возможность постепенного наращивания базы знаний различными темами. Так, для первой версии системы можно ограничиться несколькими грамматическими темами: «Времена глаголов», «Употребление артиклей», «Образование формы множественного числа существительных» и несколькими лексическими: «О себе», «Моя семья», «Город Минск».

Следуя общепринятой технологии проектирования прикладных интеллектуальных систем и ИОС, в частности, выделим в качестве основных компонентов проектируемой системы следующие (рис.2):

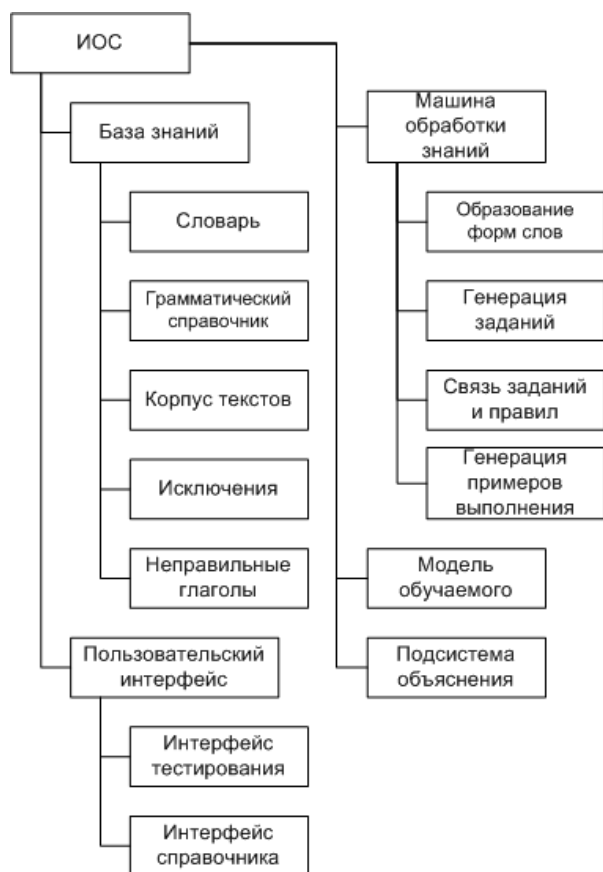


Рисунок 2 – Компоненты ИОС для подготовки к ЦТ по иностранному языку

- База знаний – информационно-содержательное ядро ИОС, основу которого составляет необходимый для подготовки к ЦТ учебный материал по иностранному языку;

- Машина обработки знаний – функциональное ядро ИОС, основной задачей которого является автоматизация создания

тренировочных тестов и реализация различных режимов тестирования знаний;

- Модель обучаемого – специально выделенный из базы знаний компонент, в рамках которого хранится и обрабатывается информация об учащихся и их достижениях;

- Подсистема объяснений – относительно автономный компонент машины обработки знаний, в задачи которого входит объяснение ошибок, допущенных учащимися в процессе выполнения заданий тестов;

Пользовательский интерфейс – стандартный компонент любой системы, но, с учетом функциональных особенностей рассматриваемой ИОС подразделяемый на два отличающихся друг от друга подвида интерфейса.

4.1. База знаний

Как указывалось выше, база знаний является информационно-содержательным ядром ИОС, включающем знания об изучаемой предметной области, а именно об иностранном языке. При проектировании базы знаний иностранного языка имеет смысл учитывать как общие лингвистические закономерности, которые берутся за основу изучения языка, так и частные особенности строения каждого конкретного языка (как лингвистического, так и методического характера). Так, например, вне зависимости от того, по какому языку будет вестись подготовка к ЦТ, в составе базы знаний ИОС выделены компоненты «Словарь», «Грамматический справочник», «Корпус текстов». С учетом особенностей, например, английского языка нами выделены также компоненты «Исключения» и «Неправильные глаголы», т.к. именно этим тематикам уделяется особое внимание в процессе подготовки к ЦТ по английскому языку.

Накопление достаточного словарного запаса является неотъемлемой частью изучения любого иностранного языка. Поэтому одним из важнейших компонентов ИОС является **Словарь**. В отличие от большинства существующих в настоящее время словарей иностранных языков, мы рассматриваем словарь не в «традиционном» смысле (как набор упорядоченных по алфавиту слов с транскрипцией и переводом), а семантически структурированный. Это означает, что в рассматриваемом словаре:

- каждому слову приписывается грамматическая информация (часть речи, ее признаки);

- каждое слово связано с другими словами различными семантическими отношениями (например, быть синонимом, антонимом, гипонимом, гиперонимом и пр.);

- словам ставятся в соответствие примеры их употребления в текстах, которые хранятся и систематизируются в компоненте «Корпус текстов».

Поясним вышесказанное на конкретном примере.

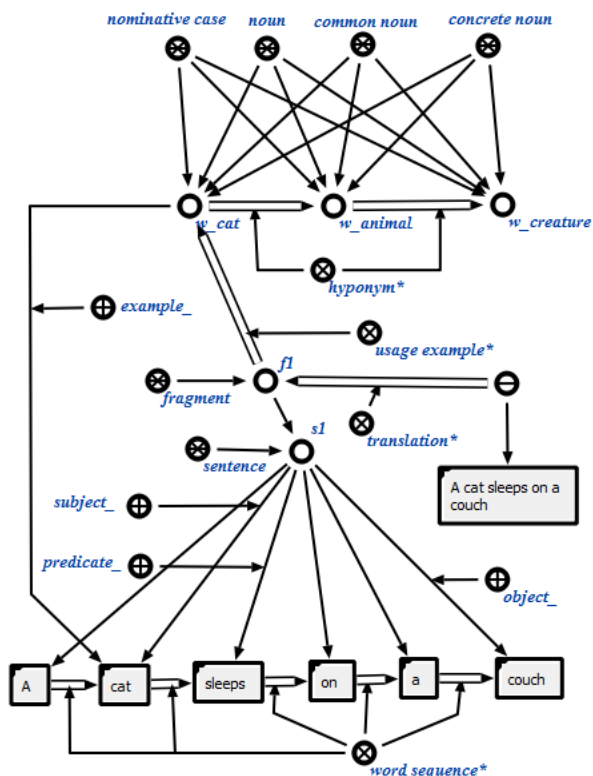


Рисунок 3 – Пример представления словарной статьи в памяти интеллектуальной обучающей системы на языке SCg.

На рисунке 3 предложено описание слова в семантически структурированном словаре, выполненное на языке SCg и отвечающее описанному выше требованиям. Подробно этот язык описан в работе [Голенков и др, 2001]. Язык SCg был выбран в силу того, что он позволяет одновременно наглядно и при этом формально корректно изображать лингвистические структуры в графической форме.

Произведем содержательную интерпретацию приведенной SCg-конструкции. Под словом будем понимать множество реализаций слова в различных предложениях. Заметим также, что к идентификатору знака слова добавляется префикс $w_$, чтобы отличать знак слова в семантической сети от знака понятия, которое это слово обозначает. Слово «cat» (кошка, кот) является конкретным нарицательным существительным в именительном падеже. Это слово является гипонимом по отношению к слову «animal» (животное), которое, в свою очередь является гипонимом по отношению к слову «creature» (живое существо), что отражено с помощью бинарных ориентированных связей отношения *гипоним** (гипоним*). Слова «animal» и «creature» также являются конкретными нарицательными существительными в именительном падеже. Примером употребления слова «cat» является фрагмент текста $f1$, который состоит из предложения $s1$, которое записывается на естественном языке как «Cat sleeps on a couch» (Кот спит на диване). В предложении $s1$ входят слова «cat», «sleeps», «on», «a», «couch», причем слово «A» предшествует слову «cat», которое

предшествует слову «sleeps», и так далее. Слово «cat» в предложении $s1$ является подлежащим; слово «sleeps» в предложении $s1$ является сказуемым; а слово «couch» в этом предложении является дополнением, что обозначается соответствующими атрибутами.

Особенностью представления **Грамматического справочника** ИОС для подготовки к ЦТ по иностранному языку является то, что информация в нем рассчитана как на использование (изучение) учащимися, так и на реализацию операций автоматического создания заданий тестов. Основу справочника составляет набор грамматических правил соответствующего языка, которые представлены в двух видах: 1) формализованных правил образования словоформ, словосочетаний, предложений; 2) учебных естественно-языковых текстов объяснений для учащихся указанных правил.

Для создания разнообразных заданий и тестов необходимы источники иноязычных текстов [Централизованное, 2012]. Практически все тесты создаются на основе оригинальных текстов. Именно поэтому в рассматриваемой ИОС специально выделен компонент «**Корпус текстов**», который планируется использовать с двумя целями: а) для автоматизированного создания заданий тестов; б) для обеспечения связей слов из Словаря с их словоупотреблениями в текстах (пример такой связи приведен выше на рисунке 3). В связи с этим рассматриваемый корпус размечается с учетом того, какие тестовые задания могут в дальнейшем создаваться на их основе.

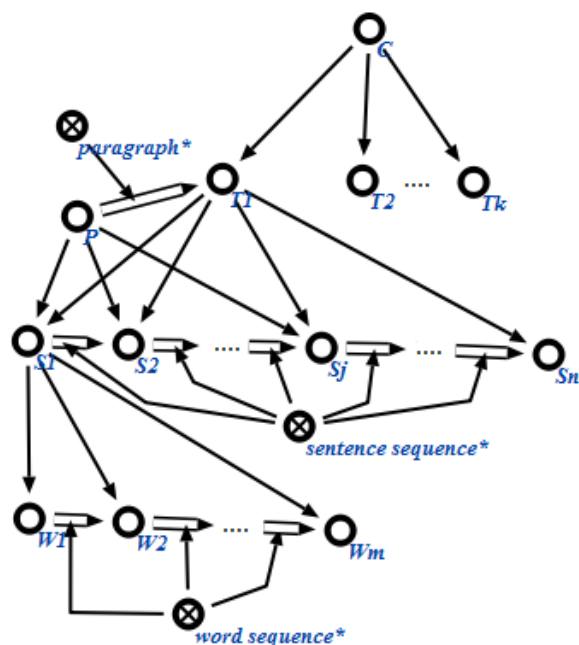


Рисунок 4 – Схематичное представление корпуса текстов в памяти интеллектуальной обучающей системы

Дадим интерпретацию структуре корпуса, схематично изображенной на рисунке 4. Корпус текстов C представляет собой неориентированное множество текстов:

$$C = \{T_1, T_2, \dots, T_k\} \quad (1)$$

Текст T_1 представляет собой ориентированное множество предложений:

$$T_1 = \langle S_1, S_2, \dots, S_j, \dots, S_n \rangle \quad (2)$$

Предложение S_1 представляет собой последовательность слов

$$S_1 = \langle W_1, W_2, \dots, W_m \rangle \quad (3)$$

Отметим, что каждое из слов имеет описание, подобное приведенному на рисунке 3.

Подмножество предложений текста, выделяемое на письме красной строкой, будем называть абзацем (на рисунке 4 – множество P)

$$P = \langle S_1, S_2, \dots, S_j \rangle \quad (4)$$

Принадлежность знаков текстов, предложений и слов соответствующим множествам не приводится, чтобы не загромождать иллюстрацию. Разделители (пробелы, знаки препинания) формализовывать не требуется, так как те части текста, которые они отделяют, можно формализовать как подмножества соответствующих структур. Например, простая часть сложносочиненного предложения будет являться подмножеством предложения. Проиллюстрируем данный факт на примере предложения «It rained, so I took an umbrella» (Шел дождь, поэтому я взял зонтик).

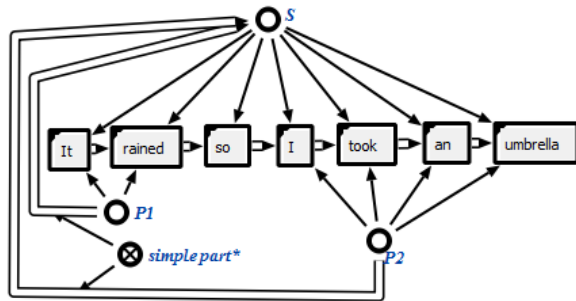


Рисунок 5 – Пример выделения простых частей в сложносочиненном предложении.

Приведенное на рисунке 5 предложение состоит из двух простых частей, границы которых определяются подмножествами P_1 и P_2 соответственно. Значит, формализовывать запятую, которая разделяет части предложения на письме, нет необходимости.

Учитывая особенности английского языка, в базе знаний выделены компоненты «Исключения» и «Неправильные глаголы», в рамках которых предусматривается хранение и формальное описание всевозможных исключений образования, например, форм множественного числа некоторых слов, а также временных форм неправильных глаголов. Все формы слова, которое является исключением из правил формообразования, представляются в базе знаний в явном виде, как на

рисунке 6.

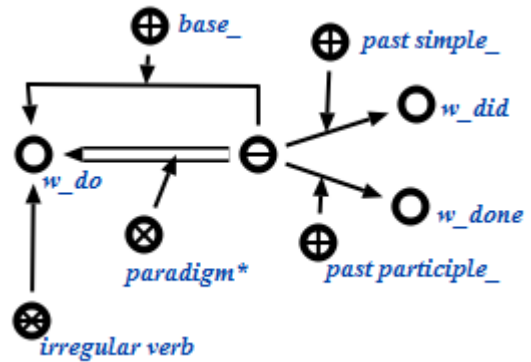


Рисунок 6 – Представление информации о неправильном глаголе в памяти системы.

На приведенном выше рисунке неправильный глагол «do» связан со множеством своих форм отношением $paradigm^*$ (парадигма*). Вхождения форм в множество словоформ помечаются соответствующими атрибутами $base_$, $past\ simple_$, $past\ participle_$ (инфинитив_, прошедшее время_, причастие прошедшего времени_).

Как указывалось выше, относительно автономным элементом базы знаний является модель обучаемого, в рамках которой сохраняется информация об индивидуальных особенностях учащихся. Частично модель обучаемого представляет собой профиль пользователя, подобный тем, которые создаются в процессе регистрации учетных записей пользователей в любой информационной системе. В отличие от «традиционных» систем, модель обучаемого рассматриваемой ИОС предусматривает хранение протокола работы с системой и фиксацию допускаемых при выполнении заданий тестов ошибок, с целью дальнейшего их исправления.

Кроме того, база знаний ИОС для подготовки к ЦТ должна содержать знания о тестировании – структуру тестов, типы и структуру тестовых заданий, и т.п.

4.2. Машина обработки знаний

Главной задачей машины обработки знаний ИОС для подготовки к ЦТ по иностранному языку является автоматизация рутинных функций по созданию заданий тестов на основе грамматических правил, словаря и корпуса текстов из базы знаний. Среди механизмов (операций), обеспечивающих указанную автоматизацию, выделим следующие наборы:

- образование словоформ и их поиск в текстах по заданным признакам;
- генерация тестовых заданий различного типа на основе оригинальных текстов;
- обеспечение связей сгенерированных заданий и использованных для этого грамматических правил;

- генерация примеров выполнения заданий тестов для проверки правильности выполнения тестов учащимися и обеспечения режима обучения.

4.3. Подсистема объяснения

Как указывалось выше, подсистема объяснения является частью машины обработчика знаний. В рамках подсистемы объяснения обеспечиваются следующие режимы работы с ИОС:

- переход к различным разделам грамматического справочника из заданий тестов, а также свободная навигация по нему;
- поиск и навигация по словарю;
- тестирования в режиме обучения (тренировки) с пояснениями и отсылками к правилам, без ограничений по времени и количеству ошибок;
- реализация экзаменационного тестирования, без объяснений, ограниченный по времени (120 минут).

4.4. Пользовательский интерфейс ИОС

Пользовательский интерфейс предназначен для организации эффективного диалога пользователя и обучающей системы в процессе достижения целей обучения. Исходя из этого, разработка интерфейса обучающей системы должна начинаться с разработки примеров диалога. Образцы диалогов являются отправной точкой для выделения основных понятий и отношений, а также являются основой этапа тестирования и отладки системы.

Важнейшим элементом диалога является цель общения. Цель пользователя - изучение некоторого иностранного языка в объеме, достаточном, чтобы успешно сдать ЦТ. Цель системы – осуществить подготовку пользователя, обеспечивающую это.

Наилучшим вариантом интерфейса для обучающей системы является естественно-языковой интерфейс (ЕЯИ), так как такой интерфейс наиболее адекватен для неопытного пользователя. Он позволяет ему эффективно взаимодействовать с системой, несмотря на невысокий уровень подготовки и с максимальной быстротой формулировать произвольные запросы к системе. Однако трудоемкость разработки ЕЯИ довольно велика, и для первой версии разрабатывать такой интерфейс нецелесообразно. Вместо этого для начала предлагается использовать в качестве языка общения уже упомянутый ранее язык SCg. Однако не стоит упускать из виду и ЕЯИ, поэтому приведем фрагмент диалога между пользователем (П) и системой (С) параллельно на естественном языке и SCg.

П: Как подразделяются предложения по сложности?

С: По сложности предложения подразделяются на простые, сложные и составные.

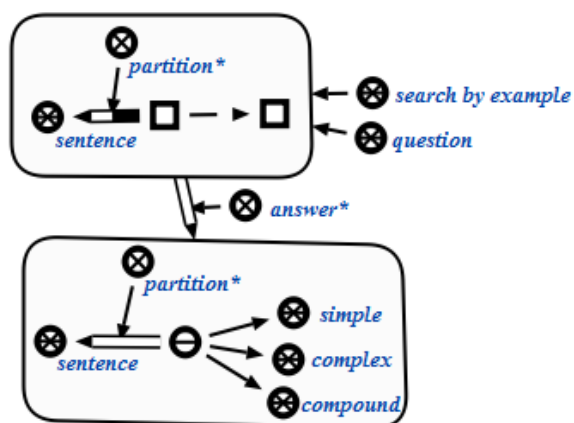


Рисунок 7 – Фрагмент диалога на SCg

Заключение

В завершение данной работы отметим, что выделенные в процессе проектирования компоненты ИОС для подготовки к ЦТ по иностранному языку требуют дальнейшего детального рассмотрения с привлечением различных компетенций. Хотя реализация ЕЯИ и не предполагается в первой версии системы, но лингвистические компоненты, не касающиеся непосредственно ЦТ, лягут в основу разработки ЕЯИ в будущих версиях.

Библиографический список

- [Chad Lane, 2006] Chad Lane H. Intelligent Tutoring Systems: Prospects for Guided Practice and Efficient Learning. 2006.
- [Голенков и др, 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах /В. В. Голенков [и др.] – Мн. : БГУИР, 2001.
- [Рыбина, 2008] Рыбина Г.В. Обучающие интегрированные экспертные системы: некоторые итоги и перспективы // Искусственный интеллект и принятие решений 1/2008. – С.22-46.
- [Централизованное, 2012] Централизованное тестирование. Английский язык : сборник тестов / Респ. ин-т контроля знания М-ва образования Респ. Беларусь. – Минск: Аверсэв, 2012.

COMPONENT DESIGN OF INTELLIGENT TUTORING SYSTEM TO PREPARE STUDENTS FOR CENTRALIZED TESTING IN A FOREIGN LANGUAGE

Yeliseyeva O.E., Rusetski K.V. *

*Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

olae@open.by
rusetski.k@gmail.com

In the process of designing an intelligent tutoring system to prepare for centralized testing in a foreign language, select the component, further design and development of which is possible in offline mode with specialists in various fields.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004:8+37

ИНТЕЛЛЕКТУАЛЬНАЯ ОБУЧАЮЩАЯ СИСТЕМА AI-TUTOR ДЛЯ ЭКСПРЕСС-ИЗУЧЕНИЯ ИНОСТРАННЫХ ЯЗЫКОВ В СЕТИ ИНТЕРНЕТ

Киселёв В.В. *, Елисеева О.Е. **

* ООО «Речевые технологии», г. Минск, Республика Беларусь

kiselev-v@speechpro.com

** Белорусский государственный университет, г. Минск, Республика Беларусь

olae@open.by

Рассматривается обучающая система для экспресс-изучения иностранных языков, в основу реализации которой положены принципы создания экспертных систем и речевые технологии. Обучающая система реализуется в виде веб-ресурса с авторизованным доступом.

Ключевые слова: экспертная обучающая система; речевой интерфейс; веб-ресурс.

ВВЕДЕНИЕ

В настоящее время всё более актуальной становится проблема быстрого изучения иностранных языков. Активным образом при этом используются ресурсы сети Интернет. Кроме того, современные технологии создания веб-ресурсов позволяют применять в их разработке средства передачи и обработки речевой информации. Такая возможность появилась относительно недавно, и уже сейчас на некоторых веб-ресурсах по изучению иностранных языков в той или иной мере это используется для повышения эффективности приобретения учащимися речевых навыков. Кроме того, начинают появляться веб-ориентированные системы обучения с использованием методов и средств искусственного интеллекта [Рыбина, 2008].

В данной работе предлагается к рассмотрению веб-ориентированная обучающая система для экспресс-изучения иностранных языков ai-tutor, в реализации которой используются современные методы обработки и передачи по сети речевой информации, а также некоторые подходы к реализации экспертных обучающих систем. Это обусловлено необходимостью создания веб-ресурса, с помощью которого у учащихся с различным базовым уровнем знаний появилась возможность обучаться по индивидуальной, оптимальной для них лично стратегии. Применение технологий распознавания речи даёт возможность быстрого и качественного приобретения учащимися навыков говорения и «выхода в речь», что, заметим, является

одной из самых сложных и актуальных проблем при изучении иностранных языков.

1. Методы и технологии

Комплекс основных применяемых для реализации обучающей системы методов и технологий представлен в табл. 1.

Таблица 1 – Методы и технологии реализации ai-tutor

Возможности ai-tutor	Методы и технологии реализации
Быстрое формирование навыков общения на иностранном языке	технологии искусственного интеллекта для построения оптимальных индивидуальных стратегий обучения
Речевое взаимодействие с системой, постановка и корректировка произношения на иностранном языке	распознавание речи
Медиа-контент: аудиозаписи, видеозаписи	технологии удаленной работы с мультимедиа
Удаленный доступ с персональных компьютеров, планшетов или смартфонов	клиент-серверные архитектура и кросс-браузерные технологии программирования

Основная задача, которую необходимо решить при разработке рассматриваемой обучающей системы, состоит в том, чтобы обеспечить быстрое приобретение пользователями речевых навыков

общения на иностранном языке. В системе ai-tutor указанная задача решается с привлечением технологий искусственного интеллекта, а именно принципов создания экспертных обучающих систем, в которых учебный процесс представляется в виде совокупности формальных моделей, а информация о нем хранится и обрабатывается в соответствующих базах знаний (БЗ). Учебная информация, а также взаимодействие пользователя с системой в рамках указанных формальных моделей представляются с использованием графовой нотации.

Для обеспечения обработки речевого сигнала, поступающего от пользователя в процессе диалога с системой, в реализации используются современные методы получения аудиоданных. Анализ и обработка аудиоданных осуществляется при помощи алгоритмов распознавания речи.

В системе ai-tutor поддерживаются технологии удаленной работы с мультимедиа, а именно средства обеспечения доступа пользователя к аудио и видеоконтенту учебных модулей. В качестве аудиоконтента используются в большинстве своем заранее записанные звуки, слова, выражения носителями изучаемых иностранных языков. Указанная аудио база данных используется также и как набор эталонов для распознавания речи при выполнении упражнений и тестов.

Будучи сетевым программным комплексом, рассчитанным на использование образовательного контента пользователями персональных компьютеров, а также планшетов или смартфонов, предлагаемая обучающая система реализуется на основе клиент-серверной архитектуры и соответствующих технологий программирования. Следует при этом заметить, что указанные технологии накладывают определенные ограничения на возможности реализации интеллектуальных функций ai-tutor.

2. Интеллектуальные функции обучающей системы ai-tutor

Как сказано выше, важнейшей особенностью предлагаемой обучающей системы является использование в её реализации методов и средств искусственного интеллекта. Рассмотрим более подробно, какие именно «интеллектуальные функции» реализуются в рамках ai-tutor (рисунок 1).

Принципиальное отличие рассматриваемой системы от других систем обучения иностранным языкам заключается в том, что в основу обучения положен принцип тренировки, а именно отработки навыков говорения и аудирования на основе достаточного количества практических упражнений. При этом количество упражнений, выполняемых различными учащимися в процессе работы с ai-tutor, различно. В зависимости от персональных успехов и достижений каждому учащемуся подбираются те упражнения, которые способствуют формированию необходимых навыков. Такое

функционирование обеспечивается, благодаря представлению учебного материала на основе модели обучения (см. ниже).

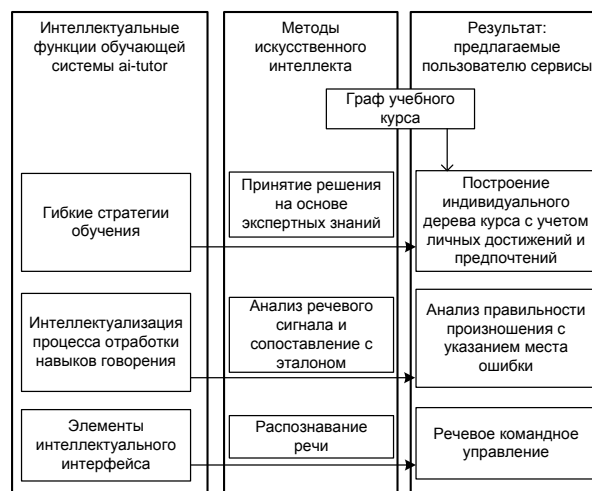


Рисунок 1 – Интеллектуальные функции ai-tutor

Итак, «интеллектуальность» ai-tutor базируется на следующих особенностях её реализации:

а) ai-tutor строится по принципам экспертных обучающих систем (ЭОС), в ней обеспечивается построение гибких индивидуальных стратегий обучения;

б) пользователю предоставляется возможность не только неоднократно вслух повторять звуки, слова, фразы вслед за диктором, но и исправлять допущенные ошибки, т.к. на основе анализа аудиозаписей и их сопоставления с эталонными записями дикторов-носителей языка можно увидеть место допущенной ошибки;

в) реализуются элементы интеллектуального интерфейса в виде речевого командного управления. На начальном этапе реализации речевое командное управление выражается в возможности устного ввода ответов на вопросы некоторых практических заданий.

3. Архитектура обучающей системы

На рисунке3 представлена архитектура системы ai-tutor, которая включает 4 основных блока:

1. Блок реализации обучения, в рамках которого осуществляется моделирование процесса обучения иностранному языку.

2. Блок служебных функций, в рамках которого реализуется ряд модулей для обеспечения надежного функционирования системы в целом.

3. Блок формирования контента, в задачи которого входит обеспечение возможностей гибкого пополнения необходимой учебной информацией модели учебных воздействий в рамках блока реализации обучения.

4. Блок интерфейса с пользователем, в рамках которого учащимся и преподавателям предоставляются разные возможности: учащимся – для взаимодействия с блоком реализации обучения;

преподавателям – для наполнения учебными материалами посредством блока формирования контента.

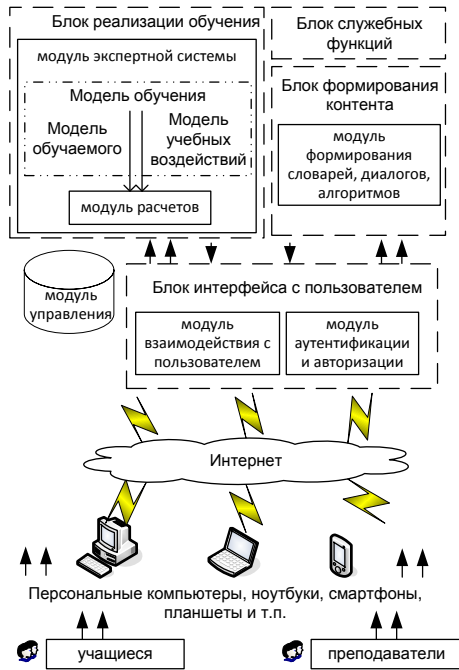


Рисунок 3 – Архитектура системы ai-tutor

4. Модель обучения

Как указывалось выше, особенностью рассматриваемой системы является обеспечение гибких стратегий обучения для различных учащихся. Поэтому в данной работе рассмотрим более подробно формальные модели, которые являются основой блока реализации обучения и обеспечивают указанную функциональность.

В основу проектирования ai-tutor положена формальная **модель обучения** LM (learning model), в рамках которой решаются задачи построения оптимальной стратегии обучения для каждого учащегося. В рамках модели обучения строятся модель учебных воздействий TAM (training actions model) и модель обучаемого SM (student model).

Модель учебных воздействий TAM представляется в виде ориентированного графа, включающего узлы трех типов: Т – теоретический компонент, Р – практическое (тренировочное) упражнение, С – контролирующее упражнение для оценки уровня достижений учащегося, на основе которого осуществляется поиск пути на графе, соответствующего оптимальному пути прохождения учебного курса конкретным обучаемым. Каждый узел трактуется нами в данной модели как отдельный урок курса, который является минимальным учебным воздействием. Урокам (узлам графа) в свою очередь ставятся в соответствие векторы составляющих процесса изучения иностранного языка:

$a(v, g, ph)$ – один или несколько аспектов языка: лексика (словарь) (v), грамматика (g), фонетика (ph);

$n(A, S, R, W)$ вид приобретаемых знаний и навыков: аудирование (A), говорение (S), чтение (R), письмо (W).

Координаты обоих указанных векторов могут принимать одно из двух значений: 0 (нет) или 1 (есть). Так, например, если координата v вектора a для некоторого узла равна 0, это означает, что в соответствующем уроке нет новой словарной составляющей. Если же указанная координата равна 1, то в данном уроке изучаются новые слова.

На рис.4 приведен пример графа, описывающего совокупность уроков (учебных воздействий).

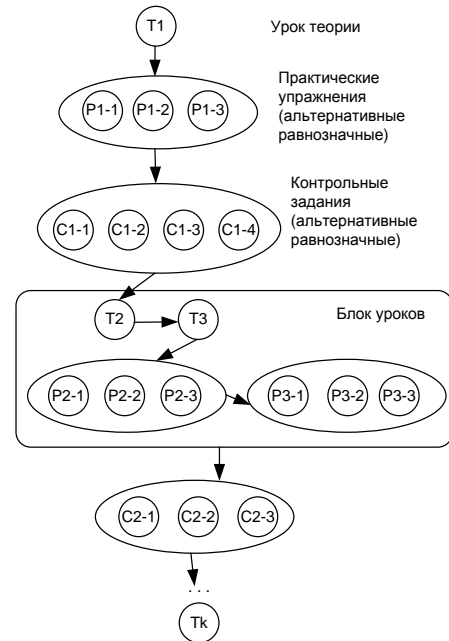


Рисунок 4. Упрощенное представление графа, описывающего совокупность учебных воздействий в модели обучения

На данном графе узлами с идентификаторами обозначены уроки различного типа: теоретические, практические, контрольные. Дуги, выходящие из узлов и входящие в области овальной формы, (например, из урока теоретического типа Т1 в область, охватывающую узлы тренировочных упражнений P1-1, P1-2, P1-3) означает, что после указанного урока (Т1) может следовать любой из равнозначных альтернативных уроков (P1-1, P1-2, P1-3). При этом выбор одного из альтернативных уроков может происходить либо случайным образом (например, на начальных стадиях «знакомства» системы с учащимся), либо на основе анализа истории взаимодействия учащегося с ai-tutor, которая сохраняется в рамках модели обучаемого. Дуги, соединяющие области овальной формы, означают, что после прохождения определенного уровня иерархии уроков (одного или нескольких, в зависимости от успехов учащегося), например, P1-1, происходит переход к одному из уроков следующего уровня иерархии, например, к одному из альтернативных равнозначных уроков множества C1-1 – C1-4. Для упрощения рассматриваемой модели и/или по усмотрению методиста (преподавателя языка, участвующего в создании

контента системы) из всех равнозначных уроков может быть выделен один обязательный, который предлагается пройти всем учащимся. Все равнозначные уроки связаны друг с другом двунаправленными дугами по схеме «каждый с каждым». Заметим, что для улучшения восприятия на рис. 4 эти дуги не показаны.

Итак, если учащийся успешно выполняет упражнение из совокупности альтернативных равнозначных, то осуществляется переход к следующему уроку (не равнозначному), т.е. к следующему уровню иерархии графа учебных воздействий. Иначе учащемуся предоставляется альтернативное равнозначное упражнение (он остается на том же уровне иерархии графа).

Для обеспечения гибкости и индивидуализации обучения учащимся разрешается пропускать те или иные уроки (в этом случае в модели обучаемого делается соответствующая пометка, и при необходимости учащемуся еще раз предоставляется возможность пройти пропущенный урок).

Описанным выше образом, в зависимости от успехов и пожеланий учащегося, происходит построение оптимального пути прохождения учебного курса индивидуально для каждого пользователя. Указанный путь сохраняется в виде истории взаимодействия с системой в модели обучаемого (рис.5). В некотором приближении можно говорить о том, что сформированные на основе рассмотренной модели учебных воздействий ТАМ структура и содержание учебного материала являются частью общей базы знаний (БЗ) ai-tutor.

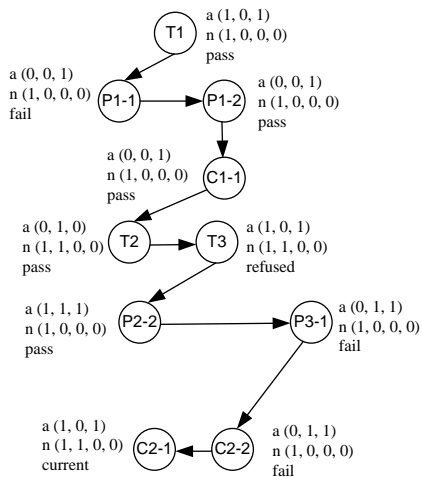


Рисунок 5. Пример графа истории взаимодействия обучаемого с ai-tutor

В модели обучаемого SM фиксируются личные данные об обучаемом и история взаимодействия с ai-tutor в виде пройденного в процессе обучения пути на графе учебных воздействий с необходимыми пометками о достигнутых результатах, отказах и пр.

Из примера графа истории взаимодействия учащегося с системой, изображенного на рис.5, видно, какие знания и навыки, в каких аспектах языка приобрел учащийся. Эта информация

копируется из базы знаний модели учебных воздействий и становится фрагментом БЗ модели обучаемого. На данном примере также отмечено, с каким успехом были пройдены те или иные уроки: pass – успешно (для теоретических уроков это значение по умолчанию), fail – не успешно, refuse – учащийся отказался от прохождения урока, current – текущее состояние прохождения курса (либо учащийся в текущий момент времени проходит данный урок, либо остановился на нем и завершил сеанс работы с ai-tutor).

Заключение

Представленная в работе обучающая система ai-tutor ориентирована на гибкие, индивидуальные стратегии обучения, благодаря использованию современных технологий искусственного интеллекта и речевых технологий. В отличие от большинства существующих на сегодняшний день электронных образовательных ресурсов, разработка ai-tutor не зависит от изучаемого языка и используемой методики преподавания. Благодаря этому, на основе предложенной архитектуры возможно построение совокупности аналогичных систем для приобретения навыков общения на различных языках. На начальной стадии реализации в системе реализуются упрощенные модели обучения. В процессе ее развития планируется существенное расширение параметров, обрабатываемых в рамках модели обучаемого в направлении анализа речевых ошибок и выработки наиболее оптимальной индивидуальной стратегии обучения. Используемые подходы получат свое воплощение также в системах, способствующих устранению дефектов речи.

Библиографический список

[Рыбина, 2008] Рыбина Г.В. Обучающие интегрированные экспертные системы: некоторые итоги и перспективы // Искусственный интеллект и принятие решений 1/2008. – С.22-46.

INTELLIGENT TUTORING SYSTEM AI-TUTOR FOR RAPID LEARNING OF FOREIGN LANGUAGES IN THE INTERNET

Kiselev V.V. *, Yelisseyeva O.E. **

* *Speech Technology Ltd.,*
Minsk, Republic of Belarus
kiselev-v@speechpro.com

** *Belarusian State University,*
Minsk, Republic of Belarus
olae@open.by

This paper considers the tutoring system for rapid language learning, based on the principles of implementing an expert systems and speech technologies. The tutoring system is implemented as a Web resource with authorized access.



УДК 004.822:514

ИСПОЛЬЗОВАНИЕ АФФИКСАЦИИ В ЭКСПЕРТНЫХ ОБУЧАЮЩИХ СИСТЕМАХ ПРИ ИЗУЧЕНИИ СЛОВООБРАЗОВАНИЯ

Головня А.И.*

** Белорусский государственный университет,
г. Минск, Республика Беларусь*

golovnjaai@bsu.by

Разработчикам систем искусственного интеллекта важно найти плюсы и минусы в разработке обучающих экспертных систем, которые могли бы помочь иностранным студентам лучше освоить русское словообразование, определить что же такое «аффиксация», каково ее значение при разработке интеллектуальных систем, что такое омонимия аффиксов, какова роль омонимии при изучении словообразования.

Ключевые слова: аффиксация, словообразование, классификация, префиксация, корень, суффиксация.

Словообразование русского языка очень важный раздел языкознания. При его освоении очень важно учитывать, что не во всех языках аффиксация развита так многогранно, как в русском языке. Поэтому важно при разработке обучающих экспертных систем обратить внимание на использование служебных и корневых аффиксов в русском словообразовании. Важным параметром при этом является и учет омонимии аффиксов.

При преподавании словообразования как русским, так и иностранным студентам отдельно даются способы префиксации и суффиксации. Но при изучении словообразования на занятиях можно попытаться создать у студентов-иностранцев целостное представление об этом языковом явлении. Особенно важно учитывать целостное представление об словообразовании при создании обучающих экспертных систем.

Возьмем для этого за исходные системообразующие три структурных элемента слова – префикс (П), корень (К) и суффикс, который будет включать и флексию (С) – и построим целостную связную систему аффиксации в виде трехмерного плюсминусового куба Карпова (аналога кода Грея, используемого широко в разных предметных областях для представления и приращения знаний от электротехники до биологии).

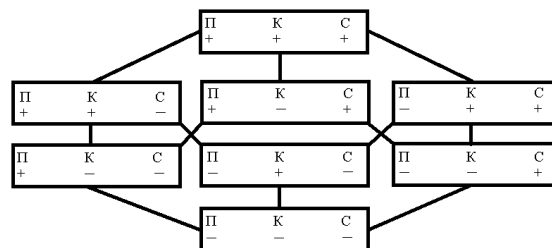


Рисунок 1 - Аффиксация как целостная система

Плюс-минусовой куб имеет 8 подсистем. Сначала покажем его использование как полного классификатора. И выясним вероятностную сторону явления словообразования. Вероятности представляют возможность реализации 4 макросостояний, имеющих от 1 до 3-х микросостояний (по числу плюсов/минусов). Они представлены таблицей 1.

Таблица 1 – Вероятности наполнения подсистем куба

макрососто- яние 1	макросостоя- ние 2	макрососто- яние 3	макросос- тояние 4
микро +++	микро ++-	микро +--	микро ---
	+ - +	- + -	
	- + +	- - +	
вероятност- ь 1/8	вероятност- ь 3/8	вероятност- ь 3/8	вероятност- ь 1/8

Как видим, вероятности подсистемы 1 и подсистемы 8 равны, также равны вероятности макросостояний 2 и 3 и составляющих их

микросостояний. Это говорит о теоретически равных возможностях, хотя мы знаем, что число префиксов и суффиксов меньше числа корней и слов, созданных префиксацией, суффиксацией и совместно префиксацией и суффиксацией. Но нужно помнить, что малочисленность предлогов, от которых созданы префиксы, компенсируется высокой частотой их использования, что снимает противоречие.

П8 представляет все множество слов русского языка до классификации в терминах префиксов, корней и суффиксов. Здесь же находится и ученый-исследователь и результаты классификации или эта подсистема может содержать только ученого или пользователя, студента.

П5 содержит список префиксов, которые после начала классификации студент может использовать при словообразовании и проводить сравнение *ходить* – *выходить*, *походить*, *сходить*, *подходить*, *приходить*, *заходить* и т.п., у которых будет симметричная часть *ходить* и асимметричные начала-префиксы. Для проверки он может взять глагол *водить* и сопоставить его связи с аналогичными приставками у другого глагола: *выводить*, *поводить*, *сводить*, *подводить*, *приводить*, *заводить* и т.п. При этом студент может не знать значения слов и действовать формально.

П6 будет содержать список корней – от самых коротких слов – *дом*, *рот*, *сыр* и т.п., до более длинных – *шаман*, *карман*, *ватман* и т.п.

П7 будет содержать список суффиксов, включая и окончания (флексии), которые опять же выделяются на базе симметрии и асимметрии – ср. *ход* и *ход-ок*, *ход-ка*, *ход-ить*, *ход-ули* и т.п.

П2 будет содержать слова из префикса и корня – *по-ход*, *за-ход*, *вы-ход*, *до-ход*, *пере-ход* и т.п. При этом для надежности можно сравнить его с другим словообразовательным рядом: *по-вод*, *за-вод*, *вы-вод*, *до-вод*, *пере-вод*.

П3 будет представлять рамку как инструмент префиксации и аффиксации одновременно. Такие рамки существуют в нашем сознании как эталоны для алгоритмов моделирования. Допустим, рамка «за- ... -ить» вычлняется в сознании на базе существования ряда переменных корней при постоянной рамке – *за-ход-ить*, *за-вод-ить*, *за-нос-ить*, *за-лож-ить* и тысячи других уже с другой длиной корневой частью – *за-пуст-ить*, *за-стрел-ить* и т.п. Количество таких рамок исчисляется комбинаторикой десятков префиксов и десятков суффиксов. С учетом полипрефиксации и полисуффиксации из тысяч рамок следует отобрать наиболее продуктивные и наиболее частотные.

П4 содержит слова из корня и суффикса и таких слов множество – *ход-ок*, *ход-ка*, *ход-ули*, *ход-унки* и т.п., другими словами все финальные остатки, получаемые при выделении префиксов и корней.

П1 должна содержать и содержит слова со всеми тремя словообразующими элементами – префикс, корень и суффикс. Большая часть Обратного стотысячного словаря представлена полным набором таких слов: *вы-дел-ение*, *про-хожд-ение*, *за-нес-ение*, *при-руч-ение*, *про-вер-ка* и т.п.

Теперь покажем действие алгоритма заполнения подсистем объектами. Сравнивая попарно объекты подсистемы 8, допустим, элементы множества *по/ход* и *за/ход*, *за/ходить* и *вы/ходить* и т.п., мы заполняем подсистему 5 элементами *по-*, *за-*, *вы-* и подобными. Аналогичным образом сравнивая элементы *вы/ход/ить* и *вы/вод/ить* или другие, получаем заполнение подсистемы 6 элементами *-ход-*, *-вод-* и т.д. Сравнение можно проводить не только парами, но и тройками, четверками, «энками». Действуя таким же образом с объектами типа *ход/ить* и *ход-ок*, *ход-ка* и т.п., загружаем подсистему 7 элементами *-ить*, *-ок*, *-ка*. Сопоставление пар таких как: *за/ход/ить* и *за/ход* даёт возможность заполнения подсистем 2 и 1; пары *за/ход/ить* и *ход/ить* – подсистем 1 и 7, а сравнение *вы/ход/ка*: *по/ход/ный* позволяет заполнить третью и шестую подсистемы.

Обязательным моментом является существование взаимодействия элементов, предполагающее и невзаимодействие. Так, взаимодействие в данном случае касается возможности композирования элементов подсистем друг с другом, допустим, элемент *за-* и элемент *-ить* могут образовать рамку «за-...-ить».

Обратимся к примерам. Пусть нам надо получить некоторый объект подсистемы 1, состоящий из префиксальной, корневой и суффиксальной части. Для этого можно взять рамку из подсистемы 3 и корень из подсистемы 6. Возможность соединения данных двух элементов в виде абстракции – это закон композиции, семантическое взаимодействие корня и рамки конкретно указанных *ход* и «за- ... -ить» образуют слово *заходить*, как видим, здесь проявляются и отношения единства.

Соединяя «+ – +» и «– + –», получаем операцию в виде умножения-взаимодействия. Наложив эту операцию на подсистему 8, извлекаем из нее и переносим в подсистему одну лексему *за-ход-ить* или обнаруживаем такую в подсистеме 1. Но точно такого же результата мы можем добиться и другими способами, например созданием элемента *за-ход-ить* из элементов подсистемы 5 приставки *за-* и подсистемы 4, содержащей глагол *ход/ить*. Операция достижения результата, который находится в подсистеме 8 «– – –», получается в результате перемножения кодов подсистем 5 и 4 («+ – –» на «– + +»), умножением-наложением операции на подсистему результата 8 получаем желаемое.

Все вышеизложенное можно обобщить в следующем виде: классификация объектов-систем как системы объектов данного рода при знаковой дихотомии «+» и «–» и признаковой дихотомии «П»,

«К», «С» для множества «первичных элементов» дает 7 новых объектов-систем своего рода.

Теперь свяжем это представление с данными Частотного словаря и наполним схему конкретными примерами, чтобы описать подробнее каждую подсистему таким образом, чтобы она была готова к использованию в экспертных обучающих системах и к преподаванию на занятиях по русскому языку.

Подсистема 8 содержит все слова русского языка до их классификации в терминах префикс, корень, суффикс, исследователя и результаты исследований.

Подсистема 5 – префиксы, связанные с предлогами (*на* и *на-*, *по* и *по-*, *из* и *из/ис-*, *под* и *под-*, *без* и *без/бес-*, *в* и *в-*, *с* и *с-* и др.) и префиксы, не связанные с предлогами (*воз-/вос-*, *пере-*, *раз-/рас-* и др.).

Подсистема 6 содержит чистые корни, которые совпадают с лексемами и в Частотном словаре приводятся с частотами: существительные мужского рода – *год* (2167), *день* (1345), *раз* (1230), *товарищ* (1162), *глаз* (1093), *мир* (1038), *свет* (1031), *друг* (817), *город* (803), *дом* (799), *отец* (605), *час* (488), *лист* (286), *номер* (110) и др.; существительные женского рода, не содержащие суффикса: *рука* (1596), *страна* (1084), *земля* (1074), *сила* (986), *вода* (891), *мать* (564), *мама* 350, *цель* (321) и др.; среднего рода – *дело* (1919), *время* (1856), *слово* (1039), *место* (989), *лицо* (745), *небо* (284) и др. Сюда же можно с оговоркой отнести и первичные наиболее древние глаголы: *быть* (13307), *мочь* (3373), *идти* (1818), *лечь* (16), *знать* (7), *звать* (190), *гнать* (65) и множество наречий – *где*, *куда*, *так*, *там* и другие. Личные местоимения – *я*, *ты*, *он*, *вы* также входят в эту подсистему и могут служить основой для создания глаголов – *тыкать* (говорить на ты), *выкать* (говорить на вы), существительных – *яканье*, *ячество* и т.п. К чистым корням следует отнести часть букв алфавита, все служебные слова нечленимые морфологически, так как и они вносят свой вклад в словообразование, переходя из подсистемы 6 в подсистему 4 – ср. *и того* (словосочетание) – *итого* – слово; *а*, *о* – буквы (звуки) и *акать*, *аканье*, *окать*, *оканье* (типы произношения). Или междометия *ох*, *ах*, *ух* и глаголы – *охнуть*, *охать*, *ахнуть*, *ахать*, *ухнуть*, *ухать* и т.п. Это ядро всей системы, развивающееся в подсистеме верхних уровней.

Подсистема 7 содержит суффиксы, с помощью которых создаются новые слова, например: *-ка*, *-ист*, *-тель*, *-ение/-ние*, *-ова*, *-енн*, *-чик*, *-ик* и т.д.

Подсистема 2 содержит слова, состоящие из префикса и корня: существительные: *на-род* (984), *во-прос* (909), *рас-стрел* (15), *рас-спрос* (4), *за-гул* (2), *вы-ход* (122); глаголы: *по-йти* (1027), *вз-ять*, *по-н-имать*, *с-делать*, *у-ходить*, *в-ходить*, *у-йти*, *у-видеть*, *с-просить*, *про-течь* (2), *про-шагать* (3), *у-плыть* (3), *со-лгать* (11); наречия: *со-обща* (7) и др.

Подсистема 3 является своеобразной рамкой, гештальтом, образцом модели словообразования,

которая и должна работать в интеллектуальных словообразовательных компьютерных системах: «*по- ... -ать*», в которую помещаются корни (*по-чит-ать*, *по-пис-ать*, *по-игр-ать*; *за-пис-ать*, *за-жим-ать*); «*у- ... -ить*» (*у-том-ить*, *у-помн-ить*, *у-ман-ить*, *у-воз-ить* и т.п.). Такие же рамки можно создать и для существительных – ср. «*по-...-ение*» (*по-руч-ение*, *по-нужд-ение*, *по-вел-ение* и т.п.) или «*по-...-ство*» (*по-мыкатель-ство*, *по-пуститель-ство*, *по-печитель-ство*, *по-мешатель-ство*, *поручатель-ство*); «*за- ... -ка*» (*за-прав-ка*, *за-лив-ка*, *за-свозд-ка*), которые могут не содержать или содержать и дополнительные суффиксы. Аналогично дело обстоит и с рамками по созданию наречий: «*не- ... -имо*» (*не-измер-имо*, *не-вынос-имо*, *не-завис-имо*, *не-обход-имо* и др.). Как видим, каждая конкретная рамка характерна для создания одинаковых или разных частей речи. Причем эти рамки-гештальты предполагают как использование чистых корней, так и сочетание корней с префиксами или суффиксами.

Подсистема 4 содержит слова из корня и суффикса с окончанием: существительные: *жи-знь* (1547), *раб-от-а* (1103), *душ-ок* (2), *ком-изм* (1); глаголы: *да-ва-ть*, *дыр-яв-ить* (1), *нян-ч-и-ть* (4), *мы-ть* (5) и др.

Подсистема 1 содержит слова, состоящие из префикса, корня и суффикса: глаголы: *вы-раиц-ива-ть* (17), *за-вар-ива-ть* (11), *на-сол-и-ть* (1), *вы-сох-ну-ть* (7); существительные: *вы-сад-к-а* (18), *о-чар-ова-ние* (1), *по-тепл-ение* (2); наречия: *не-со-мн-енн-о* (72), *по-долг-у* (10) и другие.

Следовательно, можно сделать следующие выводы. Использование ОТСУ в качестве метатеории позволяет доказать системность словообразования русского языка с разных точек зрения.

На занятиях по словообразованию обучаемыми может быть выяснена связь конвергенции и дивергенции с симметрией и асимметрией словообразования. При этом доказывается совместность этих явлений-процессов: словообразование имеет ярко выраженный конвергентно-дивергентный характер. Конвергентность связана с общесистемными явлениями семантики (способность к деривации), а дивергентность – со спецификой конкретной семантики корней, префиксов и суффиксов – т.е. со взаимодействиями и невзаимодействиями семантик.

В процессе обучения студентом применяется целостная полносвязная система аффиксального словообразования русского языка в виде непротиворечивой логико-математической модели, представленной трехмерным плюсом-минусовым кубом, что позволяет наглядно и доказательно представить системный характер словообразования в русском языке.

Рассмотрение языковых законов словообразования русского языка вылилось в

построение вероятностной модели и ее наполнение реальными фактами, что позволяет дать им содержательную непротиворечивую интерпретацию в виде классификации и создания моделей словообразования, которые показывают взаимодействия и невзаимодействия морфем всех трех типов в плане лингвистического и экстралингвистического факторов.

Сам факт нахождения словообразовательных рядов, выделения словообразующих основ и аффиксов, выявление отношений производности и мотивированности и построение на этом основании словообразовательных моделей – все эти явления в лингвистике не новы. Но применение системного подхода к традиционному материалу в виде ОТСУ дает приращение знания на качественно ином уровне, математическом. Именно такой тип приращения знаний за счет их переструктурирования, «новых способов организации уже имеющегося знания» считается некоторыми философами одним из важнейших.

В данном исследовании мы попытались рассмотреть лишь один языковой аспект как систему, состоящую из подсистем. Особенно важным итогом является то, что системный метод действительно эффективен при изучении языка. Одним из результатов применения системного подхода в изучении словообразования русского языка стало обнаружение лингвистической симметрии и асимметрии, что, безусловно, является одним из наиболее впечатляющих и фундаментальных научных достижений, к тому же представленных в виде ряда симметрично-асимметричных матриц и трехмерного плюс-минусового куба. Каждую из подсистем этого куба можно представить в виде такого же или двухмерного плюс-минусового куба. Например, все корни можно разделить на свои и заимствованные, все префиксы разделить по генетическому признаку (созданные на базе предлогов и иным способом с дальнейшей детализацией по степени однозначности или многозначности). Суффиксы и префиксы можно разделить на суффиксоиды и префиксоиды на системной основе. Далее можно рассматривать появление уже не новых слов, а новых значений – метафоризацию, субстантивацию и другие способы, сохраняющие оболочку слова.

Замечательно, что наука в своей истории поднималась на качественно новую ступень каждый раз, когда обнаруживалась какая-либо новая симметрия и/или нарушение уже известных и когда симметрию/ асимметрию и диссимметрию использовали в качестве средств познания. Поэтому отмеченным достижениям следует придать принципиальное значение.

К возможностям системного подхода можно отнести не только представление изучаемого объекта как объект-систему, но и получение системы объектов одного и того же рода, обнаружение в системе объектов данного рода полиморфизма и изоморфизма,

симметрии/асимметрии и диссимметрии, отношения противоречия, непротиворечия, описываемые математическим способом. Системный подход при изучении языковых явлений и объектов в лингвистике позволяет реализовать неограниченные возможности, которые мы будем применять при дальнейшей работе в избранном направлении. Системный метод позволяет также:

1. Давать новые обобщения (например, установлена связь конвергенции с омонимией).

2. Делать предсказания и открытия посредством как традиционных, так и системных методов.

3. Устанавливать сходство между системами объектов разных родов. Так, В.А. Карпову удалось обнаружить омонимию в кристаллах, а синонимию в биологии – в белках, антонимию – в физике.

4. Решать научные задачи посредством не только традиционных, но и системных методов.

5. Обнаруживать и исправлять ошибки (язык преимущественно конвергентно-дивергентная система, а не система, где конвергенция и дивергенция существуют раздельно).

6. Ставить новые вопросы (например, почему преобладает полисуффиксация над полипрефиксацией? и другие в рамках словообразования – нелинейность словообразования в русском языке и линейное нанизывание аффиксов в турецком?).

7. Усилить математизацию, диалектизацию и системологизацию науки, в частности в лингвистике, где пока доказательства с помощью математики – редки.

Данные положения показывают, что этот метод может быть применен к любому объекту изучения и для создания интеллектуальных систем. Такой универсализм обеспечивается двумя моментами: 1) объяснительной и созидательной мощью ОТСУ в виде системного идеала, системного метода и системной парадигмы; 2) всеобщностью симметрии-асимметрии-диссимметрии как свойств материальных и материально представимых объектов мира.

Использование системного метода как основного метода ОТСУ при обучении языку дает возможность подытожить результаты исследования словообразования как языкового объекта.

Прежде всего, надо логически обосновать и вывести объект исследования как некоторую систему. Это можно реализовать через построение матриц, плюс-минусовых кубов, дающих представление о явлениях конвергенции и дивергенции на уровне словообразования. Морфемика в данной классификации представлена как система систем с разной степенью автономности, разной мощностью множеств на каждом уровне.

Что касается преподавания русского языка как иностранного, то следует подчеркнуть недостаток традиционного метода, который проявляется в том, что, сталкиваясь с указанной ранее неполнотой

знаний на любом уровне, иностранцы вынуждены моделировать недостающие элементы системы при приеме и декодировании чужого сообщения или при создании собственного несистемно. Язык-система располагает для такого моделирования своим инвентарем в виде схем порождения слов и предложений, который и надо предлагать для изучения студентам-иностранцам. Вторым моментом, связанным с моделированием как следствием неполноты, мы считаем то, что моделирование в массовом порядке осуществляется с помощью аналогии. Именно симметрично-асимметричное моделирование на уровне словообразования позволяет создавать множество объектов своего уровня, или предполагать их существование в прошлом, помогает и омонимия.

Омонимия присуща всем языковым объектам. Первый основной момент рассуждения состоит в том, что омонимия является объектом-системой, так как содержит все системообразующие категории: множество первичных элементов (омонимов), множество отношений единства и противоположности (одна и разные части речи, а внутри части речи – одна и разные грамматические позиции, один и разные корни), множество законов композиции в омонимическом узле (слои и связки). Второй момент, связанный с первым, – омонимизация является процессом-системой, что гарантирует обнаружение действия части или всех законов преобразования объектов-систем, приводящее к омонимизации. В третьих, омонимичность является отношением-системой. Следуя этим трем положениям, мы с необходимостью получим системные результаты.

Мы логико-математическим путем доказали необходимость существования этого феномена для любого языка мира, использующего триаду понятий – звук-буква-фонема и построили типологию омонимии в виде семи разновидностей [2]. При системном подходе омонимии можно определить как симметрию формы и асимметрию содержания (ср. прибыли – глагол и однокоренное существительное, омонимичное не только с глаголом, но и внутри собственной парадигмы).

В работе «Омонимия как системная категория языка», выполненной в русле работ НИЛ ТиПЛ [1; 2], в деталях рассмотрены проблемы симметрично-асимметричного характера омонимии. Поэтому здесь мы рассмотрим лишь некоторые аспекты, которые не нашли отражения в монографии и публикациях и будут касаться словосложений и аббревиатур.

Среди них важнейшим является мерность межпарадигматической омонимии. В случае полисемии (многозначности) все формы сопоставляемых слов совпадают, именно поэтому иногда сложно отличить омонимию от многозначности (ср. ключ – источник и инструмент для открывания замка). При омонимии такие явления редки (пион – цветок и пи-мезон). Чаще всего находится несколько неомонимичных форм,

что дает право говорить о том, что в разряде существительных максимум омонимичности достигает 24 форм при минимуме в 2 формы, как это видно из сопоставления двух парадигм, представленных в таблице 2.

Таблица 2 – Парадигматическая и межпарадигматическая омонимия

	Единственное число		Множественное число	
	М. р.	Ж. р.	М. р.	Ж. р.
И.	змеЙ	змеЯ	змеИ	змеИ
Р.	змеЯ	змеИ	змеЕВ	змеЙ
Д.	змеЮ	змеЕ	змеЯМ	змеЯМ
В.	змеЯ	змеЮ	змеЕВ	змеЙ
Т.	змеЕМ	змеЁЙ	змеЯМИ	змеЯМИ
П.	змеЕ	змеЕ	змеЯХ	змеЯХ

Из 24 падежных форм неомонимичными являются только две, что говорит о высокой степени омонимичности и маркированности этих форм (омонимичные формы в примерах выделены заглавной буквой). Новым в знании об омонимии является понятие «омоузла» как системы своего рода. Суть понятия «омоузел» заключается в том, что он представляет внутриклассовую (омонимичные формы принадлежат к одной части речи) и межклассовую омонимию. Эта суть станет достаточно понятной после рассмотрения примера одного омоузла.

Омоузел как система своего рода

дно – дне
 день – дне; день:день
 дело – дело:дело; дел; дЕла:делА
 деть – день:день; дело:дело; дел; дела:дела; дЕли
 делить – делИ

Нулевой столбец омоузла представляет исход в виде словарных форм представления (в словаре дается именительный падеж единственного числа и форма инфинитива для глаголов). Следующие столбцы представляют омонимичные связки в виде форм разных парадигм.

Здесь мы имеем 5 слоев и 6 омосвязок мощностью от двух до пяти форм. Омонимия при таком представлении базируется на линейной симметрии начал слов столбца и асимметрией финалей. В следующих за исходом столбцах имеет место симметрия форм и асимметрия значений. Если столбец исхода представляет словообразование (разные слова), а строки представляют словоизменение, то целый омоузел дает совмещенное представление о двух главных процессах в языке, приводящих к омонимизации целостной системы.

Представлением омоузла как системы своего рода, а именно, системы узлов (изоузел, гомоузел и т.п.), мы хотим показать, что омонимия сложных слов качественно отличается от омонимии слов, созданных аффиксальным способом. Здесь не будет узлов вышеуказанного вида, а для двусложных слов

будут веерообразные открытые системы, ориентированные влево или вправо относительно омонимичного компонента. Для трехкомпонентных будет не две, а семь подсистем.

Таблица 3 – Веерообразное представление омонимии

	-бат	проф-	
	-кор	вол-	
	-див	воен-	
ком-	-банк	глав-	Ком
	-партия	парт-	
	-вуз	гор-	
	-бриг	испол-	
	

Особый пласт в словообразовании представляет образование сложных слов. Омонимия сложных слов, связывается с их структурными особенностями, так как в обычных словах одна упаковка может быть омонимичной и неомонимичной, а сложные слова и аббревиатуры как минимум двухкомпонентны. Этот момент с необходимостью требует рассмотрения таких общесистемных параметров как левизна, правизна и лево-правизна [2].

Более того, освоение омонимией элементов сложных слов ведет к дальнейшей экспансии омонимии, она начинает проявляться на уровне «полнозначное слово» – «элемент сложного слова» или «полнозначное простое слово» – «полнозначное простое слово», что в еще большей мере усложняет процесс автоматической обработки слов.

Омонимия сложных слов и аббревиатур возникает в процессе сжатия двух и более корней в одно слово и характеризуется с содержательной стороны тем, что омонимия системы увеличивается, т.е. сложные слова вносят свою долю в ее омонимизацию. Омонимии двух типов (омонимия элементов сложных слов и омонимия целого слова) сопутствует и полисемия, представляющая разные способы расширения значения.

Анализ материала показывает, что в структурном плане омонимия элементов сложного слова связывается с позицией омонимичного элемента: он может занимать первую или вторую позицию в двусложных словах и все три позиции в трехсложных, при этом двусложные и трехсложные слова могут содержать омонимичный элемент в одной и той же или в разных позициях. Например, *армвоенюрист* – *армглавконцерт*, в первом случае элемент *арм-* обозначает *армейский*, во втором – *армянский*; *Уралчермет* – *гидрометцентр* – в первом случае элемент *-мет* обозначает металл, во втором – метеорологический; *райнарсуд* – *минсудпром* в первом случае элемент *суд-* обозначает *судебный*, во втором – *судостроительный*; *грузкурортстрой* – *грузавтотранспорт* в первом случае элемент *груз-* обозначает *грузинский*, а во втором – *грузовой*; *центризбирком* – *белнацком* – *зубкомпарт* – *помкомвзвода* – *главком* в первом случае элемент

ком- обозначает *комиссия*, во втором – *комитет*; в третьем случае элемент *-ком-* обозначает *коммунистический*, в четвертом случае элемент *-ком-* обозначает *командир*, а в пятом случае элемент *-ком* обозначает *командующий*. Рассмотрение элемента *ком-*, встречающегося в трех позициях (в начале сложного слова, в середине и в конце слова) показывает, что этот элемент имеет самые разные омонимические значения: *командир*, *коммунистический* *коммунальный*, *комиссар*, *командующий*, *коммерческий*, *комисариат*, *комитет*, *коммуникационный* и др. (ср. *комбат* – командир батальона, *командарм* – командующий армией, *компартия* – коммунистическая партия, *горкомхоз* – городское коммунальное хозяйство, *военком* – военный комиссариат, *райвоенкомат* – районный военный комитет, *партком* – партийная комиссия, *инкомбанк* – иностранный коммерческий банк и т.п.). И эта омонимия увеличивается на глазах в пределах одного конкретного элемента и охватывает все большую часть слов. Отмечена омонимия между первым и вторым элементом: *бармен* и *миллибар* (*бар* – ‘небольшой ресторан’, *бар* – ‘единица измерения’).

Омонимия словосложений имеет симметрично-асимметричный характер, поэтому система омонимии в словосложениях может быть представлена в терминах симметрии/асимметрии как комбинаторика трех признаков: формальное и смысловое единство первых корнеэлементов сложного слова (Э1), единство формы и значения опорных корнеэлементов (Э2) симметрия, тождественность слов в плане грамматического значения и его выражения (П). Плюсом обозначим совпадение каждого из перечисленных признаков, а также полное несовпадение. Сделаем небольшое пояснение. Очевидно, что если не только форма, но и значение корнеэлементов в составе разных слов совпадает, то единство этих корнеэлементов сохраняется (например, одинаковые первые корневые компоненты в словах *правомерность*, *правопорядок* – правосудие или *однокоренные* *опорные* компоненты в словах *миноискатель* – *правоискатель*) [3, с. 139]. Но в словах с разными, не совпадающими формально, следовательно, и семантически, компонентами также сохраняется единство отношений – как умножение минуса (несовпадение форм) на минус (несовпадение значений) дает плюс (сохранение отношений).

Тогда минусом обозначим омонимичные отношения между словами или их составляющими элементами (единые формы «+», но разные значения «-»). Первый минус будет обозначать совпадение формы первых элементов при разном значении, второй минус – формальное совпадение семантически различных опорных элементов, третий – формально одинаковое выражение разных грамматических значений. Полносвязная симметрично-асимметричная система теоретически состоит из восьми и только восьми подсистем [2].

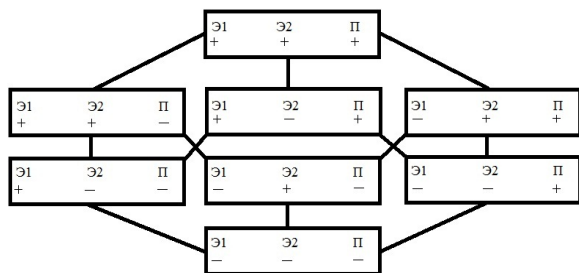


Рисунок 2 – Симметрико-асимметрия корнеэлементов в сложных словах

Самыми наполняемыми будут подсистемы, содержащие один омонимичный признак.

Подсистема 4 заполняется сложными словами, в которых первый элемент обладает лексической омонимией: **авто...1** – 1) ‘автомобильный’: *автогараж, автогонки, автодеталь, автопробег*; 2) ‘автомобиль’: *автобензовоз, автобетономешалка, автозаправочный, автосамосвал*; **авто...2** – *автоматический: автоподача, автосварка, автосцепка*; **авто...3** – ‘самоходный, самодвижущийся’: **автогрейдер, автодрезина**; **авто...4** – ‘свой, собственный, само...’: *автобиография, автографюра; правописание, правонарушение, правобережье*.

Подсистема 3 включает слова с лексически омонимичными опорными элементами и разной или совпадающей предшествующими частями. Опару могут составить слова, в которых выделяется один компонент общий, другой – омонимичный, например: *дву-жильный* (выносливый) – *дву-жильный* (от жиле) – *дву-жильный* (из двух жил). Лексическая омонимия опорных компонентов отмечается в словах с одинаковыми грамматическими характеристиками, образованных от различных мотивирующих основ, например: *богомол* (от молиться) – *мукомол* (от молоть); *пресноводный, полноводный* (от вода) – *счетоводный* (от вести) – *льноводный, пчеловодный* (от разводить) [3, с.140].

Подсистема 2 наполняется словами, различающимися грамматическими характеристиками, с семантически одинаковыми опорными компонентами. Этим условиям удовлетворяют слова с грамматической омонимией опорных компонентов: *льновод, зверовод, цветовод* (от разволить) – *групповод, экскурсовод* (от водить); *воднонос* (одушевленный) – *медонос* (неодушевленный).

Подсистемы нижнего уровня имеют значительно меньшее наполнение.

Подсистема 7, представляющая собой развитие признаков 3-й и 4-й подсистем, непродуктивна: лексическая омонимия обеих частей представлена единственным примером *Минзаг* (Министерство заготовок) — *минзаг* (минный заградитель) [3, 141]. Но ее могут заполнять и сложные слова-омонимы: *микрометр* – ‘инструмент с микротермическим особо точным винтом для измерения контактным

способом’ и *микрометр* – ‘дольная единица длины’; *гелиотрон* – ‘род кустарников’, *гелиотрон* – ‘минерал, геодезический прибор’; *душевнобольной* (прил.) и *душевнобольной* (сущ. м.р.)

Подсистема 6 представляет собой развитие признаков 2-й и 4-й подсистем, практической реализации она не имеет. Это объясняется тем, что омонимия чисто грамматическая предполагает совпадение опорных компонентов (так как два последних признака объединяются общим планом выражения), а сочетание такого опорного компонента с омонимичной предшествующей частью затруднительно. Для иллюстрации придумаем омоним слову *радиозвезда* (звезда — источник радиоизлучения) — *радиозвезда* (звезда на радио; по аналогии с *телезвездой*) Причина ограничения образования подобных пар заключается в неодинаковой валентности омонимичных компонентов [3, 141].

Подсистема 5 объединяет признаки 2-й и 3-й подсистем с лексико-грамматической омонимией опорных компонентов. Первые части слов в таких парах будут различаться: *стеклобой, мордобой, политбой* (сущ.) – *вразной* (наречие); *дискобол* (одуш.) – *пионербол, мотобол* (неодуш.); *дискоболка* (одуш.) – *футболка* (неодуш.); *линкор* (неодуш.) – *юнкор* (одуш.); *лорд-мер* (одуш.) – *секундомер* (неодуш.). Из подсистем нижнего уровня подсистема 5 – самая наполняемая.

Подсистемы 1 и 8, образованные сходными признаками, имеют максимальное и нулевое наполнение. Массив сложных слов, не содержащих омонимичных элементов, составляет 1-ую, самую мощную, подсистему. Подсистема 8, которая содержит три омонимичных признака, может быть заполнена немногочисленными примерами типа *полочка* – ‘маленькая полка’ и *полочка* – ‘половина очка’, *волком* (от волк) и *волком* – ‘волостной комитет’, *ползала* (от ползать) и *ползала* – ‘половина зала’ [1; 2].

Наличие трех типов омонимии в сфере сложных слов (лексической, грамматической, лексико-грамматической) показывает неодинаковую качественную и количественную наполненность. Омонимия присуща более всего сложным существительным, которые, как известно, составляют большинство сложных слов. Вместе с тем отмечаются случаи омонимизации компонентов других частей речи, не являющихся производными от сложных существительных, например: *паро/образовательный, слово/образовательный* (1. образовать – ‘создать’) и *общеобразовательный, самообразовательный* (2. образовать – ‘обучить’); *новообразованный* (от образовать – ‘создать’) и *высокообразованный, малообразованный* (от образовать – обучить).

Использованный выбор классификационных признаков позволил рассмотреть омонимию в рамках композитов и их однопозиционных составляющих. Омонимичными могут быть и

разнопозиционные компоненты, например *пед/совет* и *мото/пед*, *бар/мен* – *кило/бар*, *моль-мал.ютка* – *кило/моль*, *вод/хоз* – *ското/вод*, *глав/реж* – *власо/глав*, *губ/исполком* – *душе/губ*, *кол/договор*, *кол/хоз* – *дрово/кол*, *дыро/кол*, *часы-кулон* – *кулон/вольтный*, *патрон-покровитель* – *электро/патрон*, *фауст/патрон*, *вода/вод* и др.

Можно легко представить, что в аббревиатурах степень омонимичности будет еще большей, так как чисто комбинаторные возможности там возрастают.

Омонимия обычных слов служит базой для появления омонимии своеобразного «второго порядка», вторичной омонимии уже сложных слов. К ней присоединяется и специфическая омонимия, возникающая между неомонимичным элементом сложного слова и другим, возникающая в результате усечения части этого элемента. Так, слово *мел* и лексема *мелодия* не являются омонимами. Но усечение слова *мелодия* до элемента сложного слова *мело-* создает омонимичные отношения.

Вот некоторые примеры первых элементов, являющихся омонимичными: *мело* – мелодия, мел; *пар* – пар, пара; *соц* – социальный, социалистический; *авиа* – авиационный, воздушный; *авто* – автомобильный, автоматический, собственный, самодвижущийся; *архи* – старшинство, высшая степень чего-либо; *био* – относящийся к жизни, биологический; *обще* – общий, обязательный для всех, свойственный, охватывающий отрасли, распространяемый на всех, охватывающий что-либо целиком; *пол* – половина, середина, полномочный; *само* – действие направлено кем-н. на самого себя, действие совершается без посторонней помощи; *сверх* – превышение предела, высшую степень качества; *спец* – специальный, специалист; *средне* – средний период чего-н., срединное положение, средняя норма чего-н.; *суб* – нахождение внизу, подчиненность, второстепенность; *супер* – расположение сверху, главенство, высшая степень качества, вторичность; *теле* – действующий на дальнее расстояние, телевизионный; *транс* – движение через какое-либо пространство, следование за чем-либо, сообщение через посредство чего-либо; *фото* – основанный на действии света, фотографический; *лизо* – часть от лизать (*лизоблюд*) и вещество (*лизокефалин*).

Отмечается омонимия элементов сложных слов и в ряде других случаев, касающихся межъязыковой омонимии в заимствованиях. Например, *паралегия* – *пара-хронизм* – *пара-нефрит*, в первом слове элемент *пара-* обозначает *вдоль* (от гр. *para* – *вдоль*, *plexis* – *связывание* – паралич обеих конечностей), во втором – *вопреки* (от гр. *para* – *вопреки*, *chronos* – *время* – хронологическая ошибка), в третьем – *возле*, *около* (от гр. *para* – *возле*, *около*, *nephros* – *почка* – воспаление околопочечной клетчатки).

В ряде случаев отмечается омонимия вторых элементов. Компонент *-граф* вступает в

омонимические отношения по семантическому признаку (предмет и носитель профессионального признака). Отмечена омонимия по грамматическому признаку «одушевленность-неодушевленность»: *телеграф* и *фотограф*; *путеводитель* и *автоводитель*; *животновод* и *нитевод*. Отмечена лексико-грамматическая омонимия: *богомол* – *мукомол* (от глаголов *молиться* и *молоть*); *треугольный* и *древесноугольный* (от *угол* и *уголь*); *стократ* и *демократ* (от кратность и «кратос» – *власть*); *вещдок* и *плавдок* (от *доказательство* и *док*); *госдача* и *кроводача* (от *дача* и *давать*); *животновод*, *нитевод* и *углевод* (от *разводить*, *проводить* и *водород* – этот случай особо показательный, так как показывает, что сложное слово-калька "водород", введенное в систему, подвергается дальнейшему сжатию, что и приводит к омонимии – ср. ряд *водород* – *углеводород* – *углевод*); *аммолиз*, *автолиз* и *блюдолиз* (*лиз* – *саморастворение* и *усеченная основа* от *лизать*); *аэроплан* и *генплан* (*план* от *планировать* в значении *плавно летать*, *парить* и *план* от *планировать* что-либо); *автократ* и *стократ* (*крат* – *власть* и *крат* от *кратно*). Отмечена омонимия между первым и вторым элементом: *бармен* и *миллибар* (*бар* – *небольшой ресторан*, *бар* – *единица измерения давления*).

Омонимия словосложений имеет симметрично-асимметричный характер, поэтому система омонимии в словосложениях может быть представлена в терминах симметрии/асимметрии и с успехом использоваться в интеллектуальных экспертных обучающих системах.

Библиографический список

- [Головня, 2007] Головня, А.И. Омонимия как системная категория языка / А.И. Головня. Минск: БГУ, 2007.
[Карпов, 1992] Карпов, В.А. Язык как система / В.А. Карпов. Минск: БГУ, 1992.
[Свиридович, 1998] Свиридович, М.В. Системное изучение омонимичных компонентов русских словосложений / М.В. Свиридович // Материалы міжнароднай навукова-тэарэтычнай канферэнцыі «Тэорыя і практыка навучання мовам, літаратурам і рыторыцы ў сярэдняй і вышэйшай школе» Ч.2. Минск: БГУ, 1998. С.139 – 141.

USING OF AFFIXATION IN EXPERT TRAINING SYSTEMS DURING LEARNING OF WORD-FORMATION

Golovnja A.I. *

*Belarusian State University,
Minsk, Republic of Belarus

golovnjaai@bsu.by

Developers of systems of artificial intelligence should find the pros and cons during development of training expert systems which could help foreign students to learn Russian word formation, how to define terms "affixation" and homonymy affixes and their role in the development of intelligent systems, define role of homonymy in the study of word formation.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.032.26

МЕТОД ИНТЕЛЛЕКТУАЛИЗАЦИИ ТЕСТИРОВАНИЯ ИНДИВИДУАЛЬНОЙ ТРАЕКТОРИИ ИЗУЧЕНИЯ СТУДЕНТОМ НЕКОТОРОЙ ПРЕДМЕТНОЙ ОБЛАСТИ ЗНАНИЙ НА ОСНОВЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Козлов О.А., Михайлов Ю.Ф.

*Федеральное государственное научное учреждение
«Институт информатизации образования» Российской академии образования
г. Москва, Россия*

ole-kozlov@yandex.ru

mikhayurij@yandex.ru

В статье приводится описание подхода по формированию теста проверки индивидуальной траектории изучения студентом некоторой предметной области знаний с помощью искусственной нейронной сети.

Ключевые слова: вектор параметров оценки качества усвоения знаний, уровни усвоения знаний, умений, навыков, дидактический элемент модуля знаний, интеллектуальная обучающая система.

ВВЕДЕНИЕ

Рассматривается проблема изменения парадигмы информационного взаимодействия образовательного назначения, осуществляемого в информационно-коммуникационной предметной среде, когда обучающийся взаимодействует с обучающим, с информационным ресурсом некоторой предметной области, посредством информационных и коммуникационных технологий, формируя индивидуальную траекторию изучения некоторой предметной области. Актуальным с точки зрения лично-ориентированного обучения является понятие индивидуальной траектории обучения и изучения. Формирование теста для контроля индивидуальной траектории изучения является задачей со множеством неопределенных параметров, поэтому предлагается для ее решения применять искусственную нейронную сеть.

Интеллектуализация теста для контроля индивидуальной траектории изучения предметной области знаний

Индивидуальная траектория обучения – это специфический содержательный и операциональный состав обучения и последовательность его усвоения, подбираемая под конкретный образовательный запрос обучаемого в соответствии с его возможностями. Индивидуальная траектория обучения отражает уникальный путь, которым личность обучаемого должна двигаться к целям обучения. Выбор одной из нескольких

десятков и сотен возможных траекторий обучения формирует индивидуальную траекторию изучения студентом некоторой предметной области знаний. Одним из вариантов решения указанной задачи являются нейросетевые технологии.

Программа дисциплины строится по модульному принципу, разработанному и представленному в монографии доктора педагогических наук О.А. Козлова «Теоретико-методологические основы информационной подготовки курсантов военных учебных заведений» [Козлов, 2001].

В каждом модуле выделяются элементы знаний, представленные в учебной программе, устанавливаются структурные и семантические связи между элементами и разрабатываются семантические модели знаний для этих модулей. Связи между элементами знаний реализуются по гипертекстовой технологии, в виде ссылок. В качестве ссылки берется ключевое слово, терм, понятие, которое присутствует в структурно связанных между собой элементах знаний [Михайлов, 2001].

Гипертекст позволяет формировать синонимические и полисемантические связанные системы понятий. Гипертекст представляет собой нелинейную модель знаний, которая позволяет самому курсанту организовать управление обучением, получить доступ к учебным материалам, представленным в различной форме. Преподавателю просто организовать управление процессом обучения, используя аппарат ссылок в тех местах модуля знаний, которые он, в

соответствии с поставленными дидактическими целями обучения, определил для контроля усвоения.

При изучении программного материала усвоение отдельных учебных элементов различается по трем уровням, которые задаются в государственной программе [Козлов, 1999], [Роберт, 2009]. Первый уровень усвоения означает наличие у курсантов знаний – знакомств, умение выделить и отличить среди нескольких понятий то, которое им предъявлено для опознания. Этот уровень формируется проблемными ситуациями типа «выбор», вес ситуации, вопроса и ответа, которые генерируются такой ситуацией, равен единице. Второй уровень усвоения означает наличие у курсантов знаний – копий, то есть способности самостоятельно охарактеризовать явление, дать формулировку закона, умение решать типовые задачи по типовому алгоритму, или применить типовой алгоритм в новых условиях. Этот уровень формируется проблемной ситуацией типа «неопределенность решения», вес ситуации определяется количеством продуктивных операций, которые надо выполнить для ответа на контрольный вопрос, либо числом термов контролируемого модуля знаний. Третий уровень предполагает наличие у курсанта знаний – умений, знаний – навыков, то есть умение решать типовые задачи в условиях неопределенности постановочной части задачи, либо формировать новый алгоритм решения в виде определенной комбинации типовых алгоритмов, либо организация собственного пути поиска нового знания. Этот уровень формируется проблемной ситуацией типа «поиск выхода», вес ситуации определяется количеством продуктивных операций, которые надо было выполнить для ответа на контрольный вопрос.

Для контроля предлагается тест из вопросов, по числу термов, которые надо было усвоить по модулю. Тест содержит вопросы по трем уровням усвоения.

На основе анализа научной литературы по рассматриваемой проблеме [Козлов, 1999], [Роберт, 2009], [Ширшов, 2002], [Brusilovsky, 1993], [Corbett и др., 1992] умения являются реализованными на практике знаниями, навыки представляют собой закрепленные умения. Поэтому успешный ответ на вопрос теста, связанный с умениями и навыками усиливает вес соответствующего знания. В нейронной сети усиление синаптической связи реализуется через обратные связи. С другой стороны, отрицательный ответ на вопрос теста, связанный с умениями и навыками, уменьшает вес соответствующего знания.

С целью оценки начальных знаний по выбранному или заданному учебному материалу применяются ряд тестовых заданий, формирующих его рейтинг и другие характеристики. Для анализа возникающих ошибок ведется база данных об ошибках обучаемого, с указанием весового коэффициента ошибки, заполняемая автором курса. При этом задание разбивается на подзадачи (для

конкретизации), а возникающая точка n -мерного шара ошибок анализируется системой. Такая организация позволяет избежать случайного совпадения ответа пользователя с правильным ответом.

Задача оценки знаний формулируется в следующей постановке. Создается тест, вопросы которого относятся к одному из выбранных разделов. Каждый из этих вопросов предполагает выбор одного из нескольких вариантов ответов и кодируется 0, если ответ выбран неправильно, или 1, если правильно. Заметим, что любой сложный вопрос можно представить в виде конечной последовательности из M элементарных высказываний, ответами на которые являются бинарные переменные x_1, \dots, x_M . При этом значимость коэффициентов l_1, \dots, l_M переменных x_1, \dots, x_M может быть различной, определяется весом ситуации и закладывается в синаптический вес соответствующего нейрона.

В известных подходах [Саймон, 2006], [Горбань, 1990] предварительно задается архитектура нейронной сети (как правило, выбирается многослойный персептрон с заданным количеством нейронов и слоев). Далее формируется обучающая выборка X, Y :

Возможный вариант трехслойной нейронной сети представлен на рисунке 1.

Входной вектор тестирования структурирован по уровням; уровень знаний X^3 , уровень умений X^y , уровень навыков X^H . Число элементов уровня определяется числом термов, которые надо было усвоить по модулю.

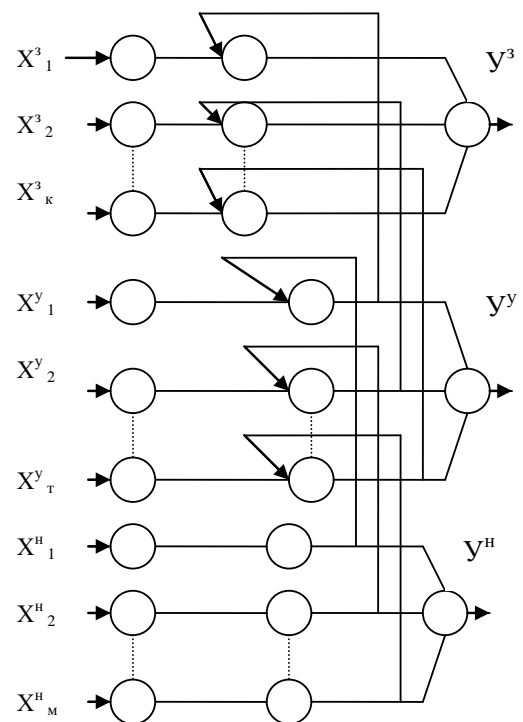


Рисунок 1 – Архитектура нейронной сети.

В качестве алгоритма обучения нейронной сети можно выбрать алгоритм обратного распространения ошибки с настройкой сигмоидальной функции активации [Савченко, 2010]. Сущность данного модифицированного алгоритма состоит в коррекции весовых коэффициентов согласно рассмотренным правилам.

Для рубежного контроля часто используются системы рейтинговых оценок [Козлов, 2001], [Ширшов, 2002].

Рейтинг – число, классифицирующее успеваемость курсанта по теме, определяется как сумма взвешенных рейтинговых чисел по каждому элементу модуля знаний, на который дан правильный ответ.

Рейтинговое число по теме определяется следующим образом. Вес одного существенного действия принят за единицу. Число баллов за вопрос определено как произведение весов на число действий. Весовая характеристика вопроса определяет долю знания вопроса в общем объеме знания (теста).

В этом случае возможный вариант трехслойной нейронной сети представлен на рисунке 2.

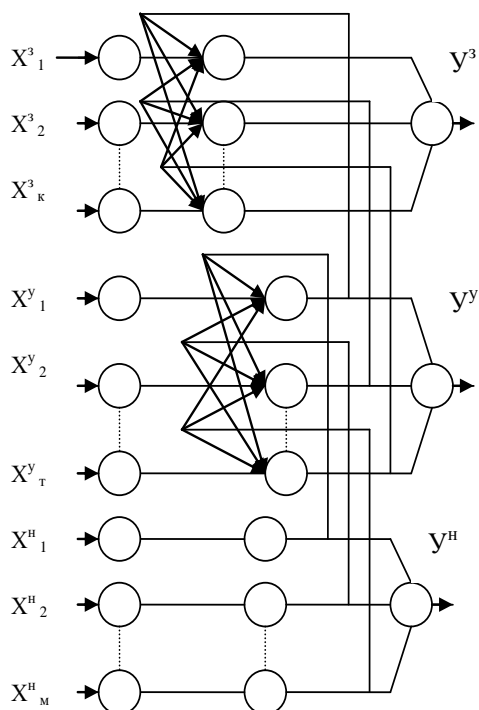


Рисунок 2 – Архитектура нейронной сети тестирования.

Дилемма стабильности – пластичность является важной особенностью обучения, необходимо обучать сеть, моделирующую студента, курсанта новым явлениям (пластичность) и в то же время сохранить стабильность, чтобы существующие знания не были стерты или разрушены. Поставленную проблему стабильности - пластичности решают нейронные сети, построенные в соответствии с теорией адаптивного резонанса. К таким сетям относятся сети ART,

которые обеспечивают решение задачи классификации. Входной вектор классифицируется в зависимости от его схожести с одним из образов, запомненных ранее при обучении.

Результаты тестирования формируют входной вектор X для сети ART. В этом случае сеть модифицируется (рисунок 3).

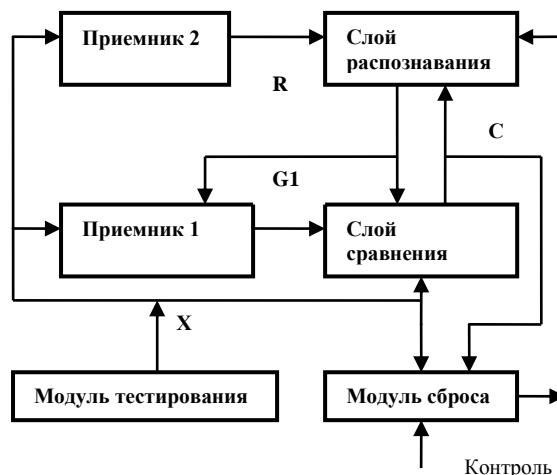


Рисунок 3 – Структура сети ART

Параметры обновленной модели обучаемого поступают на вход нейронной сети, которая по аналогии с работой человеческого мозга, может выбрать один из нескольких сотен вариантов очередного этапа индивидуальной траектории обучения. В зависимости от текущего уровня знаний по изучаемой дисциплине движение по траектории изучения дидактической единицы можно разбить на три основных направления: возврат назад по траектории к предыдущему дидактическому элементу изучаемой дидактической единицы учебного курса, если обучаемый неудовлетворительно справился с предлагаемыми тестовыми заданиями по изучаемой теме или разделу; движение вперед к новому дидактическому элементу, следующей порции учебного материала, определенной программой курса. В случае если предыдущая тема или раздел учебного курса были освоены на хорошем или отличном уровне выполняется движение вперед к новой дидактической единице учебного материала. Каждое выделенное направление движения по индивидуальной траектории изучения включает несколько вариантов, позволяющих индивидуализировать и дифференцировать процесс обучения.

ЗАКЛЮЧЕНИЕ

В данной статье предложен метод интеллектуализации теста для контроля индивидуальной траектории изучения студентом, курсантом некоторой предметной области знаний, основанный на аппарате искусственных нейронных сетей, описывающих состояние и поведение

студента во время изучения дидактической единицы модуля знаний.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Горбань, 1990] Горбань А.Н. Обучение нейронных сетей. - М.: СП ПараГраф. 1990.

[Козлов, 1999] Козлов О.А. Развитие методической системы обучения информатике курсантов военно-учебных заведений Министерства обороны Российской Федерации. Дисс.... докт. пед. наук. – Серпухов. 1999.

[Козлов, 2001] Козлов О.А. Теоретико-методологические основы информационной подготовки курсантов военно-учебных заведений. Монография. - М.: МО, 2001. – 328 с

[Михайлов, 2001] Михайлов Ю.Ф. Технология информационной подготовки курсантов в условиях моделирования экстремальных ситуаций профессиональной деятельности. Дисс.... канд. пед. наук. – Москва. 2001.

[Роберт, 2009] Роберт И.В. Теория и методика информатизации образования (психолого-педагогический и технологический аспекты), 2-е издание, дополненное.– М.: ИИО РАО, 2009.

[Саймон, 2006] Саймон Хайкин. Нейронные сети. Полный курс. 2-ое изд. 2006. — М., СПб., Киев: Вильямс

[Савченко, 2010] Савченко Е.Ю., Мусакулова Ж.А. Настройка сигмоидальной логистической функции активации в алгоритме обратного распространения. //Материалы Международной конференции «проблемы управления и информационных технологий». Проблемы автоматизации и управления. №1 Бишкек, 2010-С.241

[Ширшов, 2002] Ширшов Е.В. Применение технологий искусственного интеллекта в учебном процессе вуза. Образовательные технологии для новой экономики: Сб. материалов международной конференции – М.: Изд-во МЭСИ, 2002. – с.340 – 348.

[Brusilovsky, 1993] Brusilovsky P. Student as user: Towards an adaptive interface for an intelligent learning environment. In Proceedings of World Conference on Artificial Intelligence and Education, AI-ED'93, 386-393. Charlottesville: AACE.

[Corbett, Anderson, 1992] Corbett A.T., Anderson J.R. Student modeling and mastery learning in a computer-based programming tutor. In Proceedings of the Second International Conference on Intelligent Tutoring Systems, ITS'92, 413-420. Berlin: Springer-Verlag

METHOD OF INTELLECTUALIZATION OF TESTING INDIVIDUAL TRAJECTORY OF STUDY OF THE STUDENT OF A CERTAIN DOMAIN OF KNOWLEDGE ON THE BASIS OF ARTIFICIAL NEURAL NETWORKS

Kozlov O.A., Mikhailov J.F.

*Institution Of The Russian Academy Of Education
"Institute Of Informatization Of Education",
Moscow, Russia*

ole-kozlov@yandex.ru

mikhayurij@yandex.ru

The article describes the approach to the formation of tests of individual trajectory of study of the student of a certain domain of knowledge with the help of artificial neural network. Key words: vector of parameters of quality assessment of learning outcomes, levels of mastering of knowledge, skills, didactic element of the module of knowledge, intellectual training system.

INTRODUCTION

Addresses the problem of changing the paradigm of information interaction of the educational purpose,

carried out in the information-communication of the subject environment where the learner interacts with the training, the information resource of a certain subject area, by means of information and communication technologies, forming the individual trajectory of studying a certain subject area. Topical from the point of view of personality-oriented education is the notion of individual trajectory of teaching and learning. The formation of the test for the control of the individual trajectory of study is a task with a lot of uncertain parameters, it is therefore proposed to her decision to apply artificial neural network

MAIN PART

The program of the discipline is based on modular principle, developed and presented in the monograph doctor of pedagogical Sciences. O.A. Kozlova [Kozlov, 2001].

In each module are allocated elements of the knowledge presented in the training programme, shall be established by the structural and semantic connections between the elements and development of semantic models of knowledge for these modules [Mikhailov, 2001].

When studying the program material assimilation of some educational elements varies in three levels, which are defined in the state programme [Kozlov, 1999], [Robert, 2009]. The first level of learning means the presence of cadets of the knowledge – Dating. The second level of learning implies that students knowledge of. The third level is the presence of the trainee's knowledge, skills, knowledge, skills, i.e. the ability to solve typical tasks in the conditions of uncertainty of the production part of the task.

To control the proposed test of the issues, according to the number of terms that need to be learned in the module. Therefore, a successful response to the question of the test related to skills and increases the weight of the relevant knowledge. In the neural network, the strengthening of the synaptic connection is realized through backward linkages.

Pre-set architecture of the neural network (as a rule, select the multi-layered perseptron with the given number of neurons and layers). Then formed a learning sample X, Y.

The input vector testing is structured into levels; the level of knowledge X^3 , skill level X^Y , the level of skills X^H . The number of elements of the level defined by the number of terms that need to be learned in the module.

Conclusion

In this article the method of intellectualization of tests for monitoring individual trajectory of study of the student, the student of one of subject area knowledge, based on the apparatus of artificial neural networks, describing the situation and behaviour of students during the study of didactic unit of module knowledge.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

СОЗДАНИЕ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНОГО ПРОСТРАНСТВА ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ НА ОСНОВЕ СЕРВИСНО- ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ И МНОГОАГЕНТНОЙ ТЕХНОЛОГИИ

Тельнов Ю.Ф., Казаков В.А., Трёмбач В.М.

*Московский государственный университет экономики, статистики и информатики (МЭСИ),
г.Москва, Россия*

ytelnov@mesi.ru

vkazakov@mesi.ru

vtrembach@mesi.ru

Представлен подход к организации интегрированного информационно-образовательного пространства высшего учебного заведения на основе многоагентных и сервисно-ориентированных архитектур, направленный на обеспечение эффективного использования распределенного научно-образовательного контента в процессе освоения и генерации нового знания.

Ключевые слова: информационно-образовательное пространство; репозиторий объектов; онтология; сервисно-ориентированная архитектура; многоагентная технология

ВВЕДЕНИЕ

Создание информационно-образовательного пространства (ИОП) направлено на повышение актуальности и эффективности коллективного использования в образовании и научных исследованиях распределенного в вычислительной сети контента, имеющего различия в стандартах, платформах, структуре и форме представления. Интеграция в ИОП разнородных и распределенных источников информации и знаний в ИОП ориентировано на эффективную организацию процесса освоения и генерации нового знания на основе многоагентного взаимодействия субъектов образовательной и исследовательской деятельности (преподавателей, научных сотрудников, студентов).

Под интегрированным информационно-образовательным пространством будем понимать организационно-технологическую инфраструктурную среду функционирования системы электронного обучения, которое включает в себя распределенный контент в виде электронных образовательных ресурсов (ЭОР) и набор ИТ-сервисов, обеспечивающих в режиме удаленного доступа выполнение функций по информационному и программному обеспечению (поддержке) образовательной деятельности [Тельнов и др., 2012b].

Разработка методов и средств создания интегрированного информационно-образовательного пространства базируется на принципах построения систем управления знаниями (СУЗ), которые применяют концепции объектной

организации знаний, многоагентных технологий и онтологического подхода. Вместе с тем, в известных методологиях создания и применения СУЗ [Frank et al., 2000], [Schreiber et al., 2000], [Mentras et al., 2002], [Кудрявцев, 2010] не реализован прагматический аспект использования информационного пространства, реализуемый в первую очередь через модель компетенций. Кроме того, в этих методологиях не отражена специфика применения интеллектуальных ресурсов (источников знаний) в образовательных и исследовательских целях, и в частности коллективный характер проведения обучения и исследований с помощью многоагентных технологий.

Анализ методологий создания интеллектуальных обучающих систем и сред [Домрачев, 1996] и [Brusilivski, 1996] показал, что они не достаточны развиты с позиции использования разнородных источников знаний, организации динамического процесса освоения и генерации знаний и не базируются на современных сервисно-ориентированных архитектурах, что очень важно для создания ИОП.

Применение технологий Web-сервисов в информационно-образовательном пространстве становится актуальным инструментом, который может расширить механизм динамических многоагентных систем в ИОП. Для реализации поиска нужных конкретному потребителю сервисов и определений интерфейсов этих сервисов, предусмотрен механизм публикации метаописаний (в произвольной форме) и спецификаций

интерфейсов сервиса на языке WSDL [W3C, 2001] в регистре UDDI (Universal Description & Discovery & Integration) [OASIS, 2012]), для которого разработан специальный стандарт. Однако, с точки зрения поиска и композиции образовательных и исследовательских сервисов в ИОП этот инструментарий семантически не достаточен. Ряд авторов [McIlraith et al., 2002] указывают на необходимость проведения исследований по расширению данного стандарта на основе онтологического подхода, как, например, это предлагается [W3C, 2007]. Вместе с тем, реальные результаты по этому вопросу до сих пор не получены.

Таким образом, для создания информационно-образовательного пространства, требуется комплексное применение методов объектной технологии представления знаний в системах управления знаниями на основе онтологического и компетентностного подходов и многоагентной технологии с использованием сервисно-ориентированной архитектуры [Гаврилова и др., 2006], [Трембач, 2012], [Тельнов и др., 2012b].

Целью данной работы является разработка методов и средств создания информационно-образовательного пространства на основе динамической интеллектуальной системы управления процессами [Тельнов и др., 2010] и [Тельнов и др., 2012a], которые обеспечивают распределенное хранение и совместное использование разнородного научно-образовательного контента всеми участниками научно-образовательного процесса для проведения научных исследований, методических разработок и обучения.

Для достижения этой цели в докладе предлагаются методы и средства создания ИОП, которые направлены на решение следующих задач:

- разработка принципов и концепций семантического моделирования ИОП на основе развития объектной парадигмы представления знаний и многоагентной технологии взаимодействия субъектов научно-образовательного процесса;
- разработка семантических методов интеграции разнородных источников знаний, описания образовательных и научных сервисов на основе онтологического подхода;
- разработка методов представления объектов знаний в ИОП на основе сервисно-ориентированной технологии («знания как сервисы»);
- разработка методов композиции сервисов и объектов знаний в соответствии с моделями компетенций и потребностей образовательных и исследовательских программ;
- разработка многоагентной технологии применения сервисов в ИОП для решения образовательных и исследовательских задач;

- разработка методов коллективного решения образовательных и исследовательских задач в ИОП на основе многоагентной технологии.

Применение методологии создания интегрированного образовательного пространства, основной характеристикой которой является организация многоагентного коллективного решения образовательных и исследовательских задач с использованием сервисных архитектур на базе интегрированного репозитория объектов, позволит повысить эффективность научно-образовательной деятельности высших учебных заведений.

1. Требования к динамической интеллектуальной системе управления процессами в информационно-образовательном пространстве

Для совместного использования учебных или исследовательских объектов преподавательскими, научными или студенческими коллективами в работе осуществляется развитие методов многоагентных технологий в условиях сервисно-ориентированной организации как репозитория объектов знаний (Знания, как сервисы), так и различных процедур доступа к ним (Процедуры, как сервисы).

В качестве основных научно-образовательных сервисов выделяются:

- Формирование учебного плана с учетом индивидуальной траектории обучения,
- Формирование УМК для индивидуальной траектории обучения,
- Формирование практикума, включая постановку практической задачи с учетом профиля обучающегося, отбор проектных решений по прецедентам,
- Оценка качества полученных результатов (результатов интеллектуальной деятельности) на основе выбора и выполнения тестов и контрольных заданий для проверки знаний и умений.
- Формирование учебных коллективов на основе создания, обновления и отбора профилей преподавателей, сотрудников и обучающихся.

К классам программных агентов, реализующих отдельные функции участников научно-образовательного процесса, относятся:

- Преподаватели (научные сотрудники),
- Студенты (обучающиеся),
- Руководители подразделений (заведующие кафедрами, руководители научных подразделений),
- Менеджеры научно-образовательного процесса (дирекция учебных институтов-факультетов, менеджеры),
- Администраторы научно-образовательного пространства (ИТ-службы, службы управления знаниями, службы управления контентом, администраторы баз данных).

В рамках реализации научно-образовательных сервисов требуется использование различных видов контента, различных источников знаний. Для представления источников знаний в информационно-образовательном пространстве предлагается метод объектного представления знаний в репозитории учебных и исследовательских объектов на основе совершенствования методов выделения и структурирования объектов, их семантической разметки.

Основными элементами базы знаний (репозитория) являются учебные и научные объекты, соответствующие дидактическим единицам предметной области, в качестве которых выделяются:

- Объекты знаний: раздел – тема – дидактическая единица предметной области;
- Объекты умений: практические действия и автоматические навыки работников (обучающихся);
- Объекты компетенций, как сложных объектов, представляющих наборы знаний и умений для выполнения профессиональных функций (решения задач);
- Объекты описания сферы практических приложений;
- Объекты проектных решений или результатов интеллектуальной деятельности, полученных в ходе решения ранее поставленных задач;
- Объекты контрольных примеров (ситуаций);
- Объекты тестов (контрольных заданий);
- Объекты описания методов и инструментальных средств;
- Объекты описания нормативных документов;
- Объекты описания профилей преподавателей (научных сотрудников);
- Объекты описания профилей студентов (обучающихся).

Добавление новых элементов в единый репозиторий выполняется либо непосредственно авторами контента, либо преподавателями, использующими эти объекты для преподавания дисциплин.

Для интеграции разнородных источников знаний в информационно-образовательном пространстве предлагается использовать методы онтологического инжиниринга. Новизна предлагаемого подхода заключается в сопряжении онтологии учебных и исследовательских объектов, построенной в соответствии со спецификацией стандарта SCORM, онтологии предметной области и онтологии компетенций [Тельнов и др., 2012b], что обеспечит построение адекватных конкретной образовательной или исследовательской задаче траекторий изучения объектов и повторное использование учебных, методических и научных материалов в научно-образовательных сервисах ИОП. Структура взаимодействия онтологий и ЭОР

представлена на рисунке 1.

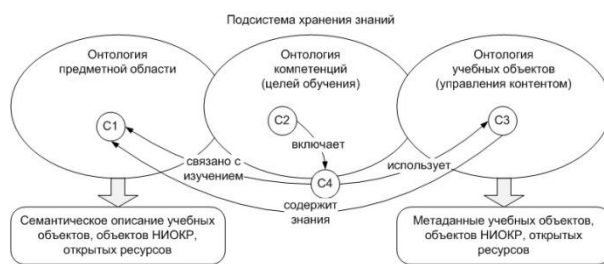


Рисунок 1 – Взаимосвязь онтологий и информационно-образовательных ресурсов

Для обеспечения поиска учебных материалов в ИОП необходимо хранить в базе знаний не только сами источники знаний, но и метаданные, описывающие эти источники. При загрузке контента в репозиторий заполняются основные обязательные поля метаданных на основе онтологии учебных объектов, а также семантическое описание объектов, формализованное на языке ObjectLogic [Ontoprise, 2010] с использованием онтологии предметной области.

2. Организация репозитория учебных объектов на основе онтологического подхода

Необходимость интеграции разнородных учебных объектов и обеспечение повторного использования учебных, методических и научных материалов обусловило разработку единой модели представления контента в открытой образовательной среде (SCORM), которая может быть использована в инновационных образовательных технологиях [Тельнов и др., 2009].

Учебный объект, согласно определению IEEE LTSC, представляет собой пакет учебных материалов, который может использоваться многократно. При этом на него можно делать ссылки при использовании соответствующей обучающей технологии. Для реализации указанных возможностей стандартом SCORM предусматривается следующая структура УО:

<Учебный объект>=<Декларация УО><Учебный блок>

Учебный блок представляет собой непосредственно учебный материал, используемый при обучении, в то время как декларация не используется непосредственно в обучении, но является неотъемлемой частью УО, содержит всю необходимую служебную информацию для обеспечения использования объекта:

<Декларация УО> = <Метаданные УО><Порядок расположения частей УО><Ссылки на ресурсы><Декларации частей УО>

В случае использования составных, сложных объектов в декларацию входят дополнительные элементы, описывающие порядок расположения частей УО, их декларации. Ссылки на ресурсы

обеспечивают доступ к учебным блокам УО в репозитории.

Учебный объект может иметь как версии исполнения разными авторами, так и дополнительные части, ориентированные на различные уровни усвоения и уровни развития компетенции. В случае компетентностно-ориентированного представления объекта в качестве его атрибута должен задаваться идентификатор компетенции.

Основным элементом декларации, представляющим сведения об УО, является блок метаданных:

<Метаданные УО> = <Общие метаданные УО><Семантическое описание УО>

Метаданные используются для обеспечения релевантного отбора учебных материалов, обмена материалами между разными системами, автоматической компиляции индивидуальных учебных пособий. Этот блок включает как общие метаданные, определенные стандартом (всего более 80 атрибутов УО, файлов, учебных ресурсов, разделов), так и дополнительные характеристики учебного объекта:

<Общие метаданные УО> =
<Идентификатор><Название><Язык><Аннотация>
<Авторы><Версия><Технические
данные><Цена>...

Стандартами определен атрибут метаданных «ключевые слова» однако для целей реализации возможностей ИОП он выделен в отдельный блок – «Семантическое описание УО».

<Семантическое описание УО> = <Ключевые слова>

Ключевые слова представляют собой описание содержания ресурсов, которое не может быть определено другими атрибутами метаданных. Это описание представляет собой набор понятий или наименований конкретных объектов предметной области, с которыми связан УО.

<Ключевые слова> = <Понятия | Имена объектов>

Таким образом, для формирования семантического описания УО необходимо определить модель предметной области, в рамках которой были бы выделены основные понятия, отношения между ними, а также объекты, соответствующие этим понятиям, то есть необходимо построить онтологию предметной области.

В основе онтологии предметной области лежит набор понятий S , которым могут соответствовать некоторые реальные объекты предметной области I . Понятия, классифицированные по признаку «род-вид», образуют таксономию [Гаврилова и др., 2006]. Дополненная атрибутами (свойствами) и отношениями между понятиями, а также правилами

вывода эта таксономия образует онтологию.

Рассмотрим структуру онтологии предметной области O^{SA} , построенную в соответствии с моделью [Тельнов и др., 2012b]:

$O^{SA} = (C, A, R, F)$ - онтология предметной области,

$C = \{C^{top}, C^P, C^{Subj}, C^{Obj}\}$ - множество понятий предметной области, где

C^{top} – корневое метапонятие онтологии,

C^P – множество понятий, определяющих процессы и задачи предметной области,

C^{Subj} – множество понятий, определяющих субъектов предметной области: организационные единицы, персоны,

C^{Obj} – множество понятий, определяющих объекты предметной области;

A – множество атрибутов, представляющих числовые и литеральные характеристики понятий предметной области,

R – множество отношений между понятиями предметной области,

F – множество функций интерпретации, определяющих взаимозависимости понятий предметной области.

Для обеспечения возможностей использования УО в виртуальном информационно-образовательном пространстве также создаются:

- онтология управления контентом O^{SCORM} , содержащая понятия, описанные в стандарте SCORM. Ключевым понятием онтологии управления контентом является понятие учебного объекта, для которого определяются атрибуты <Общие метаданные УО> и возможные связи с другими понятиями (авторы, языки и т.д.);

- онтология компетенций $O^{Competence}$, которая может быть представлена как связующее звено между онтологиями управления контентом и онтологией предметной области.

В настоящее время широкое распространение получила трактовка компетенции как стремления и готовности применять знания, умения, навыки для успешной деятельности в определенной области. Онтология $O^{Competence}$ создается на основе модели компетенции [Зиндер и др., 2011], содержит таксономию компетенций, служит для описания текущего и целевого уровней компетенций, определения знаний, умений и навыков, связанных с освоением компетенций, формирования траектории освоения компетенций в ходе обучения.

Для каждой компетенции определяется набор компонентов, которыми должен владеть обучающийся с определенным уровнем освоения компетенции:

<Компетенция> = <Идентификатор><Название компетенции><Знать><Уметь><Владеть><Уровень компетенций по Блуму>

Знания, умения и навыки описываются в терминах онтологии O^{SA} и содержат ссылки на понятия или конкретные объекты предметной области, которые необходимо знать или которыми необходимо овладеть для успешного освоения компетенции.

<Знать> = <Понятия | Имена объектов>

<Уметь> = <Понятия | Имена объектов>

<Владеть> = <Понятия | Имена объектов>

Для определения глубины изучения представленных понятий используется атрибут компетенции, характеризующий ее уровень по таксономии Б.Блума:

<Уровень компетенций по Блуму> =

<1-знать | 2-понимать | 3-применять | 4-анализировать | 5-синтезировать | 6-оценивать >

При загрузке учебного объекта в репозиторий задаются основные обязательные поля метаданных, соответствующие атрибутам понятия учебного объекта в онтологии O^{SCORM} , а также раздел <Семантическое описание УО>. Этот раздел содержит указатели на реальные объекты предметной области и понятия, представленные в онтологии O^{SA} . Таким образом, каждому i -му УО ставятся в соответствие подмножества $I_i \subseteq I$ и $C_i \subseteq C$. После этого актуализируются данные в репозитории учебных объектов. При этом, база знаний пополняется фактами о новом учебном объекте, описанными в терминах онтологий O^{SCORM} и O^{SA} .

Обновление содержания УО влечет за собой и актуализацию метаданных объектов, которая может выполняться как в автоматическом (<Дата создания>, <Авторы>), так и в ручном режиме (<Семантическое описание>). Актуализированные <Учебный блок> и <Метаданные> пополняют репозиторий учебных объектов.

3. Формирование компетентностно-ориентированного УМК на основе интегрированного информационно-образовательного пространства

Представленный способ интеграции научно-образовательного контента в информационно-образовательном пространстве на базе совместного использования онтологий $O^{Competence}$, O^{SCORM} и O^{SA} позволяет обеспечить формирование индивидуальных компетентностно-ориентированных учебно-методических комплексов.

Процесс формирования УМК подразумевает последовательное выполнение ряда этапов (рисунок 1):

- Этап 1. Определение списка целевых компетенций;
- Этап 2. Анализ целевых компетенций;

- Этап 3. Определение условий отбора учебных объектов;
- Этап 4. Поиск учебных объектов и оценка релевантности результатов поиска;
- Этап 5. Выстраивание логической последовательности изучения материалов.

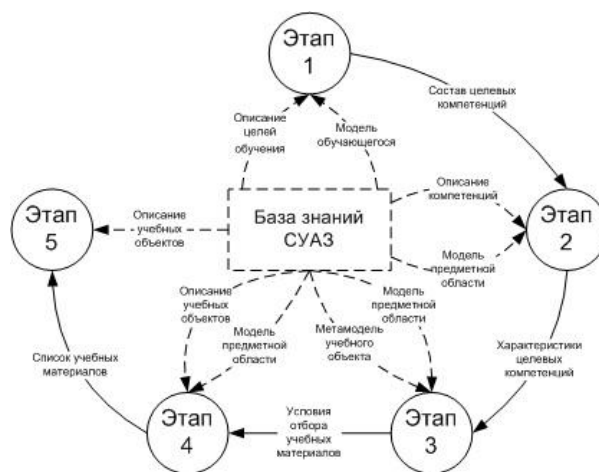


Рисунок 2 – Порядок формирования индивидуальных компетентностно-ориентированных УМК

Рассмотрим далее каждый из этапов подробнее.

1. Определение списка целевых компетенций

Согласно онтологии $O^{Competence}$ и на основе имеющихся результатов обучения формируется модель обучающегося (S). Эта модель включает в себя:

- наименование образовательной программы (EP), по которой проходит обучение студента и которой соответствует некоторый набор компетенций,
- перечень освоенных компетенций (CompS), характеризуемый полученными знаниями, умениями и навыками (KnowledgeS, AbilityS, Skills),
- список дисциплин и модулей, по которым было завершено обучение (DisciplineS, ModuleS).

Согласно онтологии $O^{Competence}$ определяется целевая компетенция (целевой набор компетенций) ?Comp. Для этого выполняется следующий запрос, позволяющий отобрать те компетенции, которые не были освоены обучающимся, но обязательно должны быть освоены в рамках данной образовательной программы:

?- ?Comp:Competence AND
EP:EducationalProgram[forms -> ?Comp] and NOT
S:Student[mastered -> ?Comp]

2. Анализ целевых компетенций

Для каждой целевой компетенции Comp осуществляется поиск ее составных частей (?C) – элементарных компетенций, которые еще не были освоены обучающимся. Для этого выполняется следующий запрос:

?- Comp:Competence[hasPart -> ?C] AND NOT

S:Student[mastered → ?C]

Для формирования полного списка составных компетенций должно применяться правило транзитивности:

?D:Competence[hasPart → ?F] : –
?D:Competence[hasPart → ?E] AND
?E:Competence[hasPart → ?F]

Для каждой целевой компетенции (Comp) определяется набор соответствующих им знаний (?K), умений (?A) и навыков (?S), представляющие собой понятия онтологии предметной области O^{SA} .

?– Comp:Competence [isKnowledgeOf → ?K,
isAbilityTo → ?A, isSkillOf → ?O]

Для каждой целевой компетенции Comp определяется выполнение правила, согласно которому компетенция считается освоенной, в случае если обучающийся получил необходимые знания, умения и навыки:

?N:Student[mastered → ?M:Competence] : –
?M[isKnowledgeOf → ?K:Knowledge, isAbilityTo →
?A:Ability, isSkillOf → ?O:Skill] AND ?N [knows →
?K, haveAbilityTo → ?A, haveSkillOf → ?O]

3. Определение условий отбора учебных объектов

Согласно онтологии O^{SCORM} определяются понятия, описывающие вид (Vid) учебных объектов ?X, которые необходимо найти. При этом, могут быть определены такие виды УО как тесты, практикумы, контрольные работы, пособия и т.п.

Согласно онтологии O^{SCORM} задаются ограничения (Constr) на значения (?Value) атрибутов (hasAttr) искомым учебных объектов. При этом, могут быть определены такие атрибуты УО как цена, длительность изучения, дата создания и т.п.

Согласно онтологии O^{SCORM} определяются возможные связи (?hasRel) искомым учебных объектов, а также связанные понятия (или имена объектов) – ?XRelations. При этом, могут быть определены такие отношения учебных объектов, как авторство, отношение к определенной организации и связи с другими объектами.

Согласно онтологий $O^{Competence}$ и O^{SA} определяются подмножества $I_j \subset I$ и $C_j \subset C$, описывающие семантику (describes) искомого учебного объекта как описание знаний, умений и навыков (Knowledge, Abilities, Skills), формируемых при его освоении.

4. Поиск учебных объектов и оценка релевантности результатов поиска

На основе условий отбора, определенных на предыдущем этапе, формируются запросы на поиск учебных объектов, имеющие следующий формат:

?– Vid(?X) and ?X[hasRel → ?XRelations] – поиск по связям учебного объекта

?– Vid(?X) and ?X[hasAttr → ?Value] and ?Value

{>,<,<= } Constr – поиск по атрибутам

?– Vid(?X) and ?X[describes → {?K, ?A, ?S}] – поиск по семантическому описанию

На основе результата выполнения запросов на поиск учебных объектов, определяется релевантность отобранных УО. При этом наивысшую оценку имеют учебные объекты, полностью отвечающие требованиям пользователя, то есть множество $LO_i^{Result} \in LO^{Result} = LO^A \cap LO^B \cap LO^C$, где LO^A, LO^B, LO^C – множества УО, отвечающие запросам a, b и c, соответственно.

Если отобранные по базовым метаданным учебные объекты не отвечают требованиям, предъявляемым пользователем к содержанию (семантике), то степень релевантности определяется в результате сопоставления отношений describes отобранных УО и параметров запроса, согласно способу оценки семантической близости понятий [Тузовский и др., 2005],[Wu et al., 1994].

5. Выстраивание логической последовательности изучения материалов

Для определения логической последовательности изучения материалов, входящих в состав учебного объекта, используются данные блока <Декларация УО> .

Для определения последовательности освоения компетенции ?C используются сведения о последовательно изучаемых объектах знаний ?KO1 и ?KO2.

?– ?KO1:KnowledgeObject AND
?KO2:KnowledgeObject AND ?KO1[isPredecessorOf –
> ?KO2]

Последовательность изучения учебных объектов зависит от педагогической методики, которая может быть задана напрямую или представлена в виде набора правил:

Прохождению теста должно предшествовать изучение материала.

?LO1[requires → ?LO2] : – ?LO1[Test[hasContext →
?Context, keywords → ?KW] AND
?LO2:Text[hasContext → ?Context, keywords →
?KW]

Изучению дополнительного материала должно предшествовать изучение основного материала.

?LO1[requires → ?LO2] : – ?LO1[isBasedOn→?LO2]

Изучению сложного материала должно предшествовать изучение простого материала.

?LO1[requires → ?LO2] : – ?LO1[hasDifficulty → ?d1,
hasContext → ?Context, keywords → ?KW] AND
?LO2[hasDifficulty → ?d2, hasContext → ?Context,
keywords → ?KW] and ?d2<?d1

ЗАКЛЮЧЕНИЕ

Организация интегрированного информационно-образовательного пространства высшего учебного заведения на основе онтологического подхода и его использования в процессе обучения при формировании профессиональных компетенций позволяет обеспечить, с одной стороны, полноту и всесторонность охвата научно-образовательным контентом потребностей обучающихся, а с другой стороны, повысить гибкость и адекватность формирования образовательных программ в соответствии с конкретными профилями и потребностями обучающихся. При этом повышается актуальность использования учебного материала в различных дисциплинах, интегрируются коллективные усилия преподавателей, научных работников, студентов в процессе создания контента и повышается качество образовательных программ.

Работа выполнена при финансовой поддержке РФФИ. Проект № 10-07-00672а «Разработка методологии создания и использования динамических интеллектуальных систем управления процессами на основе многоагентной технологии»

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Гаврилова и др., 2006] Гаврилова Т.А., Кудрявцев Д.В., Горовой В.А. Модели и методы формирования онтологий // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. 2006. № 46.
- [Домрачев, 1996] Домрачев В.Г., Ретинская И.В. О классификации образовательных информационных технологий // Информационные технологии, 1996, № 2
- [Зиндер и др., 2011] Зиндер Е.З., Тельнов Ю.Ф., Юнатова И.Г. Методика построения модели компетенций на основе профессиональных стандартов в области ИКТ для создания программ дополнительного профессионального образования // Вестник УМО. Экономика, статистика и информатика, 2011, № 6
- [Кудрявцев, 2010] Д.В. Кудрявцев. Системы управления знаниями и применение онтологий – СПб: Издательство СПбПУ, 2010
- [Тельнов, 2009] Тельнов Ю.Ф., Рогозин О.В. Разработка инновационных образовательных технологий на основе модели с использованием SCORM-спецификаций // Научно-практический журнал «Открытое образование», №4, 2009
- [Тельнов и др., 2010] Тельнов Ю.Ф., Данилов А.В., Казаков В.А. Сервисно-ориентированная архитектура динамической интеллектуальной системы управления бизнес-процессами // Научно-практический журнал "Открытое образование", №6, 2010
- [Тельнов и др., 2012а] Тельнов Ю.Ф., Казаков В.А., Данилов А.В., Формализация взаимодействия сервисов и агентов динамической интеллектуальной системы управления бизнес-процессами // Научно-практический журнал "Открытое образование", №1, 2012
- [Тельнов и др., 2012б] Тельнов Ю.Ф., Казаков В.А. Технология организации и доступа к знаниям в интегрированном информационно-образовательном пространстве // Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ 2012 (16 - 20 октября 2012 г., г. Белгород, Россия): Труды конференции. Т.2. - Белгород: Изд-во БГУ, 2012. - с. 193 - 204
- [Трембач, 2012] Трембач В.М., Интеллектуальные технологии для решения задач непрерывного образования, // Научно-практический журнал «Открытое образование», МЭСИ, №3, 2012, с. 4-11
- [Тузовский и др., 2005] Тузовский А.Ф., Чириков С.В., Ямпольский В.З. Системы управления знаниями (методы и технологии) // Под общ. ред. В.З. Ямпольского. – Томск: Изд-во НТЛ, 2005. – 260 с.

- [Brusilivski, 1996] P. Brusilivski «Adaptive and Intelligent Technologies for Web-based Education. In C. Rollinger and C. Peylo (eds.), Special Issue on Intelligent Systems and Teleteaching, 1996
- [Frank et al., 2000] Dieter Fensel Frank, Frank Van Harmelen, Michel Klein, Hans Akkermans. On-To-Knowledge: Ontology-based Tools for Knowledge Management // Proceedings of the eBusiness and eWork 2000 (EMMSEC 2000) Conference, Madrid, Spain, 2000
- [McIlraith et al., 2002] McIlraith, S. & Son, T. Adapting Golog for Composition of Semantic Web Services. Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning (KR2002), April, 2002, 2002.. <http://www.ksl.stanford.edu>
- [Mentras et al., 2002] Mentras G., Apostolou D., Abecker A., Young R. Knowledge Asset Management Beyond the Process-centered and Product-centered Approaches, Shpringer-London, 2002
- [OASIS, 2012] OASIS. Online community for the Universal description, discovery, and integration (UDDI). <http://uddi.xml.org/>
- [Ontoprise, 2010] Ontoprise GmbH. ObjectLogic Tutorial, 2010. http://www.ontoprise.de/fileadmin/user_upload/Publications_EN/ObjectLogic_Tutorial.pdf
- [Schreiber et al., 2000] Schreiber G., Akkermans H, Anjewierden A. etc. Knowledge Engineering and Management: The CommonKADS Methodology, The MIT Press, 2000
- [W3C, 2001] W3C. Web Services Description Language (WSDL) 1.1. <http://www.w3.org/TR/wsdl/>
- [W3C, 2007] W3C. Semantic Annotations for WSDL and XML Schema. <http://www.w3.org/TR/sawSDL/>
- [Wu et al., 1994] Wu Z., Palmer M. Verb semantics and lexical selection // Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, 1994 - P. 133-138.

CREATING OF INFORMATION-EDUCATIONAL SPACE BASED ON SERVICE-ORIENTED ARCHITECTURE AND MULTI-AGENT TECHNOLOGY

Telnov Yu.F., Kazakov V.A., Trembach V.M.

Moscow State University of Economics, Statistics and Informatics, Moscow, Russia

ytnov@mesi.ru

vkazakov@mesi.ru

vtrembach@mesi.ru

The approach for organization of information-educational space of university based on multi-agent and service-oriented architecture is represented. This approach is intended to ensure efficient use of distributed scientific and educational content in the process of mastering and generating new knowledge.

INTRODUCTION

Information and education space is an organizational and technological infrastructure environment of e-learning system. It includes the distributed content in the form of electronic educational resources and a set of IT services that provide remote access to perform functions on information and software (support) for educational activities.

Development of integrated information and educational space is based on the principles of building a knowledge management system (KMS). Usually they use the concept of the object organization of knowledge, multi-agent technology and ontological approach.

However, pragmatic aspect of using the information

space is not implemented in existing methodologies. And they don't reflect the characteristic features of using of intellectual resources (sources of knowledge) for educational and research purposes, in particular, the collective nature of the training and research using multi-agent technology. Also, existing methodologies are not based on modern service-oriented architectures and not sufficiently developed to use heterogeneous knowledge sources and in aspect of dynamic process of mastering and the generation of knowledge.

Creating of information and educational space requires a comprehensive application of object technology to knowledge representation in knowledge management systems based on the ontological and the competence approach and multi-agent technology, using a service-oriented architecture. This provides a distributed storage and sharing of heterogeneous scientific and educational content for research, methodical workings and learning for all participants of the scientific and educational process.

MAIN PART

The main scientific and educational services:

- The formation of the curriculum with the individual learning paths;
- The formation of educational and methodical complex for individual learning paths;
- The formation of workshop including definition a practical problem considering the profile of the student, the case-based project solutions selection;
- Evaluation of the quality of the results (the results of intellectual activity) based on the control tasks selection and implementation to test knowledge and skills;
- The formation of of learning groups by creating, updating and selecting profiles of faculty, staff and students.

The classes of software agents that implement certain functions of participants of the scientific and educational process:

- Teachers (academic staff);
- Students;
- Division managers (heads of chairs, heads of research departments);
- Managers of research and education process (Directorate of educational institutes, faculties);
- Administrators of scientific and educational space (IT departments, knowledge management departments, content management departments, database administrators).

The main elements of the knowledge base (repository) are learning and research objects, which include:

- Description of the knowledge, skills and competencies;
- Description of the practical applications;
- Design solutions;
- Intellectual activity results obtained in the course of solving the previously assigned tasks;

- Test cases (situation);
- Tests;
- Description of the methods and tools;
- Descriptions of regulations;
- Profile descriptions of teachers (researchers);
- Profile descriptions of students.

The methods of ontological engineering are proposed to use to integrate the heterogeneous sources of information and knowledge in the information and educational space. The novelty of our approach is in the conjugation of the learning objects ontology, subject ontology and competencies ontology. Learning objects are loaded into the repository. In this case, the basic mandatory metadata fields that correspond to the attributes of the concept of learning object in the learning objects ontology are specified. Also defined semantic description of learning objects in terms of subject area. After that, the data is updated in the repository of learning objects.

The formation of educational and methodical complex includes a sequential execution of stages. The model of student generated according to the competencies ontology on the basis of learning results. Then, a set of target competencies should be defined. Elementary skills that have not been mastered student for each target competence will be searched. According to the learning objects ontology determine the form of learning objects (tests, workshops, tests, manuals), restrictions on the values of the attributes (cost, duration of study, date of creation), connectivity (authorship, relation to a particular organization and links to other sites) and semantics of required learning object as a description of the knowledge, skills and abilities. Search request are formed on the basis of certain criteria. Then determine the relevance of the selected learning objects. The degree of relevance is determined by estimating the semantic proximity of selected learning objects description and query parameters. The sequence of learning objects depends on the pedagogical techniques, which can be given directly or represented as a set of rules.

CONCLUSION

Organization of integrated information and educational space based on ontological approach and its use in the learning process in the formation of professional competence ensures, on the one hand, complete and comprehensive coverage of the educational needs of the students, on the other hand, flexibility and adequacy of the formation of educational programs. This increases the relevance and quality of educational material, integrates the collective efforts of teachers, researchers and students in the creation of new content.

This work was supported by RFBR. Project № 10-07-00672a «Development of the methodology of creating and using of dynamic intelligent systems in the process management based on multi-agent technology»



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

РЕГУЛЯРНАЯ АКТУАЛИЗАЦИЯ УЧЕБНО-МЕТОДИЧЕСКОГО КОНТЕНТА С ИСПОЛЬЗОВАНИЕМ МЕТАДАННЫХ НА ОСНОВЕ МОДЕЛИРОВАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ

Горбачёв Н.Н.*

** Минский филиал федерального государственного бюджетного образовательного учреждения высшего профессионального образования "Московский государственный университет экономики, статистики и информатики (МЭСИ)"*

г. Минск, Республика Беларусь

ngorbachev@mesi.ru

В работе рассматриваются методы регулярного обновления образовательного контента на основе модели предметной области вуза в виде онтологии. Предложены методы управления перспективными информационными запасами вуза, возникающими в результате индивидуальной и коллективной работы преподавателей и студентов с внешними источниками знаний с использованием современных информационных технологий совместной работы.

Ключевые слова: управление контентом, модель предметной области, учебно-методический контент, электронные образовательные информационные ресурсы.

ВВЕДЕНИЕ

Развитие информационно-коммуникационных технологий (ИКТ) и внедрение инструментальных средств Web 2.0 привело к широкому использованию средств совместной (коллаборативной) работы в процессе создания учебно-методических материалов. Во многих случаях они являются результатом коллективного творчества не только многих преподавателей, но и других участников образовательного процесса (учеников, студентов, слушателей), которые совершенствуют учебно-методический контент (контент - информационно значимое наполнение информационного ресурса, которое может быть предоставлено пользователю [Стратегия, 2015]) в процессе обучения: указывают на возникающие несоответствия другим информационным ресурсам, предлагают альтернативные источники знаний.

Постоянный обмен данными, информацией, знаниями в процессе общения между всеми участниками образовательного процесса привел к росту объемов контента, который потенциально может быть использован в учебном процессе и представляет собой перспективные информационные запасы.

В этой связи, стратегическими задачами при создании учебно-методических комплексов

становятся:

– обеспечение высокого уровня актуальности, достоверности и оперативности доступа к образовательным информационным ресурсам с возможностью выбора альтернативных источников знаний;

– оценки его полноты и ценности для развития актуальных компетенций, а также систематизация новых знаний, генерируемых участниками сетевых учебных сообществ, которые могут трактоваться как перспективные информационные запасы.

1. Повышение мобильности, коллаборативная работа – основные направления развития современных образовательных технологий

В прогнозе развития рынка электронных образовательных услуг до 2014 г., подготовленном аналитическим агентством Ambientinsight [Ambientinsight, 2008] выделяется 4 поколения учебно-методического контента:

- 2002 – 2004 гг.: контент для самостоятельной подготовки, используемый в электронном обучении (в 2010 г. наблюдался рост в академическом сегменте при устойчивом снижении в других сегментах).

- 2005 – 2007 гг.: контент для самостоятельной подготовки с использованием технологий совместной (коллаборативной) работы, виртуального тьюторства, виртуальных аудиторий и лабораторий.
- 2008 – 2011 гг.: контент для мобильного обучения, доставляемый с использованием высокоскоростных сетей 3G на мобильные устройства пользователей, работающий с на основе интерактивных мобильных программных приложений, основанный на решении реальных учебных проблемных ситуаций.
- 2012 – 2014 гг.: контент для мобильного обучения с использованием технологий и инструментальных средств совместной работы, доставляемый с использованием сетей 4G на многофункциональные мобильные устройства пользователей; использование возможностей среды облачных вычислений и кроссплатформенных решений; разработка учебно-методического

контента происходит с участием конечных пользователей (студентов, слушателей); обучение производится на основе высокоинтерактивных обучающих мобильных приложений, моделирующих в режиме реального времени учебные проблемные ситуации.

На этом фоне, важным является развитие средств совместной работы, характеризующее концептуальное движение от автоматизации процессов обучения к внедрению инновационных методов передачи знаний (рисунок 1). На рынке электронных образовательных услуг выделяется отдельный сегмент – электронное обучение с использованием технологий совместной работы и социальных сетей с прогнозом роста на период до 2013 года до 28% от всего объема рынка электронных образовательных услуг [Strategy, 2008].

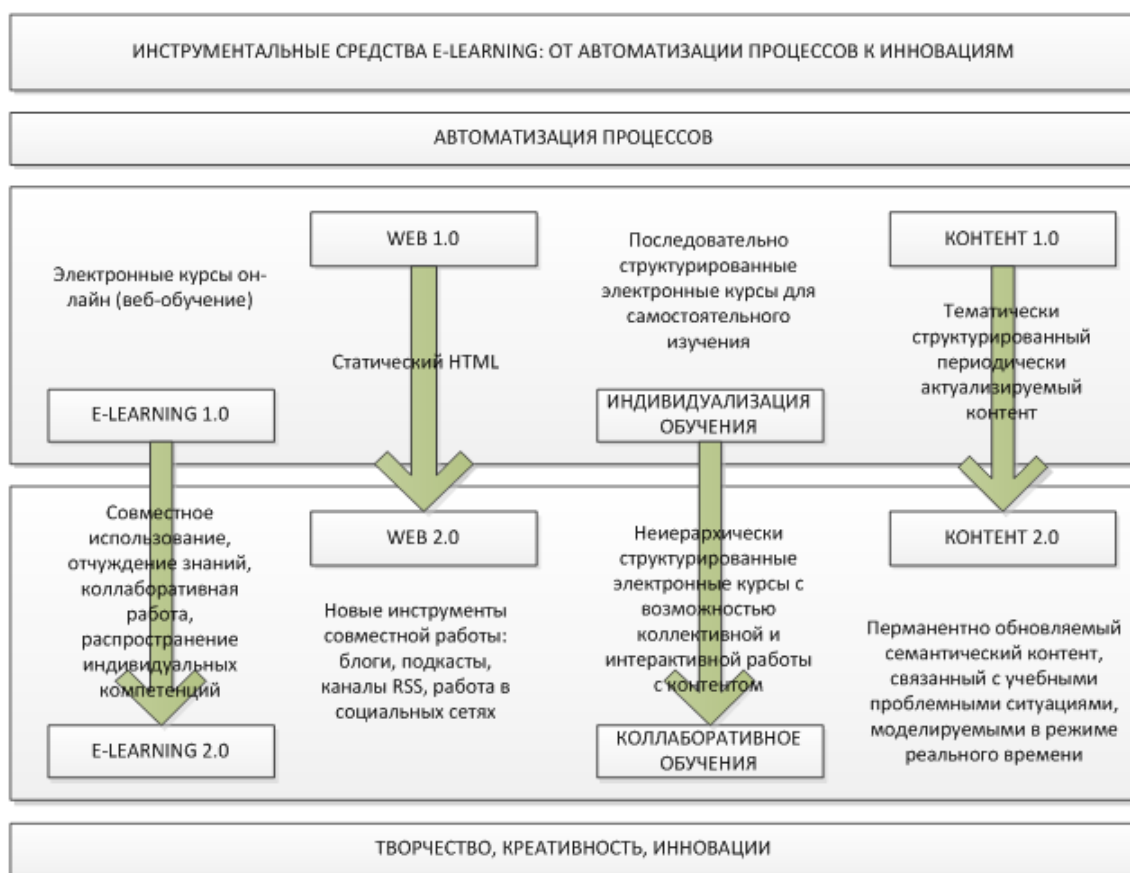


Рисунок 1 - Развитие инструментальных средств коллаборативной работы для электронного обучения

2. Место онтологии предметной области в структуре информационных ресурсов

При эффективном использовании таких технологий становится возможной формализация части знаний, которые ранее считались их с использованием модели предметной области ВУЗа, анализ и последующее применение их для обновления и развития имеющихся

неотчуждаемыми, неотделимыми от индивида и вовлечение их в образовательный процесс. Первоначальный учебно-методический контент совершенствуется участниками сетевых сообществ и становится результатом групповой работы. Одним из перспективных решений, позволяющих управлять знаниями, является классифицирование электронных образовательных информационных ресурсов (э-ОИР) (рисунок 2).

Онтология предметной области является основой инструментария управления учебно-методическим контентом, их актуализацией и развитием с использованием семантического веб-портала, позволяющего [Гринберг, 2009]:

- осуществлять эффективную актуализацию нескольких э-ОИР с использованием нового (дополнительного) информационного ресурса (ИР), обеспечивая экономию трудозатрат и сокращение времени актуализации;

- развивать официальные и альтернативные э-ОИР в рамках предметной области ВУЗа, увеличивая стоимость НМА;
- учитывать требования на актуализацию существующих э-ОИР с точки зрения соответствия официальным тезаурусам по дисциплинам, обеспечивая улучшение качественных показателей э-ОИР.

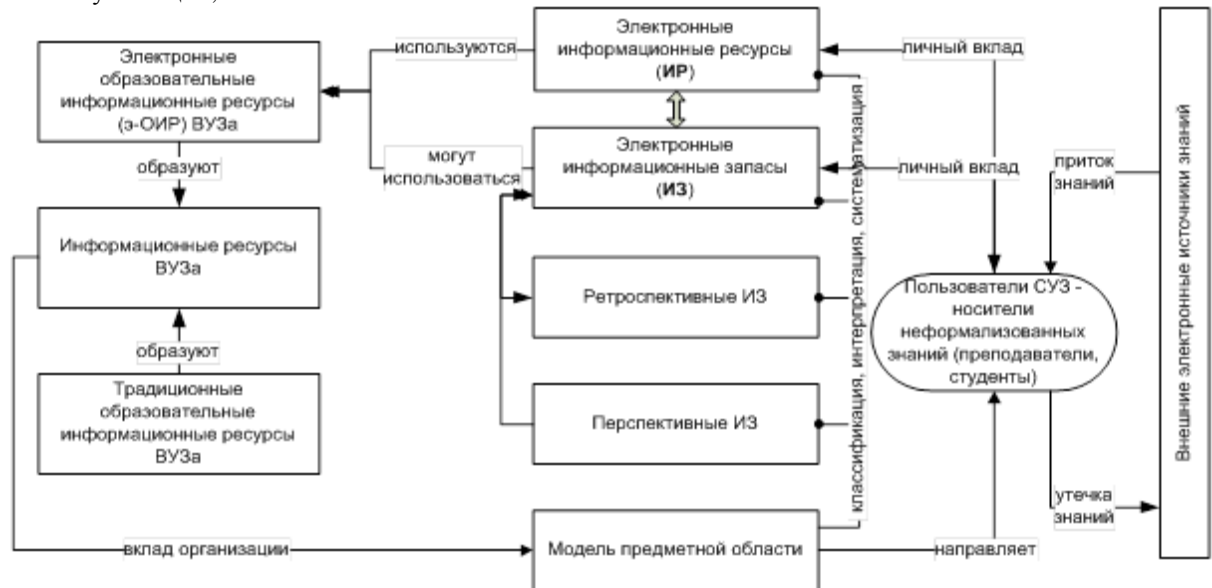


Рисунок 2 - Схема взаимодействия элементов информационной среды ВУЗа

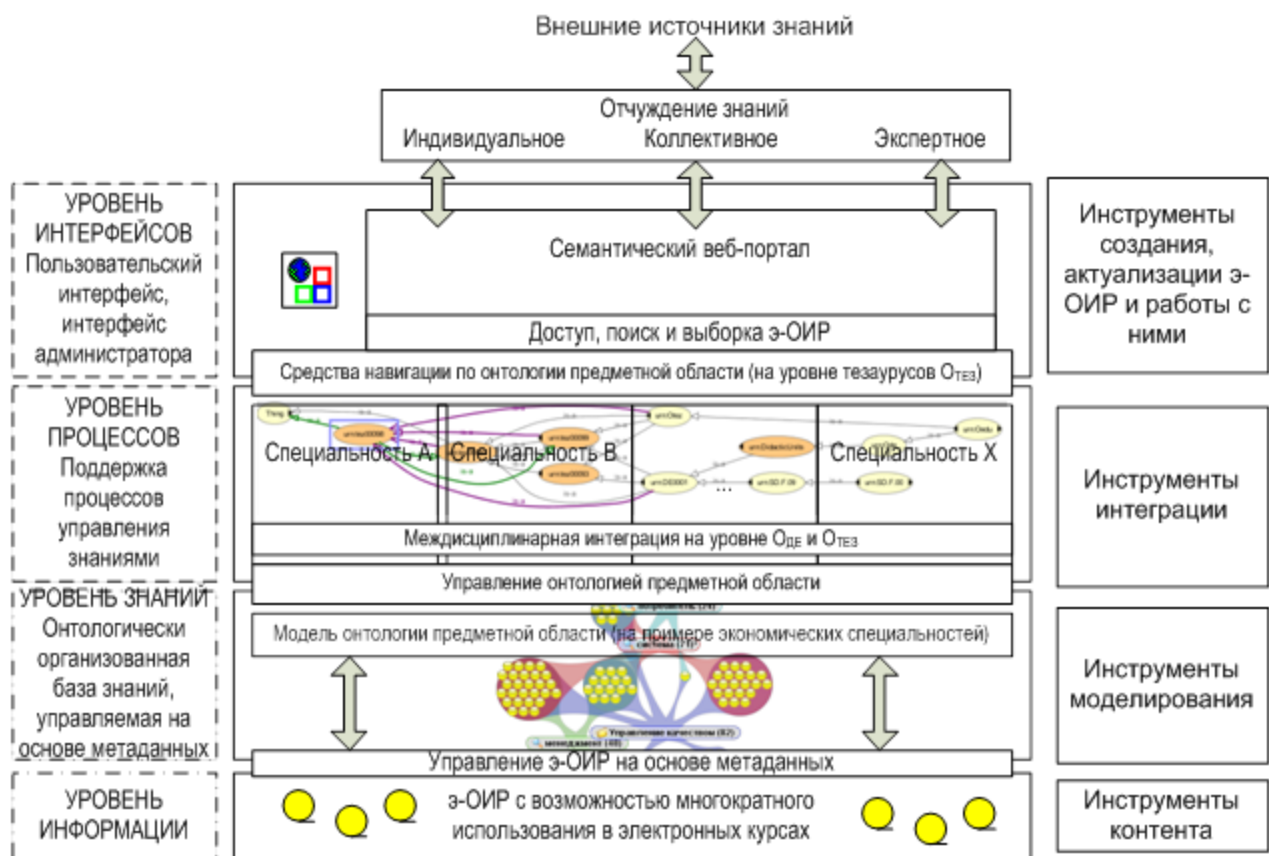


Рисунок 3 - Архитектура СУЗ, функционирующей с использованием онтологии предметной области

Таким образом, модель предметной области ВУЗа имеет структурное значение, являясь основой для управления э-ОИР на основе метаданных, обеспечивая междисциплинарную интеграцию на уровне дидактических единиц и тезаурусов (создание перекрестных ссылок, управление э-ОИР, имеющим отношение к нескольким дисциплинам), создавая основу для работы пользователей со внешними источниками при развитии и актуализации существующих э-ОИР, управления дополнительными ИР, их аттестацией как образовательных.

3. Инструментальный комплекс управления динамической публикацией обновлений контента

В качестве объектов управления в СУЗ выступают элементы электронных УМК, представленные в виде э-ОИР с возможностью многократного использования в электронных курсах, представляющие собой уровень информации СУЗ. Инструментальные средства оперирования э-ОИР на этом уровне представлены инструментами контента, семантические возможности управления ими реализуются в переходном слое инструментария после разметки метаданных с использованием онтологии предметной области ВУЗа. На уровне знаний управление э-ОИР осуществляется с использованием онтологически организованной базы данных, управляемой на основе метаданных. Инструментарий управления СУЗ представлен в виде инструментов моделирования предметной области, индексирования метаданных, поиска и рационального отбора э-ОИР для использования при актуализации и развитии элементов электронных УМК на основе э-ОИР многократного применения. На промежуточном уровне поддержки процессов управления знаниями в структуре СУЗ осуществляется управление моделью предметной области, ее актуализацией и развитием для наиболее эффективного управления э-ОИР с использованием онтологии. На уровне поддержки процессов управления знаниями используются инструменты интеграции знаний, позволяющие обеспечить междисциплинарную интеграцию при использовании э-ОИР (рисунок 3).

4. Реализация предложенных методов управления контентом на основе моделирования предметной области

В качестве примера можно привести фрагмент модели предметной области по экономическим специальностям, разрабатываемой в рамках внутреннего НИР Минского филиала МЭСИ. Использование онтологии как метода формализованного представления предметной области позволяет создать единую технологическую платформу, позволяющую осуществлять управление актуализацией и развитием элементов электронных

УМК, проводить индексирование и поиск э-ОИР для создания электронных курсов и проверки их в части соответствия требованиям ГОС на уровне тезаурусов.

Планомерное развитие информационной составляющей СУЗ ВУЗа на основе онтологии предметной области обеспечивает развитие э-ОИР и увеличивает возможности рационального выбора содержания учебных курсов по дисциплинам.

Графовая модель связей тезауруса, визуализированная с помощью инструментария Protégé OntoViz, дает представление о связи между элементами тезауруса. Проверка на непротиворечивость позволяет сделать вывод о правомерности использования логических связей между элементами O_{TEZ} (на схеме используются латинская транскрипция обозначений O_{TEZ}). Свойство расширяемости модели реализовано в возможности корректировки O_{TEZ} путем введения новых понятий и отношений.

Рассматриваемую онтологию предметной области ВУЗа можно представить как:

$O_{EDU} = \{Def, Atr, Rel, Func\}$, где: $Def = \{O_{оп}, O_{уп}, O_{де}, O_{тез}\}$, при этом $Def = \{def_1, \dots, def_i\}$ – конечное множество понятий онтологии, где: $O_{оп}$ – онтология образовательных программ ВУЗа; $O_{уп}$ – онтология дисциплин учебных планов, включенных в ОП; $O_{де}$ – онтология, созданная на основе дидактических единиц, описывающих компетенции в соответствии с ФГОС; $O_{тез}$ – онтология, созданная на основе тезаурусов для конкретизации дидактических единиц ГОС; $Atr = \{atr_1, \dots, atr_i\}$ – конечное множество атрибутов понятий Def , необходимых для построения онтологии; $Rel = \{rel_1, \dots, rel_i\}$ – множество отношений между понятиями, определяющие их взаимосвязь (например, в структурно-логических схемах – последовательностях преподавания дисциплин (СЛС), между объектами репозитория учебно-методического контента (RLO); $Func = \{func_1, \dots, func_i\}$ – функции, позволяющие определить зависимости между понятиями.

O_{EDU} не включает все административные процессы обучающего, и ограничивается только описанием процессов, связанных с подготовкой и описанием образовательных объектов. Для программ дополнительного обучения, консалтинга формируются расширения онтологии $OEDU$ для описания понятий, которые не включены в $O_{де}$. Кроме этого, эксперты по дисциплинам (преподаватели) могут расширять онтологию $O_{ГОС}$ для введения новых понятий, ее детализации, актуализации с учетом достижений современной науки за счет $O_{тез}$, т.к. $O_{де}$ является более статичной. Онтологии различных ВУЗов идентичны на уровне $O_{де}$. В свою очередь, $O_{тез}$ позволяет сделать структуру онтологии более динамичной, с учетом конкурирующих концепций преподавания различных дисциплин.

Модель, представленная на рисунок 4, иллюстрирует, что понятие urn:tez00098 («качественные показатели») непосредственно связано с кратной дугой с понятием urn:tez00097 («виды показателей»), и некротной дугой с urn:tez00099 («показатели»), опосредовано связано с понятием urn:tez00093 («измерения») и относится к дидактической единице urn:DE0001 («квалиметрия»). urn:DE0001 urn:DidacticUnits urn:DE0001 urn:SD.F.09 (где urn:DidacticUnits – множество дидактических единиц, urn:SD.F.09 – дисциплина «управление качеством»). Графовая модель не дает полного представления об отношениях между понятиями. Для более полного визуального представления связей между элементами тезаурусов воспользуемся средством Altova SemanticWorks (рисунок 5).

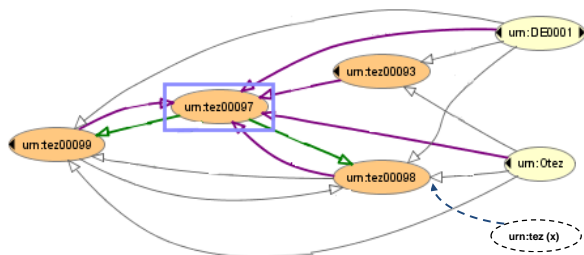
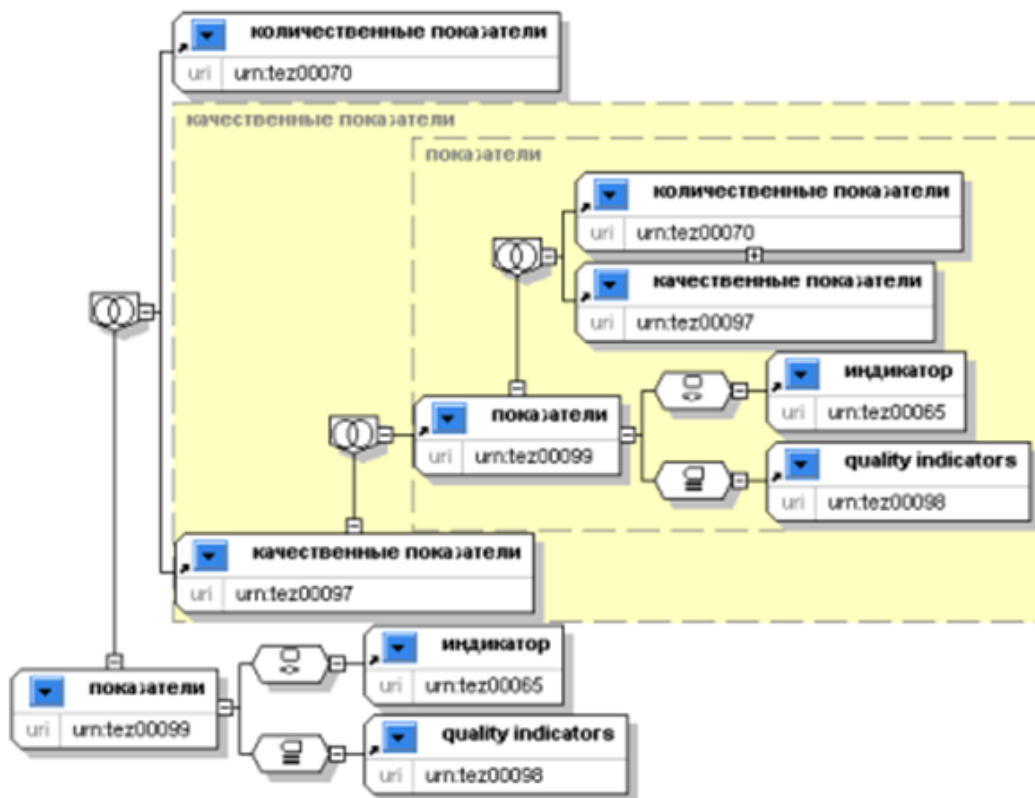


Рисунок 4 - Фрагмент графового представления модели онтологии на уровне тезаурусов (связи понятия tez00097 «качественные показатели»)

Таблица 1 – Связи понятия tez00097 «качественные показатели»

Идентификатор термина	Наименование термина	Источник
urn:Otez	Модель тезауруса	
urn:DE0001	Квалиметрия как наука	Федеральный государственный образовательный стандарт (ФГОС)
urn:tez00093	Измерения	Тезаурус по дисциплине «Управление качеством»
urn:tez00097	Качественные показатели	
urn:tez00098	Quality indicators	
urn:tez00099	Показатели	
urn:tez (x)	Новый элемент тезауруса	Активность пользователей СУЗ (преподавателей, студентов)



Generated by SemanticWorks

www.altova.com

Рисунок 5 - Фрагмент функционального представления онтологии (фрагмент связей элемента тезауруса tez00099 «показатели»), используемое в качестве инструментария навигации по онтологии

Функциональное представление онтологии является более содержательной, что дало возможность визуально представить отношения между понятиями модели ОТЕЗ, в которой присутствуют сложные связи между элементами (по сравнению с Оуп, Оде, Ооп).

Пример результата поискового запроса с использованием индексированных метаданных по дисциплине «Управление качеством» приведен на рис. 3.2.1. Для организации семантического поиска в данном примере с использованием метаданных используется программный продукт Aduna Autofocus Server (рисунок 6).

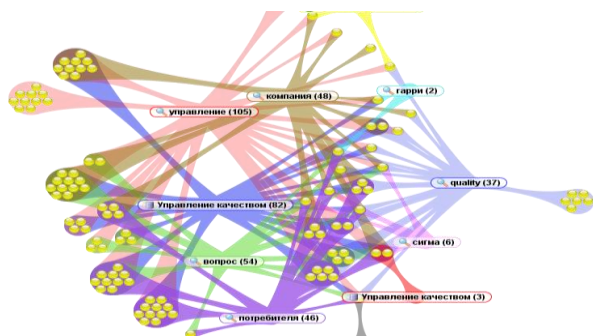


Рисунок 6 - Результат поискового запроса по дисциплине «Управление качеством» по э-ОИР, размещенным в репозитории контента с использованием метаданных

ЗАКЛЮЧЕНИЕ

Построение онтологии на основе математической модели позволило:

- создать описание предметной области экономических специальностей, установления связей между понятиями предметной области, описания атрибутов и отношений между ними;
- создать масштабируемую модель предметной области на основе требований ГОС для управления э-ОИР с учетом требований масштабируемости, интероперабельности, интернализации. Разработанная модель имеет возможность расширения до уровня реальной онтологии и конкретизации отдельных понятий. При необходимости взаимодействия с другими онтологиями в модели реализована возможность определения отношений с соответствующими классами и понятиями. Дополнительные свойства элементов онтологии описываются путем введения новых свойств классов: атрибутов, отношений и функций;
- организовать основу с использованием концепции отчуждения знаний для функционирования системы актуализации ОИР в рамках расширяемой системы тезаурусов по учебным дисциплинам. Устойчивость и качественные показатели системы обеспечиваются ограничениями на количество компетенций у одного эксперта и рациональным числом экспертов по каждому классу тезаурусов;
- создать на основе онтологии инструментальные средства управления э-ОИР,

позволяющие осуществлять мониторинг несоответствий, систематизацию и классификацию ИР и ИЗ ВУЗа, публикацию актуализированных э-ОИР на семантическом веб-портале

- разработать рекомендации по описанию метаданных для документов, используемых в системах дистанционного обучения и электронных библиотеках по специальностям экономической направленности;
- предложенная модель может быть применена для описания других предметных областей с учетом адаптации под требования конкретных организаций.

Дальнейшие исследования в области разработки инструментальных средств управления актуализацией и развитием учебно-методическим контентом направлены на расширение области использования модели предметной области ВУЗа: оценка полноты элементов электронных УМК (баз тестовых заданий, учебных проблемных ситуаций, текстов электронных учебников)

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Стратегия, 2015] Стратегия развития информационного общества в Республике Беларусь на период до 2015 года [Электронный ресурс] // URL: <http://pravo.by/webnpa/text.asp?RN=C21001174> (дата обращения: 21.10.2012)

[Ambientinsight, 2008] The US Market for Learning Technology Products and Services: 2008-2013 Forecast and Analysis, Ambient Insight, LLC [Электронный ресурс] // URL: www.ambientinsight.com/Resources/Documents/AmbientInsight_US_2008-2013_LearningTechnologyMarket_ExecutiveOverview.pdf (дата обращения: 21.10.2012)

[Strategy, 2008] E-learning: A Global Strategic Business Report, 2008 [Электронный ресурс] // URL: http://www.strategyr.com/eLEARNING_Market_Report.asp (дата обращения: 21.10.2012)

[Гринберг, 2009] Горбачёв Н.Н., Гринберг А.С. Инструментальный комплекс управления динамической публикацией образовательных информационных ресурсов // Открытое образование, №3, 2009. С. 34-43.

PERMANENT UPDATING OF EDUCATIONAL CONTENT USING KNOWLEDGE DOMAIN BASED METADATA

Gorbachev N.N.*

* *Moscow State University of Economics, Statistics and Informatics (MESI), Minsk, Republic of Belarus*

ngorbachev@mesi.ru

This paper describes methods of permanent updating of educational content using model of knowledge domain of university in a form of ontology. Methods of management of prospective information resources resulting from personal and collaborative works with external knowledge sources of lecturers and students using contemporary information technologies are introduced.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.432:4

СЕМАНТИЧЕСКАЯ МОДЕЛЬ ЭЛЕКТРОННОГО УНИВЕРСИТЕТА

Шарипбаев А.А. *, Омарбекова А.С. *, Барлыбаев А.Б. *

** Евразийский национальный университет имени Л.Н. Гумилева,
г. Астана, Республика Казахстан*

sharalt@mail.ru

omarbekova@mail.ru

frank-ab@mail.ru

В статье представлен подход к построению интеллектуального электронного университета. В качестве информационной модели портала используются онтологии.

Ключевые слова: интеллектуальный электронный университет, представление знаний, онтология.

ВВЕДЕНИЕ

Для высшего учебного заведения знания являются одним из важнейших ресурсов. Вуз приобретает несомненные конкурентные преимущества, если быстрее других создает и находит новые знания, обеспечивает их развитие и практическое внедрение, формируя научные и профессиональные компетенции, как сотрудников, так и выпускников вузов.

Знания, интеллектуальный капитал, интеллектуальная собственность получают растущее признание в качестве нового источника развития [Тузовский, 2007]. В связи с этим инновационные вузы стремятся эффективно распоряжаться, управлять имеющимся у них знаниями. Задача данной статьи является построение семантической модели интеллектуального электронного университета в виде онтологии.

1. Онтология интеллектуального электронного университета

1.1. Онтология и интеллектуальный электронный университет

Онтология играют решающую роль в модели описания знания, без которой, как утверждают специалисты, вход в любую предметную область запрещен.

Проектирование онтологии – это творческий процесс, и поэтому потенциальные приложения семантической сети, а также понимание

разработчиком предметной области и его точка зрения на нее будут, несомненно, влиять на принятие решений [Noy N et al., 2001].

Одним из современных направлений развития е-университета является преобразование процесса электронного обучения в пространство знаний и компетенций. В связи с этим, задача разработки онтологии интеллектуального электронного университета является актуальной как с научной точки зрения, так и практически значимой, поскольку позволяет применить возможности семантической технологии для создания web-портала системы управления знаниями современного вуза. [Балова и др., 2010]

Интеллектуальный электронный университет (далее ИЭУ) - это программный комплекс для автоматизации проведения обучения и контроля знаний по кредитной системе через глобальную сеть на основе методов искусственного интеллекта.

Информационную основу ИЭУ составляют – онтологии, концептуальная модель, с помощью которой осуществляется формализация некоторой области знаний. Вводя формальные описания понятий предметной области в виде классов объектов и отношений между ними, онтология задает структуры для предоставления реальных данных и связей между ними. Использование онтологий для построения информационной основы ИЭУ позволяет не только целостно представить такие трудно формализуемые предметные области, как технические, но и автоматизировать процесс сбора и накопления информации по выбранной тематике. Такая концептуальная модель позволяет обеспечить единообразное представление знаний

данных по выбранной тематике, их семантическую связанность.

1.2. Онтология интеллектуального электронного университета

В связи с переходом Республики Казахстан на дистанционную и модульную технологии обучения задача эффективной организации информационных ресурсов, сопровождающих и обеспечивающих образовательный процесс, ориентированный на получение компетенций, становится первостепенной. В рамках кредитной системы обучения каждый обучающийся формирует свой индивидуальный план, который вуз обязан обеспечить электронными образовательными ресурсами (ЭОР) (курсы лекций, методические пособия, методические указания к лабораторным и практическим работам, указания по выполнению самостоятельной работы и т.д.). Обучающемуся предоставляется возможность самостоятельно осуществлять выбор того или иного ЭОР в соответствии с поставленными им самим образовательными целями. Сложность самостоятельного выбора учебно-методических материалов обусловлена тем, что существующие ЭОР имеют разный уровень качества, слабо структурированы, плохо систематизированы и распределены на различных образовательных порталах вузов и в Интернет.

Становится очевидным, что качество обучения в соответствии с принципами компетентного подхода определяется качеством образовательных ресурсов, которое может оцениваться различными показателями, но в первую очередь, образовательные ресурсы должны соответствовать государственному общеобязательному стандарту образования Республики Казахстан.

Использование онтологических моделей и семантических методов при построении систем управления знаниями вузов не достаточно распространено, а образовательные порталы вузов чаще выступают как информационные, нежели чем семантические с функциями управления знаниями и оценки качества электронных образовательных ресурсов.

Начать разработку онтологии можно с определения ее области и масштаба.

На данном этапе работы можно ответить на вопросы следующим образом:

- так как «Электронный университет» является очень масштабной темой, то семантическая часть данной работы будет охватывать область, связанную с процессом электронного обучения;

- онтология будет использоваться для обеспечения доступа к информационным ресурсам вуза и взаимодействия распределенных учебных сред в дистанционной и модульной технологий обучения;

- данную онтологию будут использовать тьюторы вуза, а поддержкой будет заниматься эксперт онтологической базы знаний.

В процессе электронного обучения участвуют объекты и субъекты. Субъектами являются: обучаемый, обучающие, организаторы обучения и образовательный портал (бизнес-работник). Главным объектом e-learning является процесс обучения, знания которого определяет syllabus дисциплины, а обеспечивает электронное учебное издание. В тоже время дисциплина реализуется через электронное учебное издание. Дисциплина содержит: шифр, название, тип (базовая, профильная, общеобразовательная, дополнительные виды обучения). Дисциплина содержится в учебном плане. Для разработки и дисциплины, и учебного плана применяют Государственный общеобязательный стандарт образования (ГОСО). Учебный план – это есть образовательная программа специальности. Все специальности описаны в классификаторе специальностей, который применяют для организации обучения вузы и для разработки ГОСО и других образовательных нормативных документов.

Специальности высшего и послевузовского образования в Классификаторе объединены по группам и представлены восьмизначными цифровыми кодами.

Первый и второй знак кода, имеющий цифровое и буквенное выражение означает уровень высшего и послевузовского образования (бакалавриат, магистратура, докторантура).

Третий и четвертый знак кода обозначает группу специальностей высшего и послевузовского образования, указывающий область знания.

Пятый, шестой, седьмой, восьмой знаки кода определяют специальности высшего и послевузовского образования и их образовательные программы.

На рисунке 1 представлена онтология системы электронного обучения в университете, которая была написана на унифицированном языке моделирования UML. В качестве инструмента был использован IBM Rational Rose Version 7.0.0.

2. Пользователи и функции интеллектуального электронного портала

Основными функциональными возможностями информационной системы является организация учебного процесса, ведение базы данных студентов и преподавателей, отслеживание текущей успеваемости студентов и графиков выполнения ими учебных планов, прогнозирование результатов очередной экзаменационной сессии и формирование списков "групп риска" студентов, могущих потенциально получить неудовлетворительные оценки в сессию.

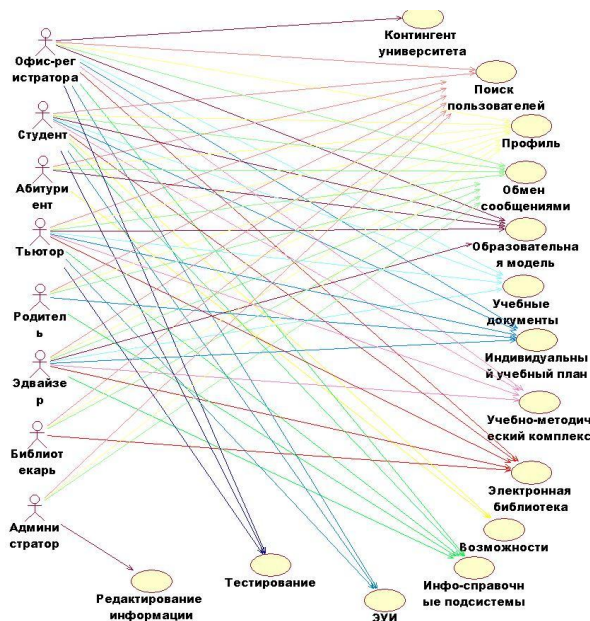


Рисунок 2 – Пользователи и их функции

При создании приложения были использованы фирменные технологии InterSystems, такие как средства построения многомерных баз данных на основе постреляционной технологии Cache, язык программирования приложений Cache Object Script, технология формирования динамических WEB-страниц Cache Server Pages и ZEN [Кирстен В., 2001]. Ссылка на портал 188.127.228.224.

ЗАКЛЮЧЕНИЕ

Рассмотрено решение задач по разработке интеллектуального электронного университета, этап построения онтологии системы e-университета.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Тузовский, 2007] Тузовский А.Ф. Онтолого-семантические модели в корпоративных системах управления знаниями // Томск, 2007. – с. 3-4.
- [Noy N et al., 2001] Noy N., McGuinness D.L. Ontology Development 101: A Guide to Creating Your First Ontology // Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001. URL: http://protege.stanford.edu/publications/ontology_development/ontology101.html. – P. 41-42.
- [Балова и др., 2007] Балова Т.Г., Рохас Криулько. Этапы разработки онтологии электронного университета. Современные

информационные технологии/Компьютерная инженерия/2010/ URL:http://www.rusnauka.com/18_DNI_2010/Informatica/69459.doc.htm

[Кирстен В., 2001] В. Кирстен, М. Ирингер, Б. Рериг, П. Шульте. СУБД CACHE: объектно-ориентированная разработка приложений. Учебный курс /- СПб.: Питер, 2001. - 384 с.

SEMANTIC MODEL OF AN INTELLECTUAL E-UNIVERSITY

Sharipbayev A.A. *, Omarbekova A.S. *, Barlybayev A.B. *

* L.N. Gumilyov Eurasian University, Astana, Republic of Kazakhstan

sharalt@mail.ru
omarbekova@mail.ru
frank-ab@mail.ru

The paper presents the approach to the construction of intellectual electronic university. As an information model of portal used ontology.

INTRODUCTION

Knowledge is one of the most important resources for the higher education institutions. Knowledge, intellectual capital and intellectual property are growing recognition as a new source of development. In connection with this innovative universities seek to effectively manage, control of their knowledge. The objective of this paper is to construct a semantic model of intellectual electronic university in the form of ontology.

MAIN PART

Intellectual electronic university (the IEU) - a software package to automate the training and control of knowledge in credit system through a global network based on artificial intelligence methods.

Information base of IEU are – ontology, conceptual model, which helps in the formalization of a field of knowledge..

CONCLUSION

Problems of the development of intellectual electronic university are was being opened, stage of construct ontology of the e-university.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.432.4

ГЕНЕРАЦИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ЭЛЕКТРОННЫХ УЧЕБНЫХ ИЗДАНИЙ НА ОСНОВЕ ОНТОЛОГИЧЕСКОЙ МОДЕЛИ

Шарипбаев А.А. *, Омарбекова А.С. *, Кинтонова А.Ж. *, Ниязова Р.С. *, Барлыбаев А. *

** Евразийский национальный университет им. Л.Н.Гумилева,
г. Астана, Республика Казахстан*

sharalt@mail.ru

Omarbekova_as@enu.kz

aliya_kint@mail.ru

Rozamgul@List.ru

Frank_b@mail.ru

В статье описана методология создания генератора интеллектуальных электронных учебных изданий (ИЭУИ) для высшего образования по кредитной технологии обучения. Целью является разработка технологии, позволяющей автоматизировать этапы создания ИЭУИ.

Ключевые слова: электронное обучение, генератор, интеллектуальные электронные учебные издания

ВВЕДЕНИЕ

Информатизация образования предполагает наличие ИЭУИ по всем изучаемым дисциплинам. При этом разработчики ИЭУИ используют различные технологии, создают разные интерфейсы для пользователей, предлагают разные требования к составу, структуре, эргономике и другим элементам. Это порождает все новые несовершенные ИЭУИ, которые не всегда пригодны к использованию в учебном процессе.

1. Генератор интеллектуального электронного учебного издания

1.1. Генератор и интеллектуальное электронное учебное издание

ИЭУИ, созданные прямым методом программирования, отличаются качеством разработки, разнообразием стилей реализации (цветовая палитра, интерфейс, структура ИЭУИ, способы подачи материала). Но для таких ИЭУИ характерны также:

– сложность модификации и сопровождения ИЭУИ, так как такие продукты, как правило, являются закрытыми, подвластными только программисту-создателю,

– большой трудоемкостью и длительностью создания ИЭУИ.

Следует упомянуть еще одну проблему прямого метода - эффект второго автора: в создании ИЭУИ принимает участие программист и методист (первые авторы), а использует ИЭУИ в обучении другой преподаватель (второй автор), который часто имеет методику изложения материала, отличную от заложенной в конкретном ИЭУИ. А закрытость, характерная для таких ИЭУИ, не позволяет преподавателю изменить данное ИЭУИ и адаптировать для своих нужд [Шарипбаев, 2005].

Самым отрицательным моментом данного метода является то, что непрограммирующий методист не может применить его самостоятельно, без участия высококвалифицированного программиста.

Опыт создания ИЭУИ показывает, что необходим переход к более производительной технологии создания ИЭУИ различных типов. Причем, преимущественно, ИЭУИ должны создаваться с помощью какой-либо инструментальной системы. Применение инструментальной системы создания ИЭУИ позволяет сократить трудоемкость и сроки разработки, а также дает возможность непосредственной работы с компьютером для преподавателей, которые не являются высококвалифицированными программистами [Omarbekova, 2012 a].

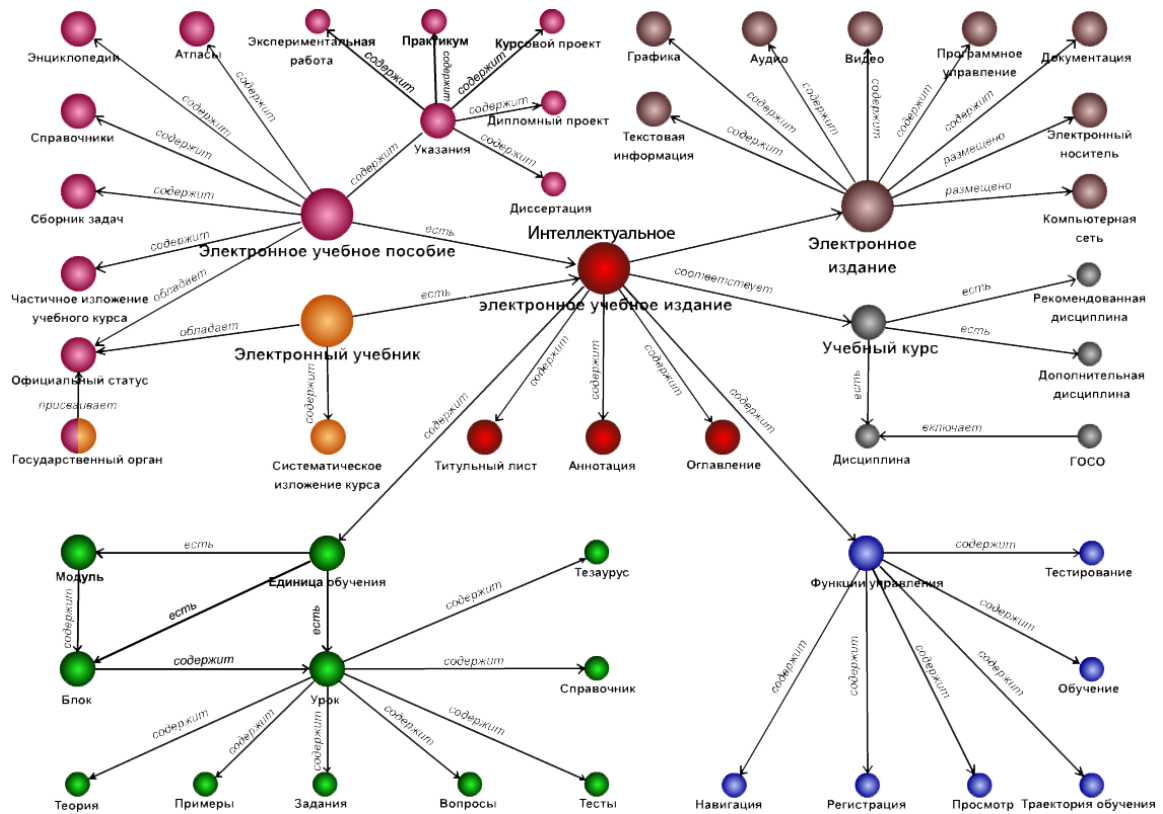


Рисунок 1 – Семантическая модель интеллектуального электронного учебного издания

Согласно стандарту интеллектуальное электронное учебное издание (ИЭУИ) описывается семантической моделью, показанной на рисунке 1 [Шарипбаев, 2012].

Примерная онтологическая модель генератора интеллектуальных электронных учебных изданий с основными понятиями и дефинициями показана на рисунке 2.



Рисунок 2 – Онтологическая модель «Генератора интеллектуальных учебных изданий». Основные понятия и дефиниции

В последнее время использование онтологий для моделирования предметных областей автоматизированных информационных систем получает все более широкое распространение. Наиболее часто такой подход применяется для интеллектуальных систем, в частности, предназначенных для функционирования в сети Интернет. Интернет все более и более становится

образовательной площадкой для большинства населения не только Казахстана, но и всего мира. Это связано с тем, что онтологическая модель позволяет разработать модель метаданных, что значительно улучшает использование системы широким кругом пользователей с точки зрения организации взаимодействия. Более развернутая модель генератора ИЭУИ показана на рисунке 3.

Онтология – это структура, описывающая значения элементов некоторой системы, попытка структурировать окружающий мир, описать какую-то конкретную предметную область в виде понятий и правил, утверждений об этих понятиях, с помощью которых можно формировать отношения, классы, функции и пр. Онтологии предметных областей ограничиваются описанием мира в рамках конкретной предметной области.

Естественно каждая модель может устаревать и выходить из эксплуатации, как и любое программное обеспечение, таким образом, имея жизненный цикл. Модель жизненного цикла – структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении жизненного цикла [Gruber, 1991]. Модель жизненного цикла зависит от специфики, масштаба и сложности проекта и специфики условий, в которых система создается и функционирует. Авторами статьи также рассмотрен данный вопрос (Рисунок 4).



Рисунок 3 – Онтологическая модель

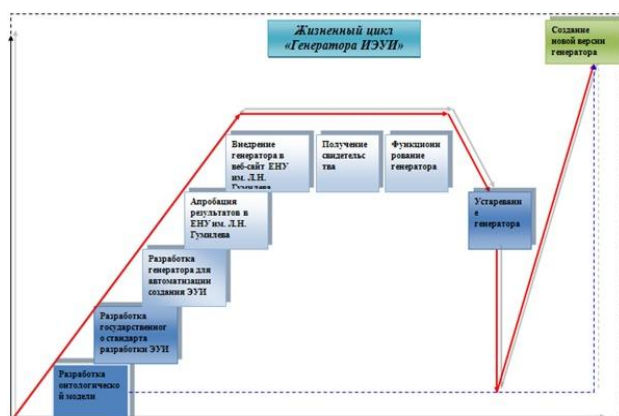


Рисунок 4 - Жизненный цикл генератора ИЭУИ

Основная цель создания генератора ИЭУИ это создание единой стандартной и систематизированной модели электронного учебного издания с интеллектуальными и адаптивными функциями, которая сможет функционировать не только в пределах образовательной территории единого информационного пространства Казнет, но и за его пределами, включаясь в общую глобальную среду (рисунок 5).

Задача построения онтологической модели предметной области генератора ИЭУИ для поддержки коммерциализации результатов инновационных разработок в научных исследованиях является актуальной и сложной научно-практической задачей. Сложность поставленной задачи определяется, в частности, наличием множества межпредметных и междисциплинарных связей и различными целями конечных пользователей системы: ученых, экспертов, пользователей [Omarbekova, 2012b].

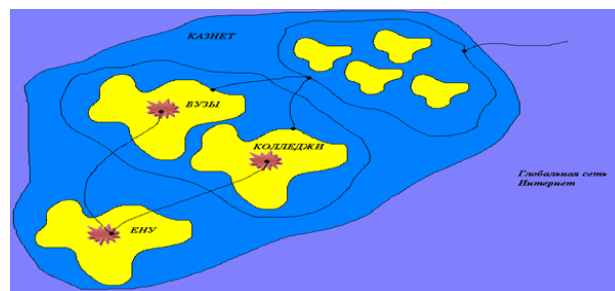


Рисунок 5 - Расширение образовательных границ генератора ИЭУИ

2.Процесс построения ИЭУИ

В научно-исследовательском институте «Искусственный интеллект» Евразийского национального университета им. Л.Н.Гумилева авторами статьи проводятся исследования по созданию интеллектуальных электронных учебных изданий. Процесс создания интеллектуальных электронных учебных изданий продвинулся в своей жизнедеятельности пройдя следующие этапы на сегодняшний день:

- разработан государственный стандарт по созданию и внедрению в образовательный процесс интеллектуальных электронных учебных изданий;
- разработана технология автоматизации создания локальных ИЭУИ;
- разработана технология автоматизации создания интернет-ИЭУИ;
- с помощью генератора ИЭУИ преподавателями разработано по различным дисциплинам множество интеллектуальных электронных учебных изданий, которые применяются в учебном процессе, о чем свидетельствуют акты внедрения. Некоторые ИЭУИ размещены на официальном сайте факультета

Информационные технологии ЕНУ им.Л.Н.Гумилева (проект e-content). На генераторы ИЭУИ и на разработанные ИЭУИ получены свидетельства о государственной регистрации объекта интеллектуальной собственности.

В настоящее время реализован синтезатор казахской речи для озвучивания контента ИЭУИ, также ведутся работы по подключению функции управления ИЭУИ голосом. Для контроля знаний будет подключена подсистема экспертной системы на основе нечеткой логики.

Подробно о разработанной технологии автоматизации создания ИЭУИ можно ознакомиться на сайте www.e-zerde.kz/generator.

Работа выполнена при поддержке гранта, по программе «Целевое развитие университетской науки, ориентированной на инновационный результат» по проекту «Методология, алгоритмы и программы генерации электронных учебных изданий»

ЗАКЛЮЧЕНИЕ

Применение разработанной технологии позволит преподавателю без привлечения программиста в кратчайшие сроки создавать свои локальные и интернет-ИЭУИ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Шарипбаев, 2005] Шарипбаев А.А. Автоматизация создания электронных учебных изданий / Шарипбаев А.А., Омарбекова А.С. // Вестник ЕНУ им. Л. Гумилева, 2005, С.119-125.

[Omarbekova, 2012a] A.S.Omarbekova. Automatization create electronic learning. Journal of International Scientific Publications Education Alternatives / A.S. Omarbekova , A.Seifullina. // Volume 10, Part 1, ISSN 1313-2571. Bulgaria, 2012, P.242-250

[Шарипбаев, 2012] Шарипбаев А.А. Семантическая модель электронного учебного издания / Шарипбаев А.А., Омарбекова А.С. // Спецвыпуск Вестник ЕНУ им. Л. Гумилева, 2012, С. 432-435.

[Gruber, 1991] Gruber T.R. The role of common ontology in achieving sharable, reusable knowledge bases // Principles of Knowledge Representation and Reasoning. Proceedings of the Second International// Conference. J.A. Allen, R. Fikes, E. Sandewell – eds. Morgan Kaufmann– 1991– P.601-602.

[Omarbekova, 2012b] A.S.Omarbekova. Building automation ontological generation of intellectual electronic textbooks / A.S.Omarbekova, A.Seifullina // The 6th International Conference on Soft Computing and Intelligent Systems, Kobe, Japan, 2012. P.2217-2219.

GENERATION OF INTELLECTUAL ELECTRONIC EDUCATIONAL EDITIONS ON THE BASIS OF ONTOLOGIC MODEL

Sharipbaev A.A. *, Omarbekova A.S. *, Kintonova A.J. *, Niyazova R.S. *, Barlybayev A. *

*L.N. Gumilyov Eurasian University,
Astana, Republic of Kazakhstan

sharalt@mail.ru

omarbekova_as@enu.kz

aliya_kint@mail.ru

rozamgul@list.ru

frank_ab@mail.ru

INTRODUCTION

Current work describes the methodology of creating the generator of the intellectual electronic educational editions (IEEE) for the higher education with credit technology of training. The purpose is development of the technology which allows automating the stages of creation of IEEE.

MAIN PART

The Intellectual Electronic Educational Edition (IEEE) is a set of digital, text, graphic, audio, video and other information which have means of program control and documentation, have adaptive properties and the intelligent interface and can be thus placed in the global Internet network.

CONCLUSION

Work was supported by grant under the program "Targeted Development of the University Science Focused on Innovative Result" within the "Methodology, Algorithms and Program for Electronic Educational Editions Generation" project.



УДК 004.89:004.4

СЕТЕЦЕНТРИЧЕСКАЯ СМАРТ-КОБОРГ СИСТЕМА ДЛЯ ОПЕРАТИВНОГО ДИАГНОСТИРОВАНИЯ И УПРАВЛЕНИЯ

Соловьев В.И.*

** Научно-Инженерный Центр Ассоциации содействия Всемирной Лаборатории,
г. Москва, Россия*

sicwl@newmail.ru

В статье рассмотрены вопросы применения коборг-технологии для создания сетецентрических коборг-систем, представляемых вертикальной конфигурацией неоднородных мультиагентных коборгов (МА-коборгов). Предлагаемая архитектура позволяет реализовывать принципы самоорганизации МА-коборгов для достижения поставленных «центром» задач.

Ключевые слова: коборг – технология; мультиагентные коборги; сетецентрические системы.

ВВЕДЕНИЕ

Основная сущность коборг-технологии для диагностирования и управления сложными организованными объектами (A complicated organized objects-Coborgs) была освещена в [Соловьев В.И., 2011а], [Соловьев В.И., 2011б], [Соловьев В.И., 2012]. Она заключается в представлении таких объектов как некоторого организованного единства всех согласованно действующих в них процессов, органов, систем или функциональных узлов, не относящиеся к животному или растительному миру, но обладающие основными свойствами живых организмов. В этом случае при проектировании интеллектуальных систем в некоторой проблемно-ориентированной области необходимо в составе коборга: а) выделить органы (процессы, функциональные узлы или системы); б) сформулировать перечень информативных внутренних параметров состояния (ВПС) в виде различных проектных нормативов, заданных диапазонов и траекторий, предельно допустимых минимальных или максимальных величин; в) обеспечить возможность непрерывного контроля (измерения, регистрации) ВПС в реальном времени; г) определить перечень возможных расстройств (заболеваний) или аномальных состояний коборга при выходе ВПС за установленные интервалы. Дальнейшее построение интеллектуальной системы диагностирования и управления коборгом (SmartCoborgSystem) может производиться, например, на основе процедур использования и обработки знаний в соответствующей предметной области [Уэно Х. и др., 1989].

Коборг-технология достаточно хорошо зарекомендовала себя при создании систем диагностирования и управления локальными (сосредоточенными) объектами. Здесь укажем на металлургические агрегаты (доменные печи, агломерационные машины).

Основная часть

В данном докладе рассмотрена возможность применения указанного подхода к созданию интеллектуальных систем оперативного диагностирования и управления сложными мультиагентными объектами, находящиеся в едином сетевом пространстве, которые мы в соответствии с основной концепцией коборг-технологии назовем мультиагентными коборгами (МА-коборги). МА-коборгами могут быть:

- действующая группа интеллектуальных роботов - агентов;
- действующие отряды (соединения) локальных техно-коборгов (надводных и подводных судов, самоходных наземных машин, воздушных аппаратов), а также любое неоднородное сетецентрическое пространство;
- подразделение (коллектив, команда, персонал), выполняющее определенное задание или решающая определенную задачу.

Здесь на примере создания интеллектуальной системы оперативного управления деятельностью металлургического предприятия, рассмотрим неоднородное сетецентрическое пространство, включающее в себя на нижнем уровне в качестве программных агентов локальные техно-коборги

(технологические агрегаты) или отделы и участки, оснащенные автоматизированными рабочими местами (АРМ). На следующих "этажах" сети находятся соответственно цеховые коборги, коборги производств и, наконец, основные функциональные коборги, отражающие главные области деятельности предприятия: производство, коммерцию и финансы. Таким образом, создается структура "матрешечного" типа, в которой каждый локальный программный агент, оставаясь частью общей системы, осуществляет самостоятельно свои функциональные задачи.

На рисунке 1 представлен фрагмент сетевидной архитектуры предприятия.

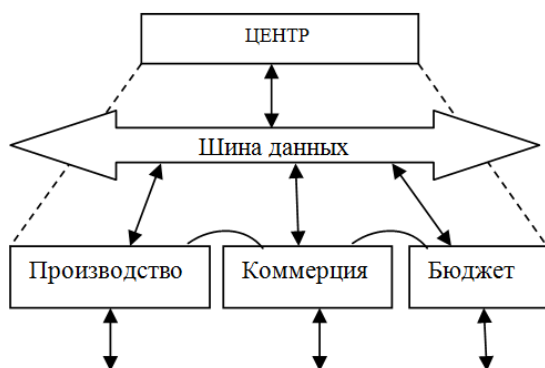


Рисунок 1 - Фрагмент (верхний этаж) сетевидной архитектуры управления предприятием

Известны теоретические и прикладные разработки [Скобелев П.О., 2010], [Феоктистов А.Л. и др., 2012] применения сетевидного подхода для целей управления сложными распределенными объектами. Сущность его заключается «...в организации управления сложными процессами в распределенной коммуникационной инфраструктуре, реализующей максимальную ситуационную осведомленность каждого узла и переходу к работе каждого узла в режиме самоорганизации на достижение поставленных задач» [Шабунин Б.А. и др., 2012].

Основным отличием предлагаемой сетевидной архитектуры от общеизвестной заключается в том, что элементы и узлы на каждом «этаже» сети представляются сложными организованными объектами – коборгами. Такое представление позволяет возложить на коборг не только функцию планировщика, но и осуществлять диагностирование и прогноз текущего состояния коборга на предмет его функционального расстройства (заболевания), выявлять причины расстройства или аномального состояния и своевременно формировать параметрические и симптоматические воздействия по ликвидации развивающегося расстройства.

С точки зрения коборг-технологии сетевидное пространство на каждом этаже представляется в виде автономного МА-коборга, включающего коборг-координатор с соответствующим числом нижележащих

интеллектуальных коборгов-исполнителей (на рисунке 1 выделены пунктирной линией). В свою очередь, коборги-координаторы и коборги-исполнители изменяют соответственно своё назначение на других этажах сети.

Такой подход позволяет рассматривать МА-коборг как некое типовое самостоятельное подразделение, встроенное в общую архитектуру системы управления, соответствующее организационной структуре предприятия. Отметим основные отличительные свойства такого МА-коборга:

1. Каждый локальный коборг, входящий в состав МА-коборга имеет свою структуру и органы (узлы, компоненты, процессы) в соответствии со своей физической сущностью. Так, основными исполнительными органами координационного МА-коборга «ХЭД», представляющего оперативное состояние хозяйственно-экономической деятельности предприятия являются коборги «ПРОИЗВОДСТВО», «КОММЕРЦИЯ» и «БЮДЖЕТ». В свою очередь, основными органами координационного МА-коборга "ПРОИЗВОДСТВО" являются цеховые исполнительные коборги: "Доменное производство", "Сталеплавильное производство". "Прокатное производство", "4-ый передел". И наконец, в состав цеховых координационных МА-коборгов, например, "Доменный цех", входят исполнительные технокоборги: "Доменные печи".

Органами координационного МА-коборга «БЮДЖЕТ» являются функциональные исполнительные коборги: а) "финансовые обязательства предприятия"; б) "встречные обязательства" (платежи по обязательствам получателей); в) "ожидаемые финансовые резервы"; г) "фактические финансовые резервы". На следующем нижнем этаже исполнительными коборгами будут являться органы: "Приход", "Расход", "Отгрузка товаров и услуг", "Поступление платежей", в состав которых, в свою очередь, входят соответственно исполнительные коборги по основным статьям обязательств и расходов предприятия, получателям и плательщикам.

Органами координационного МА-коборга «КОММЕРЦИЯ» являются функциональные исполнительные коборги: а) "Сбыт готовой продукции"; б) "Поступление сырья"; в) "Отгрузка сырья поставщиками". На следующем нижнем этаже в состав координационных МА-коборгов: "Сбыт готовой продукции"; "Поступление сырья"; "Отгрузка сырья поставщиками" входят соответственно исполнительные коборги по видам и количеству отправляемой готовой продукции, поставляемого на предприятие и отгружаемого поставщиками сырья и материалов.

2. Функциональная структура МА-коборга содержит следующие два типа ВПС:

- а) индивидуальные параметры состояния (ИПС)

каждого исполнительного коборга в составе соответствующего функционального органа МА-коборга. Индивидуальными параметрами состояния могут являться:

- на уровне технологических агрегатов (техно-коборгов) – тепловые, механические и газодинамические показатели, характеризующие ход технологического процесса (характер движения шихты, химический состав и изменение содержания газов в процессе плавки металла, температуры нагрева зон печи и прокатываемого металла и др.);
- на уровне цехового коборга – текущие значения вероятности расстройств или заболеваний каждого техно-коборга, входящего в состав цеха;
- на уровне коборга «ПРОИЗВОДСТВО» – текущие значения вероятности расстройств или нежелательного состояния каждого цехового коборга, входящего в состав производства;
- на уровне коборга «ХЭД» – текущие значения вероятности расстройств или нежелательного состояния производственной, коммерческой и финансовой (здесь бюджетной) деятельности предприятия.

б) координатные параметры состояния (КПС), отражающие общие выходные показатели координирующего МА-коборга. На цеховом и общепроизводственном уровнях таковыми являются: выпуск готовой продукции, отправка готовой продукции со склада, удельные расходы сырья и материалов на производство (эффективность производства). На уровне коборга «ХЭД» этими показателями в оперативном плане могут являться доход и затраты. КПС используются при диагностировании состояния МА-коборга для корректирования (усиления или ослабления) влияния коэффициентов уверенности вывода на возможное изменение состояния коборга. Текущие значения и характер изменения КПС и ИПС отражают состояние и определяют результат деятельности и функционирования мультиагентного объекта на все этапы его жизненного цикла. Очевидно, что независимо от назначения МА-коборга его эффективное функционирование возможно тогда, когда он находится в нормальном состоянии. Нормальное (заданное) состояние МА-коборга, соответствующее достижению или приближению к поставленной цели (целевой функции), обеспечивается в случае, когда его основные ВПС находятся в заданных интервалах. Так заданные интервалы ИПС удобно представлять в виде проектных нормативов или оперативных заданий каждому интеллектуальному исполнительному агенту с учетом его принадлежности к определенному органу МА-коборга. Основные принципиальные технико-алгоритмические решения по диагностированию и управлению МА-коборгом представлены на рисунке 2.

Здесь: МА-Коборг – мультиагентный сложный организованный объект, состоящий из коборга –

координатора и индивидуальных (локальных), в общем случае, неоднородных коборгов;

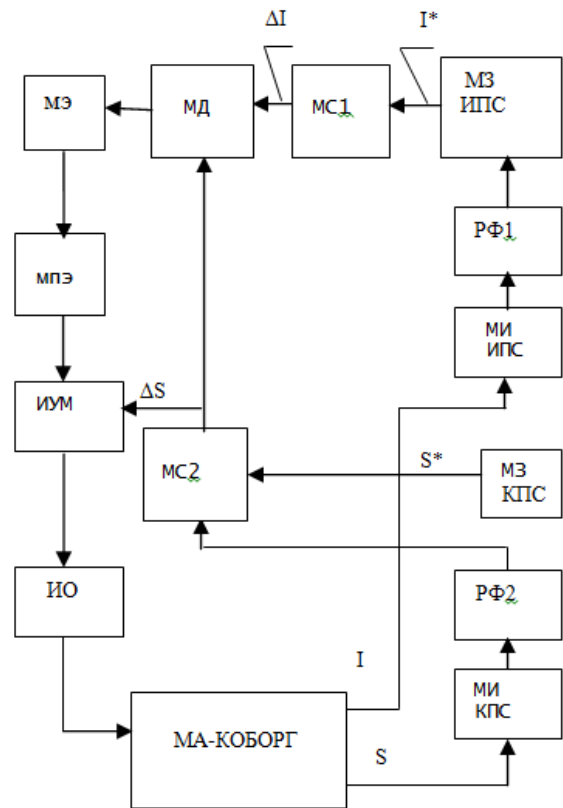


Рисунок 2 - Схема диагностирования и управления МА-коборгом

S^* , I^* , S , I – соответственно заданные и текущие значения координатных и индивидуальных параметров состояния МА-коборга; ΔS и ΔI – отклонения текущих значений параметров состояния от заданных; U – управляющие воздействия. МЗ КПС и МЗ ИПС – модули заданных координатных и индивидуальных параметров состояния МА-коборга; МИ КПС и МИ ИПС – модули измерения текущих значений координатных и индивидуальных параметров состояния; МФ-1 и МФ-2 – робастные фильтры; MC1 и MC2 – модули сравнения; МД – модуль диагностирования текущего состояния МА-коборга; МЭ-модуль экстраполяции; МПЭ – модуль пороговых элементов; ИУМ – информационно-управляющий модуль; ИО – исполнительный орган.

3. Под функциональным расстройством или нежелательным изменением состояния МА-коборга будем понимать нарушение нормального функционирования какого-либо органа или какой-либо системы, входящих в состав этого МА-коборга, носящее периодический или постоянный характер. Как правило, расстройство организма связано с закономерным сочетанием и развитием симптомов расстройства или заболевания, представляющих собой устойчивые отклонения текущих значений ВПС от заданных диапазонов. В коборгах используются так называемые объективные симптомы, наблюдаемые в процессе его эксплуатации и тестирования в реальном времени. Отметим, что полезную информацию о

дополнительных характеристиках симптома (устойчивость тренда, дисперсия, скорость нарастания и др.) можно извлекать по всему диапазону текущего изменения ВПС. Система диагностирования производит выявление и оценку состояния каждого органа или процесса по текущему значению и тренду кривой вероятности развития расстройства в соответствии с процедурами (1) представления и использования знаний в соответствующей предметной области. Затем принимается решение о принадлежности текущего состояния МА-коборга к заранее определенному нечеткому терму состояния типа: «нормальное», «удовлетворительное», «неудовлетворительное», «критическое». Это позволяет оперативно оценивать как текущее состояние «здоровья» действующего МА-коборга, как в целом, так и его отдельных органов. Очевидно, что для таких сложных организованных мультиагентных объектов, не имеющих в отличие от медицины своей собственной терминологии расстройств и заболеваний, эти приемы нечеткой логики при оперативном диагностировании состояния остаются пока единственным решением.

Каждое из состояний, в котором может находиться коборг, характеризуется определенным диапазоном, так называемого коэффициента уверенности вывода CF_i , принимающего значения из интервала $[-1,1]$. При этом CF_i , принадлежащее к интервалу $(0,1]$, интерпретируется как вероятность наличия одного из этих состояний, а абсолютное значение CF_i , принадлежащее к интервалу $[-1,0)$, – как вероятность отсутствия какого-нибудь из этих состояний. Количественная оценка достоверности вывода о развитии расстройства функционирования коборга производится по следующим процедурам нечеткой логики MYCIN [Shortliffe E. H., 1976]:

1. При связи логического И (END) :
 $CF[A] = CF[X \text{ и } Y, .] = \min\{CF[X, .], CF[Y, .]\}$
 2. При связи логического ИЛИ (OR):
 $CF[A] = CF[X \text{ или } Y, .] = \max\{CF[X, .], CF[Y, .]\}$
 3. При комбинированной связи COMB (И - ИЛИ):
 - а) $CF[A, (X,Y)] = +1$, если $CF[A,X] = 1$ или $CF[F,Y] = 1$
 - б) $CF[A, (X,Y)] = CF[A,X] + CF[A,Y] - CF[A,X] \cdot CF[A,Y]$ (1)
- если $CF[A,X] > 0$ и $CF[A,Y] > 0$;
- с) $CF[A, (X,Y)] = CF[A,X] + CF[A,Y]$,
 если $CF[A,X] \neq \pm 1$ и $CF[A,Y] \neq \pm 1$, а $CF[A,X] \cdot CF[A,Y] \leq 0$;
 - д) $CF[A, (X,Y)] = CF[A,X] + CF[A,Y] + CF[A,X] \cdot CF[A,Y]$,
 если $CF[A,X] < 0$ и $CF[A,Y] < 0$;
 - е) $CF[A, (X,Y)] = -1$, если $CF[A,X] = -1$ или $CF[A,Y] = -1$,

где $CF[A, X]$ и $CF[A, Y]$ – коэффициенты уверенности вывода развития расстройства A , определяемые экспертными знаниями или промежуточными результатами доказательств на непрерывном интервале $[-1 + 1]$ от соответствующих симптомов x и y , появление которых обуславливается отклонением текущих значений параметров состояния объекта от установленного диапазона. Коэффициент CF , полученный из трех и более независимых доказательств выводится последовательно, используя указанные выше формулы.

Пример результата оперативного диагностирования МА-Коборга на высшем этаже сетецентрической сети предприятия приведен на рисунке 3.



Рисунок 3 - Диаграмма оперативного диагностирования МА-Коборга

На рисунке 3:

- «б» – текущее состояние исполнительного коборга «БЮДЖЕТ»;
- «п» – текущее состояние исполнительного коборга «ПРОИЗВОДСТВО»;
- «к» – текущее состояние исполнительного коборга «КОММЕРЦИЯ»;
- «хэд» – текущее состояние центрального координирующего коборга «ХЭД» (хозяйственно-экономическая деятельность предприятия).

4. В общем случае онтологический базис диагностирования и управления мультиагентными коборгами представляется некоторым множеством знаний в области диагностирования текущего состояния определенного функционального коборга, о числе и характере контролируемых внешних и внутренних возмущений, способов и приемов применения управляющих воздействий по ликвидации намечающихся расстройств или заболевания коборга. Общая функциональная структура любой коборг-системы содержит:

- а) модуль обработки входной информации (прием в реальном или псевдореальном времени измеренных или регистрируемых входных параметров); б) процессор искусственного интеллекта для формирования базы знаний на внешнем языке представления знаний, подсистему контроля и отладки базы знаний, отображения базы знаний и результатов диагностирования; в) базы

знаний расстройств (заболеваний) коборга. Представляется так называемым «коллективным экспертом», содержащего в себе некую совокупность знаний, извлекаемых из прикладных трудов предметной области, технологических и других инструкций типовых положений, правил, стандартов, а также знаний экспертов; г) базы знаний контролируемых внутренних и внешних возмущений (причин) расстройства или заболевания коборга; д) базы знаний управляющих воздействий. В оперативном плане в зависимости от степени расстройства или заболевания коборга применяются соответственно симптоматические или параметрические воздействия для ликвидации симптомов или причин, вызвавших расстройство функционирования коборга; е) архивный модуль для хранения непрерывных реализаций кривых ВПС, выходных параметров и фактических управляющих воздействий; ж) модуль развития и коррекции базы знаний для корректировки CFi порождающих правил.

5. Для работы с неоднородными объектами в системе использованы связанные друг с другом различные математические формализации и экспертные знания. В качестве формальной базовой спецификации, пригодной для описания математических моделей, произвольных вычислений, диалога и экспертных знаний в реальном времени используется математический аппарат функциональных сетей (ФС) [Юрченко В.В., 1992], который в качестве средства представления знаний открывает большие возможности в автоматизации построения систем искусственного интеллекта. База знаний диагностирования состояния МА-коборга представляет собой набор продукций, позволяющий сочетать логический вывод и вычисления коэффициентов уверенности CFi. Она имеет многослойную структуру, схематически изображённую на рисунке 4. Первый (нижний) слой образуют N блоков выходных переменных – результат измерений (x^j_1), причем x^j_1 может принимать несколько значений. Каждый блок входных переменных считывает результаты измерений одного из значений параметров соответствующего коборга. Так, например, группа $x^1_i = (x^1_i)$, $i=1, \dots, 9$ служит для описания параметров текущей отгрузки готовой продукции. При этом x^1_1 задает диапазон изменения количества отгружаемой продукции, x^1_2 – тенденцию изменения, а x^1_3 – описывает ритмичность отгружаемой продукции.

При этом: x^1_1 – диапазон по отгрузке имеет три альтернативы: а) заданное значение; б) выше заданного значения; в) ниже заданного значения.

x^1_2 – тенденция изменения отгрузки может принимать следующие значения: а) увеличивается; б) уменьшается; в) не меняется.

x^1_3 – показатель ритмичности : а) меньше или равно 0,5; б) больше 0,5, но меньше 0,8 .;

в) больше 0,8.

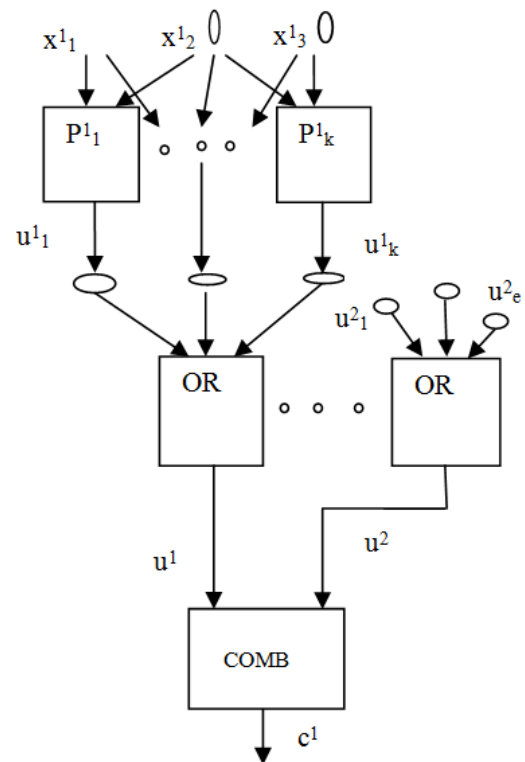


Рисунок 4 - Фрагмент базы знаний диагностирования коборга

Второй слой образуют блоки продукций P_{ij} вида:

IF «условие» THEN $u_i = a$ ELSE $u_i = 0$.

В качестве условия используются предикаты вида $u_{ij} = \langle \text{значение} \rangle$, объединенные логическим "И". Результатом выполнения продукции является значение переменной, имеющее смысл промежуточного коэффициента уверенности. Эти коэффициенты определяются экспертом при вводе правил в базу знаний. k-тый блок образуют продукции, в условиях которых используются переменные из k-того блока переменных. Фактические элементы j-того блока продукций описывают воздействие j-того параметра на состояние коборга.

Третий слой образуют блоки так называемых OR-правил. Эти правила выполняются по следующему алгоритму. Пусть s_1, \dots, s_n – входные переменные OR-правила, а s – выходная переменная. Тогда:

- 1) если все $s_i = 0$, то $s = 0$.
- 2) если $s_i \leq 0$, то в качестве s принимается максимальная отрицательная s_i .
- 3) если s_i имеют положительные, нулевые и отрицательные значения, то $s = \max s_i$.

Как видно из рисунка 4 OR-правила используются для объединения результатов, получаемых во втором слое. Четвертый и последующие слои образуют блоки так называемых COMB-правил. Эти правила выполняются в соответствии с алгоритмом (1). Многослойная структура позволяет удобно добавлять, расширять или удалять блоки правил в любом слое.

Заключение

Предложенная Коборг-технология позволяет:

- 1) создать универсальную основу при проектировании интеллектуальных систем оперативного диагностирования и управления сложными организованными объектами различного назначения (локальными коборгами), в том числе для неоднородных мультиагентных объектов сетцентрической архитектуры (МА-коборгами);
- 2) представление сетцентрической системы в виде совокупности многоуровневых сложных организованных мультиагентных объектов (МА-Коборгов) из центрального коборга и исполнительных интеллектуальных агентов (локальных коборгов). Это позволяет возложить на коборг не только функцию планировщика, но и оперативно осуществлять диагностирование и прогноз текущего состояния коборга на предмет его функционального расстройства (заболевания), выявлять причины расстройства или аномального состояния и своевременно формировать параметрические и симптоматические воздействия по ликвидации развивающегося расстройства.

Библиографический список

- [Соловьев В.И., 2011] Smart Coborg Systems. Материалы международной научно-практической конференции OSTIS-2011. г. Минск. Секция 8.
- [Соловьев В.И., 2011] Интеллектуальная система управления сложными организованными объектами (коборгами). Патент на изобретение RU №2435187 C2, 2011г.
- [Соловьев В.И., 2012] Интеллектуальные мультиагентные коборг-системы. Интеллектуальные системы: Труды Десятого международного симпозиума / Под ред. К.А. Пупкова.- М: РУСАКИ, 2012. С.254 – 258.
- [Уэно Х. и др., 1989] Представление и использование знаний. Под ред. Уэно Х, Исидзука М. – М.: Мир, 1989.
- [Скобелев П.О., 2010] Мультиагентные технологии в промышленных приложениях: к 20-летию основания Самарской школы мультиагентных систем. – «Мехатроника, Автоматизация, Управление», №12, 2010. – с. 33-46.
- [Феоктистов А.Л. и др., 2012] Разработка принципов построения многоуровневой мультиагентной системы для управления проектами и ОКР РКК «ЭНЕРГИЯ» - Проблемы управления и моделирования в сложных системах: Труды XIV Международной конференции (19-22 июня 2012г., Самара, Россия / Под ред.: акад. Е.А. Федосова, акад. Н.А. Кузнецова, проф. В.А. Виттиха. – Самара: Самарский научный центр РАН, 2012. с.718-723
- [Шабунин Б.А. и др., 2012] Сетцентрический подход к созданию распределенных систем управления ресурсами ОАО «РЖД» на основе мультиагентных технологий. - Проблемы управления и моделирования в сложных системах: Труды XIV Международной конференции (19-22 июня 2012г., Самара, Россия / Под ред.: акад. Е.А. Федосова, акад. Н.А. Кузнецова, проф. В.А. Виттиха. – Самара: Самарский научный центр РАН, 2012. с.724-734

[Shortliffe E. H., 1976] Computer-based Medical Consultations: MYCIN, Elsevier, New York.

[Юрченко В.В., 1992] Функциональные сети. - М: Наука, 1992. – 184 с.

NETWORK-CENTRIC COBORG-SYSTEM FOR OPERATIONAL DIAGNOSIS AND CONTROL

V.I.Soloviev

*Association for World Laboratory Assistance
Science and Engineering Center,
Moscow, Russia*

sicwl@newmail.ru

The paper describes approach creating network-centric system based on multi-agent MA-Coborgs. The offered Network – Centric architecture permits to realize principles of self-organizing MA-Coborgs for achievement of the tasks delivered by «Centre».

INTRODUCTION

The paper considers a system of diagnosis and management of a complicated organized multi-agent objects (MA-Coborgs). MA-objects are understood to be a certain unity of all the processes, organs and systems acting inside it. They belong neither to animal nor organic, vegetable, life, nevertheless they possess the main characteristics of life organisms. We name objects of this type Coborgs (A complicated organized objects) and the systems which operate them Smart Coborg Systems.

MAIN PART

Elements and nodes network-centric are submitted at every level by complicated organized objects-Coborgs. It allows not only accomplish planner function, but also include a self-diagnosis and prognosis status of the coborg, identify the causes of a disorder or a disease and timely form parametric and symptomatic impacts on the fault disorders. Group of Coborgs on every level forms the so-called multi-agent coborg (MA-Coborg), consisting of coordinating Coborg and executive Coborgs, which are bodies of MA-Coborg. The creation of intellectual systems of diagnostic and operational control was based on mathematical methods of fuzzy-logic, use knowledge the so-called collective expert.

CONCLUSION

Proposed Coborg-technology allows a universal basis for the designing intelligent systems for diagnosis and control of a complicated organized objects of different purposes (local Coborgs), including multivendor multi-agent objects Network-Centric architecture (MA-Coborgs). The offered Network Centric architecture permits to realize principles of self-organizing MA-koborgs for achievement of the tasks delivered by «Centre».



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ИНТЕЛЛЕКТУАЛЬНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ПО СОВРЕМЕННОМУ ИСКУССТВУ

Лейченко А.Н.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

nastassia.leichanka@gmail.com

В работе приводится описание процесса создания прототипа интеллектуальной справочной системы по современному искусству, который начинается с анализа актуальности поставленной проблемы и ее сложностей, а также выбора наиболее подходящих технологий для создания подобной системы. Далее в работе приводится описание результата проектирования структуры онтологии и ее реализации при помощи Protégé. В конце статьи приведен анализ полученных результатов.

Ключевые слова: интеллектуальная система; информационно-справочная система; база знаний по современному искусству; онтология; инженерия знаний.

ВВЕДЕНИЕ

В настоящее время большую актуальность приобретает использование технологий искусственного интеллекта для повышения качества и скорости поиска информации. Создаются информационно-справочные системы по определенным предметным областям. Существуют и справочные системы обобщенного типа (например, по всем естественным наукам), однако качество и глубина проработки информации в них существенно ниже, чем в специализированных системах.

Целевой группой могут быть при этом как работники специализированных научных организаций и учреждений, так и простые пользователи интернета.

Основной вопрос при построении электронных информационно-справочных систем состоит в том, как организовать большие количества информации так, чтобы пользователи могли найти то, что им нужно. При этом важно, чтобы метаданные, используемые в системах, были приспособлены к машинной обработке. Кроме того, общим требованием для таких систем является совместимость на уровне обмена данными и совместной работы сервисов. Чтобы достичь семантической совместимости, смысл информации, которой обмениваются, должен быть понятен во всех системах. Использование онтологий для объяснения неявного и скрытого знания – возможный подход для достижения этой цели.

Предметная область, которая была выбрана для данного проекта обладает некоторой особенностью – она принадлежит гуманитарным наукам. В настоящее время существует уже значительное количество интеллектуальных систем по техническим наукам и медицине, в то время как гуманитарные знания гораздо реже представлены на языке представления знаний.

В отличие от технических наук, источником информации для которых являются эксперимент и наблюдение и данные в которых являются надежными и логически взаимосвязанными, информация в гуманитарных науках, в том числе и по искусству, часто субъективна. Главным средством утверждения представлений в сознании человека является их образность, необходимо включающей такие моменты, как эмоции и сотворчество. Сотворчество ведет к идеализации, упорядочению и закреплению однажды возникших образных представлений, а эмоции играют в этом процессе важнейшую роль движущих сил [Фоминых, 2002].

Благодаря этим особенностям предметная область современного искусства, и гуманитарных наук в целом, предполагает другой образ формализации знаний, нежели технические науки, однако интерес к этому растет, а так же повышается необходимость в создании интеллектуальных систем по гуманитарным наукам для систематизации знаний.

В данной статье будет описан процесс создания прототипа интеллектуальной информационной

системы по современному искусству, который включает анализ и выбор технических средств, проектирование онтологий, а также реализацию прототипа системы при помощи выбранного инструментария. Полученные результаты будут проанализированы.

1. Анализ технологий инженерии знаний

Предварительный анализ методов и моделей представления знаний показал, что наиболее качественно предметная область современного искусства может быть представлена при помощи сетевой либо фреймовой моделей представления знаний, так как эти модели достаточно гибки, универсальны и позволяют вводить различные типы отношений. Этот вывод повлиял на последующий выбор наиболее подходящих инструментальных средств для реализации системы, анализ которых приведен ниже.

1.1. Платформа Protégé

Платформа Protégé [Musen, 1998] – бесплатная, свободно распространяемая платформа, предоставляющая пользователям набор инструментов для работы с приложениями на основе онтологий, а именно для создания и поддержки явных моделей предметной области и включения этих моделей непосредственно в программный код. Программа была разработана группой биомедицинской информатики Стенфордского университета и предназначена для создания, визуализации и манипуляции онтологиями прикладной области. Protégé включает редактор онтологий, позволяющий проектировать онтологии разворачивая иерархическую структуру абстрактных или конкретных классов и слотов.

Структура онтологии организована аналогично иерархической структуре каталога. На основе сформированной онтологии, Protégé может генерировать формы получения знаний для введения экземпляров классов и подклассов. Инструмент имеет дружелюбный интерфейс, удобный для использования неопытными пользователями, снабжен справками и примерами. Возможности программы могут быть расширены за счет использования плагинов (для визуализации либо поддержки особых форматов).

Платформа Protégé выпущена в нескольких версиях. Более ранние, такие как Protégé 3.1-3.2, основаны на фреймовой модели представления знания ОКВС, более поздние версии основаны на представлении знаний согласно OWL [Chaudri, 1998].

Уже разработанный ряд плагинов [Noy, 2001] позволяет адаптировать Protégé для редактирования моделей хранимых в разных форматах (стандартный текстовый, в базе данных JDBC, UML, языков XML, XOL, SHOE, RDF и RDFS, DAML+OIL).

Другим преимуществом Protégé является его

трехуровневая архитектура [Овдей, 2004], которая позволяет осуществить разделение между хранением онтологий, модулями логики приложений и приложениями интерфейса пользователя.

В то время, как большинство инструментов инженерии онтологий хранит свои онтологии в текстовых файлах, в результате чего размер онтологий сильно ограничен, Protégé позволяет хранить онтологии в базах данных [Овдей, 2004], что позволяет управлять большими онтологиями.

Protégé также имеет графические средства, позволяющие просматривать и редактировать онтологии. При этом классы представляются узлами на графах, а отношения – дугами между ними.

Платформа имеет несколько встроенных машин вывода, таких как FACT++, HermiT. Однако, при помощи плагинов может быть также обеспечена возможность использования и более средств.

1.2. Система OntoEdit

Система OntoEdit [Sure, 2002] была разработана в институте AIFB (Institute of Applied Informatics and Formal Description Methods) Университета Karlsruhe, а затем передана коммерческой компании Ontoprise GmbH, которая впоследствии лицензировала OntoEdit Professional.

Система позволяет выполнять проверку, просмотр, кодирование и модификацию онтологий. В настоящее время OntoEdit поддерживает языки представления: FLogic, включая машину вывода, OIL, расширение RDFS и внутреннюю, основанную на XML, сериализацию модели онтологии используя OXML - язык представления знаний OntoEdit (OntoEdit's XML-based Ontology representation Language).

К достоинствам инструмента можно отнести удобство использования, разработку онтологии под руководством методологии и с помощью процесса логического вывода, разработку аксиом, расширяемую структурой посредством плагинов, а также очень хорошую документацию.

Так же как и Protégé, OntoEdit – автономное Java-приложение, которое можно локально установить на компьютере, но его коды закрыты. Архитектура OntoEdit подобна Protégé и имеет три уровня, разделяющие хранение онтологий, модулями логики приложений и приложениями интерфейса пользователя. Онтологии, в отличие от Protégé, хранятся в текстовых файлах, что автоматически ограничивает их размер [Овдей, 2004].

Существует две версии OntoEdit: свободно распространяемая OntoEdit Free (ограничена 50 концептами, 50 отношениями и 50 экземплярами) и лицензированная OntoEdit Professional (без ограничений). OntoEdit Professional имеет к тому же более широкий набор функций и возможностей, такие как машина вывода, графический инструмент

запросов, больше модулей экспорта и импорта, графический редактор правил, поддержка баз данных JDBC и т.д..

1.3. Технология OSTIS

Проект OSTIS является разработкой кафедры интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники. OSTIS (Open Semantic Technology for Intelligent system) – это массовая семантическая технология компонентного проектирования интеллектуальных систем различного назначения [OSTIS, 2012]. Разработчики данного проекта стремятся создать комплексную и активно развивающуюся технологию, которая включает в себя теоретические и практические, программные и аппаратные аспекты. Кроме того, целью является создание инфраструктуры, обеспечивающей сочетание научной, учебной, инженерной и коммерческой деятельности в области искусственного интеллекта. Проект OSTIS является открытым и предоставляет полную документацию по всем его компонентам, включая исходные тексты программных средств. Проект ориентирован на широкий круг разработчиков прикладных интеллектуальных систем, а также на массовое распространение разрабатываемой технологии [OSTIS, 2012].

Для представления знаний используется семантические сети. При этом, пользовательский интерфейс, базы знаний и машина обработки знаний, описываются с помощью языка SC. SC-код (Semantic Code) представляет собой универсальный способ представления семантических сетей в виде однородных семантических сетей с базовой теоретико-множественной семантической интерпретацией [Голенков, 2011], информационными конструкциями которого являются SC-графы.

Поскольку алгоритмы обработки знаний представлены на том же языке, что и сами знания, существует возможность модифицировать сами алгоритмы прямо в процессе их работы, что позволяет строить программы, способные к самоконфигурации в процессе работы [Голенков, 2011], [Шункевич, 2013]. Одним из важных принципов технологии является возможность использовать повторно уже разработанные модули и компоненты, хранящиеся в библиотеках совместимых компонентов.

2. Проектирование онтологии по современному искусству

Предметная область современного искусства является достаточно широкой, и, как любая гуманитарная предметная область, она не обладает строгостью технических наук. Отсутствие ярко выраженных аксиом, однако, не означает, что знания об искусстве сугубо субъективны [Golitsyn, 1995]. Произведения искусства обладают множеством

точных характеристик. Научные знания об искусстве выводятся на основе максимально объективного анализа произведений искусства.

Предметная область современного искусства ограничена временным фактором. Современным традиционно принято считать искусство, начиная с фовистов и немецкого экспрессионизма, то есть с начала двадцатого века [Chilvers, 1997; Davies, 2007]. При этом не столько период деятельности художника, сколько его стиль является определяющим. При проектировании данной онтологии было уделено внимание изобразительному искусству (живописи, графике, скульптуре), а также таким современным формам искусства как инсталляция, перформанс и хэппенинг.

В результате анализа предметной области было выделено множество понятий и отношений между ними. Каждое выделенное понятие входит в один из следующих суперклассов:

- персоны;
- объекты;
- художественные признаки;
- институты.

Иерархию классов обеспечивает отношение «класс-подкласс». Рассмотрим вышеуказанные суперклассы и входящие в них понятия подробнее.

2.1. Понятия, входящие в суперкласс «Персоны», и отношения между ними

Суперкласс «Персоны» включает понятия, соответствующие участникам процесса создания, созерцания, оценивания и изучения искусства:

- художник – создатель произведений искусства;
- модель – человек, позирующий перед художником при создании произведения искусства;
- критик – журналист, анализирующий произведение искусства в средствах массовой информации;
- теоретик – ученый, изучающий искусство.

В пределах данного суперкласса можно выделить следующие отношения: симметричное отношение «работал в паре/группе с», связывающее сотрудничающих художников либо теоретиков; симметричное отношение «работал с», связывающее художников и моделей; отношения «вдохновил» и обратное ему «был вдохновлен» между художниками либо художником и произведением искусства; отношения «написал критическую статью о», «отозвался положительно о», «отозвался отрицательно о», «взял интервью у» между экземплярами классов «Критик» и «Художник», а также соответствующие им обратные отношения; отношения «изучал», «написал научную статью о», «написал книгу о», связывающие теоретиков с художниками либо произведениями искусства, а также соответствующие им обратные отношения; рефлексивные отношения «ссылается

на/опирается на» и «противоречит», связывающие теоретиков между собой и теоретиков с научными изданиями.

2.2. Понятия, входящие в суперкласс «Объекты», и связанные с ними отношения

В пределах данного суперкласса можно выделить иерархию следующих понятий:

- произведение искусства:
 - картина;
 - скульптура;
 - инсталляция;
 - видео;
 - фотография;
 - рисунок;
 - графика;
 - хэппенинг;
 - перформанс;
- художественный объект: включает те же самые классы, что и «Произведение искусства», кроме «Хэппенинг» и «Перформанс» и соответствует материальному результату создания произведения искусства;
 - копия – копия произведения искусства;
 - научное издание – статья, книга теоретика;
 - критическая статья – статья критика.

Понятиям данного суперкласса соответствуют следующие отношения: «имеет автором» (и обратное ему отношение «является автором») между художником и произведением, научным изданием и теоретиком, критической статьей и критиком; «представлен» (и обратное функциональное отношение «представляет») между хэппенингом либо перформансом и художественным объектом, являющимся его результатом; «цитирует» и «процитировано в», «выражает мнение о» между двумя произведениями искусства; «изображен на» (и обратное отношение «изображает») между моделью и произведением; функциональное отношение «является копией» между копией и оригинальным художественным объектом, а также обратное ему «имеет копию»; «проанализировано в» между произведением искусства либо художником и научным изданием либо критической статьей.

2.3. Понятия, входящие в суперкласс «Признаки», и связанные с ними отношения

В данный класс входят следующие понятия, соответствующие художественным признакам произведений искусства:

- мотив – то, что физически представлено произведением;
- тема – тема, которую художник затрагивает конкретным произведением искусства;
- тип изображения – фигуративный, абстрактный и переходные формы;
 - жанр;
 - палитра – использованные цвета;
 - композиция;
 - стиль;

- стилистические характеристики – характерные признаки стилей;
- цель/сущность произведения искусства;
- техника – техника, при помощи которой художник создал произведение искусства;
- материал:
 - носитель – материал, на котором изготовлено произведение искусства;
 - средство – материал, которым создано произведение искусства;
- функция произведения искусства (например, эстетическая, философская);
- роль художественного объекта (например, этюд).

Отношениями, связывающими понятия данного суперкласса между собой, а также с понятиями других суперклассов, являются: функциональное отношение «изображает», связывающее произведение искусства и мотив; «имеет тему», связывающее произведение искусства и тему, научное издание либо критическую статью с художником либо любым подклассом суперкласса «Признаки»; транзитивное отношение «является примером/характеризуется» между произведением искусства и типом изображения, жанром, палитрой, композицией, стилем, техникой, а также между стилем и стилистическими признаками; «имеет цель» между произведением искусства и целью; функциональное отношение «сделано на», связывающее художественный объект и носитель; «сделано из» между художественным объектом и средством; «имеет функцию», связывающее произведение искусства и функцию; «играет роль» между художественным объектом и ролью.

2.4. Понятия, входящие в суперкласс «Институты», и связанные с ними отношения

Данный суперкласс включает следующие понятия:

- музей – некоммерческий институт, выполняющий роль хранилища художественных объектов, а также выставляющий их на обозрение публики;
- галерея – коммерческий институт, выполняющий схожие с музейными функции;
- прочие владельцы – частные коллекции;
- выставка – временная структура, существующая зачастую самостоятельно и не привязанная к конкретному музею либо галерее.

Важными отношениями для понятий суперкласса «Институты» являются следующие: функциональное отношение «принадлежит», связывающее художественный объект с музеем, галереей либо прочими владельцами; функциональное отношение «специализируется на» между музеем, галереей либо прочими владельцами и художником, жанром, стилем; отношение «выставлен на» между художественным объектом и выставкой, а также обратное ему отношение «имеет в составе»; отношение «проходит в» (и обратное

ему отношение «проводит»), связывающее экземпляры классов «Выставка» и «Музей» либо «Галерея».

2.5. Классы запросов

В результате анализа структуры онтологии и предметной области были выделены следующие классы запросов к системе:

1. вывести экземпляры класса;
2. вывести все экземпляры, связанные с определенным отношением;
3. вывести все классы, связанные с определенным отношением;
4. вывести классы экземпляра;
5. вывести суперклассы класса;
6. вывести подклассы класса;
7. вывести все экземпляры, связанные определенным отношением с определенным классом;
8. вывести все классы, связанные определенным отношением с определенным классом;
9. вывести все экземпляры, связанные определенным отношением с определенным экземпляром;
10. вывести экземпляры, удовлетворяющие пересечению условий (определенных отношений с конкретными экземплярами и классами).

3. РЕАЛИЗАЦИЯ ИНТЕЛЛЕКТУАЛЬНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО СОВРЕМЕННОМУ ИСКУССТВУ

3.1. Обоснование выбора инструментального средства

Для реализации прототипа интеллектуальной справочной системы по современному искусству была выбрана программная среда Protégé. Это программное средство имеет следующие преимущества: простота в использовании, доступность, независимость от платформы, поддержка разных форматов. В частности, использовались форматы PINS, PONT и OWL.

При реализации базы знаний по современному искусству использовались две версии Protégé. Protégé 3.2 была применена для создания классов разрабатываемой онтологии и введения отношений между ними, так как интерфейс этой версии более нагляден. Особенно удобным оказался интерфейс введения конкретных экземпляров классов благодаря возможности использования Форм, позволяющих последовательно ввести все данные экземпляра и установить связи с другими экземплярами. При этом проект онтологии был создан в формате PINS и PONT.

Далее разработанный проект был экспортирован в формат OWL, поддерживаемый более поздней версией Protégé 4.1. В этой версии оказалось возможным определить свойства отношений, а также визуализировать разработанную базу знаний

и ее фрагменты. Возможность постановки более сложных запросов также оказалась весьма полезной и расширила возможности разработанной информационно-справочной системы. При этом для тестирования ее работы использовалась встроенная машина обработки знаний FACT++, которая позволила протестировать прототип системы.

Фреймовая модель, на которой основывается функционирование Protégé, является достаточно удобной для представления знаний по современному искусству. В частности, полезным оказался принцип наследования свойств между фреймами. Особенно ярко выражено опирается на фреймовую модель интерфейс Protégé 3.2.

3.2. Формирование классов и их слотов

Выделенные понятия были представлены в Protégé в виде иерархической структуры классов и подклассов. При этом использовалось отношение subClassOff, которое является стандартным аналогом в языке OWL введенному в разделе проектирования онтологии отношению «класс-подкласс». Отношение subClassOff было использовано во избежание дублирования отношений.

Суперклассы, равные пересечению множеств экземпляров своих подклассов, были выделены свойством «Абстрактный», что означает, что для них не могут быть созданы экземпляры. Классы, не имеющие подклассов, а также классы, только частично покрываемые своими подклассами, были выделены как конкретные, то есть способные иметь экземпляры.

Следующие классы приобрели свойство абстрактности:

- Объекты;
- Произведение искусства;
- Художественный объект;
- Персоны;
- Признаки;
- Материал;
- Тип изображения;
- Композиция;
- Институты.

Благодаря этому свойству, заполнение базы знаний является более последовательным. При необходимости дополнить базу знаний третьим лицом, добавление экземпляров интуитивно происходит по принципу, выбранному разработчиком.

Далее для каждого класса (фрейма) был определен и создан список слотов, то есть данных, характеризующих экземпляры этого класса. При этом учитывалось свойство наследия слотов от суперкласса к подклассу. Вводились слоты, описывающие как индивидуальные данные (Data Properties), так связывающие данный экземпляр с другими экземплярами (Object Properties).

Выделенные отношения предметной области

также были представлены в виде слотов, значениями которых были определены экземпляры соответствующих классов. Слоты, описывающие отношения между экземплярами, составляют на языке OWL компоненту Object Properties создаваемой онтологии. При их создании учитывалась их связь с обратными отношениями, а также свойства симметричности, транзитивности, функциональности и рефлексивности. При создании функционального, транзитивного или рефлексивного отношения эти свойства были указаны как тип отношения. Симметричные отношения были представлены как обратные сами себе.

В результате выполненной работы на этом этапе был реализован каркас базы знаний: были созданы иерархия классов и отношения между ними.

3.3. Заполнение базы знаний

При создании экземпляров была использована компонента Формы. Для каждого конкретного класса (то есть класса, способного иметь экземпляры) была создана своя форма, отражающая все слоты в виде ячеек, которые должны быть заполнены. Не все ячейки являлись обязательными для заполнения, однако стояла цель заполнить как можно больше информации о каждом экземпляре. Благодаря наличию большого количества введенных обратных отношений часть ячеек форм заполнялась автоматически.

качестве информации о произведениях искусства. Поэтому при введении экземпляров всех классов кроме произведения искусства учитывался необходимый минимум, который достаточен для описания художественных работ.

В первую очередь были созданы экземпляры ограниченных по объему классов, таких как функция, композиция, жанр, стиль, носитель, тип изображения, техника, роль. На следующей стадии были введены стилистические характеристики и описаны стили методом введения связи между набором стилистических характеристик и самим стилем. Несмотря на то, что стилистические характеристики могут встречаться у нескольких стилей одновременно, каждый набор характеристик уникален (рисунок 1). Совокупность характеристик для каждого стиля была подобрана в результате изучения специализированной литературы [Davies, 2007; Chilvers, 1997; Ruhrberg, 2005; Lucie-Smith, 1988]. При этом была совершена попытка найти в различных научных изданиях по искусствоведению наиболее универсальные характеристики стилей, отмеченные большинством авторов и характерные для всех относящихся к определенному стилю художественных работ.

На следующей стадии были введены художники и их произведения искусства. При создании экземпляра, соответствующего художнику, вводилась базовая биографическая информация (годы жизни, имя, место проживания), ссылки на

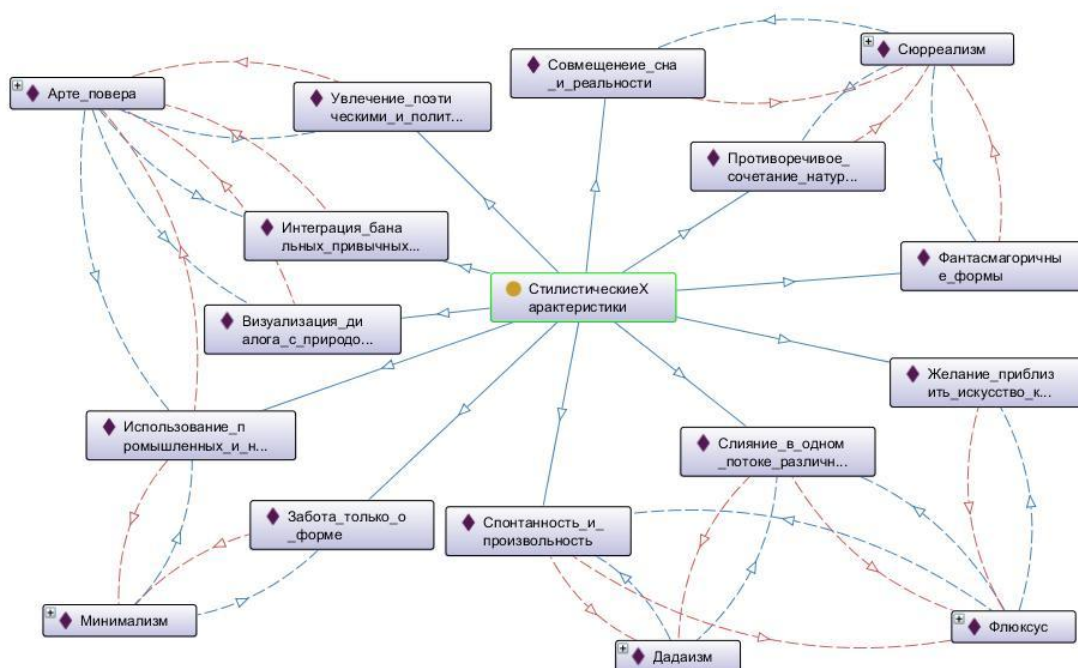


Рисунок 1 – Описание стилей арте повера, сюрреализм, флюксус, дадаизм и минимализм на уровне их стилистических признаков

При создании базы знаний было принято решение фокусироваться на произведениях искусства и их описании, а не на художниках и теории об искусстве. Соответственно все остальные понятия онтологии были необходимы в основном в

произведения искусства, а также проанализировавшие работу художника научных издания. Для того, чтобы процесс протекал последовательно, каждый раз создавался сначала экземпляр художника, затем научных изданий,

проанализировавших его, а затем экземпляры произведений искусства этого художника. Затем совершался переход к следующему художнику. На рисунке 2 приведен пример представления конкретного произведения искусства и структура его отношений с другими классами и экземплярами разработанного прототипа базы знаний.

системы доказал свою работоспособность в результате тестирования, однако для повышения его статуса до полноценной информационной системы необходима дальнейшая проработка знаний по предметной области в глубину, а также введение максимально возможного количества информации. Общая структура разработанной онтологии

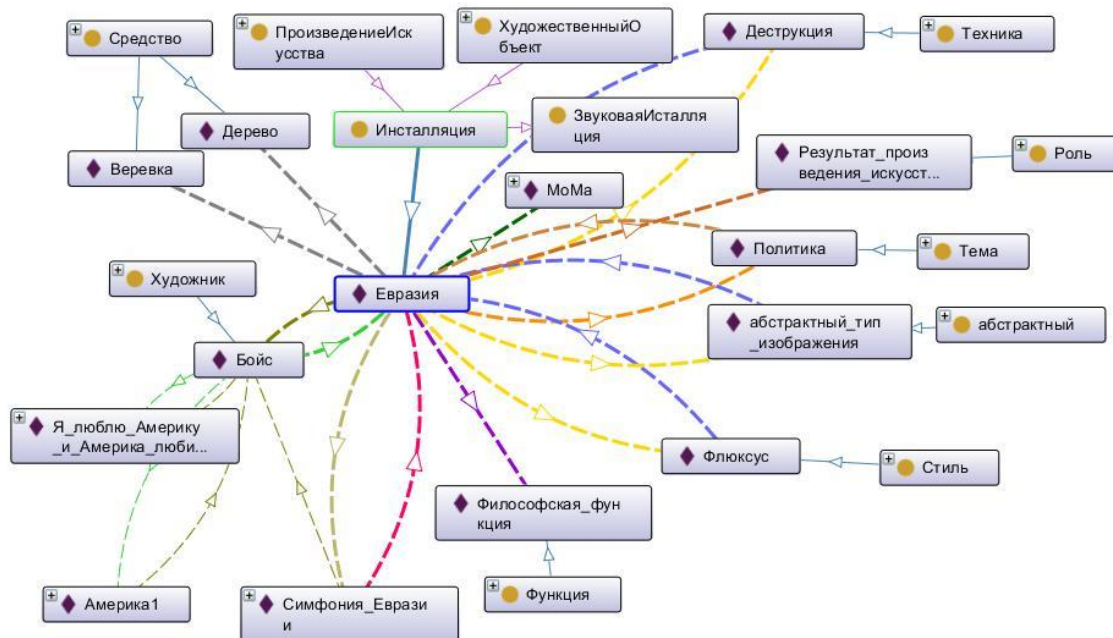


Рисунок 2 – Структура отношений между художественным объектом «Евразия», другими экземплярами и классами

ЗАКЛЮЧЕНИЕ

В данной работе приведено описание процесса создания прототипа интеллектуальной информационной справочной системы по современному искусству. В результате анализа особенностей существующих моделей представления знаний и наиболее подходящих технологий инженерии знаний, таких как Protégé, OntoEdit и OSTIS, было замечено, что по общим описаниям для данного проекта подходит как Protégé, так и технология OSTIS, так как они основаны на фреймовой и семантической моделях представления знаний, обладают необходимым функционалом, а также являются свободно доступными (в отличие от OntoEdit). Описанный прототип системы был реализован при помощи Protégé.

Как было указано выше, преимуществами платформы Protégé оказались простота в использовании, доступность, независимость от платформы, поддержка разных форматов. Функциональность Protégé позволила реализовать все задуманные аспекты прототипа. Однако, использованная встроенная машина обработки знаний FACT++ оказалась слабой для полноценной интеллектуальной информационной системы.

Разработанный прототип информационной

позволяет дальнейшее нюансирование понятий без нанесения серьезных изменений в уже существующую концептуализацию.

Данный проект осуществлен в рамках более крупного исследования «Модели и средства построения интеллектуальных справочных систем по изобразительному искусству», целью которого является компаративный анализ наиболее подходящих технологий для реализации интеллектуальных информационных систем по гуманитарным наукам и, в частности, по искусству. Этот проект предполагает также создание прототипа интеллектуальной справочной системы по искусству с использованием технологии OSTIS для последующего сравнительного анализа.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Голенков, 2011] Голенков, В. В. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н. А. Гулякина //Материалы международной научно-технической конференции OSTIS-2011, Мн.: БГУИР, 2011. - 21-58 стр.
- [Овдей, 2004] Овдей, О. М. Обзор инструментов инженерии онтологий / О. М. Овдей, Г. Ю. Проскудина // Электронные библиотеки. – 2004. – Том 7, выпуск 4. – С. 46-65.
- [Фоминих, 2002] Фоминих, И. Б. О технологии решения творческих задач// Сборник трудов 8: Национальная конференция по искусственному интеллекту “КИИ-2002”, том 2. - М.:изд.физ.-матлит., 2002.

[Шункевич, 2013] Шункевич, Д. В. Семантические модели и средства проектирования машин обработки знаний / Д. В. Шункевич // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2013.

[Chaudri, 1998] Chaudri, V. OKBC: A Programmatic Foundation for Knowledge Base Interoperability / V. Chaudhri, A. Farquhar, R. Fikes P. Karp J. Rice // Fifteenth National Conf. on Artificial Intelligence. – AAAIPres/The MIT Press, Madison, 1998. – P.600-607.

[Chilvers, 1997] Chilvers, I. The Oxford Dictionary of Art. New edition/ I. Chilvers, H. Osborne. – Oxford University Press, Oxford, 1997 – 647 p.

[Golitsyn, 1995] Golitsyn, G. A. Information and Creation/ G. A. Golitsyn, V. M. Petrov. – Birkhauser Verlag, Basel, 1995.

[Davies, 2007] Davies, P. J. A. Janson's History of Art. Seventh Edition/ P. J. A. Davies, W. B. Denny, F. F. Hofrichter, J. Jacobs, A. M. Roberts. – Pearson Education, Inc., UpperSaddle River, NJ, 2007. – 1112 p.

[Lucie-Smith, 1988] Lucie-Smith, E. Moderne Kunst. Van abstract expressionisme tot hyperrealisme/ E. Lucie-Smith. – Elsevier, Amsterdam/Brussel, 1988. – 516 p.

[Musen, 1998] Musen, M. Domain Ontologies in Software Engineering: Use of Protege with the EON Architecture // Methods of Inform. in Medicine. – 1998. – P. 540-550.

[Noy, 2001] Noy, N. Creating Semantic Web Contents with Protege-2000 / N. Noy, M. Sintek, S. Decker, M. Crubezy, R. Fergerson, M. Musen // IEEE Intelligent Systems. – 2001. – March/April. – P. 60-71.

[OSTIS, 2012] Проект OSTIS [Электронный ресурс]. Минск, 2012. – Режим доступа: <http://ostis.net/>. – Дата доступа: 13.12.2012.

[Ruhrberg, 2005] Ruhrberg, K. Kunst van de 20e eeuw. Schilderkunst – Beeldhouwkunst – Nieuwe media – Fotografie/ K. Ruhrberg, M. Schneckenburger, C. Fricke, K. Honnef. – Taschen, Köln, 2005. – 840 p.

[Sure, 2002] Sure, Y. OntoEdit: Collaborative ontology development for the Semantic Web / Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke // In Proc. of the Inter. Semantic Web Conference (ISWC 2002). – Sardinia, Italia, June 2002. – P. 158-215.

INTELLIGENT INFORMATION SYSTEM ON MODERN ART

Leichanka N.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

nastassia.leichanka@gmail.com

The work provides a description of the development process of a prototype of an intelligent information system on modern art, which starts with an analysis of the relevance of the posed problem and the choice of the most suitable technologies. Further a description of the designed ontology and its implementation by means of Protégé are provided. The work will be concluded with an analysis of the obtained results

INTRODUCTION

Nowadays utilization of artificial intelligence technologies to improve the quality and the velocity of information retrieval gains popularity. Recently a great amount of domain-specific intelligent information systems has been developed, as well as generic systems (for instance CYC), but the quality of knowledge acquisition in the latter is much lower than in domain-specific systems. In addition, it was noticed that systems representing technical knowledge are more

common. Humanitarian knowledge is seldom represented.

The work describes the whole process of developing of a prototype of an intelligent information system on modern art, including the analysis of tools and technologies, ontology design and its implementation.

MAIN PART

The analysis of the various knowledge representation models showed that the frame and semantic models can be both applied to develop an intelligent system based on humanitarian knowledge. Among the existing technological tools for constructing a knowledge-based system the following can be put forward: Protégé, OntoEdit and OSTIS. All of these tools can be used to develop a knowledge based system, but the disadvantage of OntoEdit for this project lies in the fact that it is licensed (thus paid). Protégé combines frame and semantic knowledge representation models and supports OWL-format, whereas OSTIS is based on the semantic model of knowledge representation and is an open technology for component-based design of various intelligent systems.

During the process of ontology design domain knowledge was thoroughly analyzed. Thus the most significant concepts and their characteristics have been specified, and the most important relationships between these concepts and their properties were defined. In this way, four main abstract classes were distinguished – *People, Objects, Characteristics and Institutes*. All other concepts fit hierarchically in one of these main classes. Furthermore, classes of possible queries were determined.

The implementation of the designed prototype was realized by means of the platform Protégé. Two versions of the platform – Protégé 3.2 and 4.1 – and such file formats as PINS, PONT and OWL were used. Initially, the framework of the ontology (concepts, their slots and relationships) was encoded. Then the effort was made to fill the knowledge base on modern art as well as possible with domain-specific instances, using forms created specifically for each concept. The embedded in Protégé reasoner FACT++ proved the efficiency of the constructed system prototype.

CONCLUSION

The work describes all steps of creating an intelligent information system on modern art. The analysis of suitable technologies showed that Protégé and OSTIS can provide the best results. The designed ontology was built in the framework Protégé. The functionality of the framework allowed realization of all aspects of the conceived prototype.

The given work fits in an overarching project which aims a scientific comparative analysis of suitable technologies for building an intelligent system based on humanitarian knowledge, and which assumes a development of a similar prototype by means of technology OSTIS for further comparison.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8:621.031

ОРГАНИЗАЦИЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ОЦЕНКИ ТЕХНИЧЕСКОГО СОСТОЯНИЯ СЛОЖНЫХ МЕХАНИЗМОВ

Бранцевич П.Ю.* , Ли И.**

* *Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

branc@bsuir.edu.by

** *Шаньдунский университет
г. Цзинань, Китайская народная республика*

liyibing@sdu.edu.cn

Отмечается важность организации технического обслуживания оборудования в процессе эксплуатации в соответствии с реальным техническим состоянием. Это может быть достигнуто путем внедрения современных систем мониторинга, оценки технического состояния, диагностики. Рассмотрены основные составные части системы оценки состояния сложных механизмов с вращательным движением по параметрам вибрации.

Ключевые слова: вибрация, диагностика, решение, система.

ВВЕДЕНИЕ

В общей сумме затрат расходы на эксплуатацию производственного оборудования достигают значительных величин, причем существенную долю в них занимает ремонтно-эксплуатационная составляющая.

Считается, что наиболее изнашиваемым является оборудование с вращательным движением (турбины, генераторы, двигатели, редукторы, насосы, компрессоры, вентиляторы и т.д.). Снизить затраты на его эксплуатацию можно путем внедрения современных систем технического обслуживания, которые базируются на использовании технологий мониторинга, оценки состояния, диагностики, прогнозирования развития дефектов, которые с точки зрения их организации и функционирования являются интеллектуальными системами.

Состояние производственного оборудования может характеризоваться многими параметрами основных и вторичных процессов, развивающихся при его работе. Для контроля целесообразно выбирать те, которые достаточно хорошо отражают функциональное состояние объектов и не требуют слишком больших затрат на их измерение. В этом плане, для механизмов с вращательным движением, такими являются параметры вибрации [1].

На основе анализа вибрационного состояния группы однотипных механизмов при их функционировании на различных режимах, в

различном техническом состоянии и на протяжении длительного времени могут быть обоснованы и сформулированы диагностические признаки для локализации мест и причин повышения вибрации. Это создает условия для построения автоматизированных интеллектуальных систем оценки технического состояния и диагностики, значительно облегчающих работу инженерно-технического персонала [2].

1. ОРГАНИЗАЦИЯ СИСТЕМЫ ОЦЕНКИ ТЕХНИЧЕСКОГО СОСТОЯНИЯ МЕХАНИЗМОВ

Практически невозможно разработать подобную универсальную систему, так как каждый тип оборудования имеет свои специфические особенности и характеризуется определенным набором параметров. В связи с этим, представляется целесообразным создание обобщенной модели системы оценки технического состояния и диагностирования, которая адаптируется под конкретное применение, и может быть представлена следующим набором взаимодействующих элементов.

1. Множество информативно значимых параметров, характеризующих техническое состояние объекта

$$P = \{p_1, p_2, p_3, \dots, p_N\}. \quad (1)$$

Например, амплитуды спектральных составляющих вибросигнала на фиксированных

частотах, мощность или СКЗ вибрации в полосе частот (может задаваться несколько полос), пик-фактор, эксцесс, асимптота, коэффициент модуляции огибающей вибросигнала, амплитуды оборотных составляющих вибрации и т.д. Параметры выбираются на основе теоретических и экспериментальных исследований, экспертных знаний, эвристических предположений.

2. Множество дефектов, требующих обнаружения, для конкретного механизма

$$D = \{d_1, d_2, d_3, \dots, d_M\}. \quad (2)$$

Конкретный состав этого множества определяется типом механизма, его техническими параметрами, выполняемыми функциями, условиями эксплуатации и т. п.

Например, дефектами подшипников качения являются: неуравновешенность ротора, овальность внутреннего кольца, раковины на поверхности тел качения, перекос наружного кольца, раковины и трещины на наружном или внутреннем кольце, дефекты смазки.

3. Базовые значения каждого из параметров множества P , которые соответствуют нормальному (исправному, бездефектному) состоянию исследуемого объекта

$$Z = \{z_1, z_2, z_3, \dots, z_N\}. \quad (3)$$

Эти значения принимаются: на базе нормативных документов, регламентирующих допустимые уровни вибрации механизмов; на основе исследования вибрационных характеристик достаточно большого количества однотипных механизмов на протяжении продолжительного периода времени; путем теоретических расчетов; в результате проведения натурного или полунатурного моделирования.

4. Подмножества информативно значимых параметров, по изменению которых можно идентифицировать проявление каждого из дефектов, и весовые коэффициенты значимости для каждого признака (в качестве примера см. таблицу 1). Весовые коэффициенты принимают значения в диапазоне от 0 до 1.

Таблица 1 – Соответствие информативно значимых признаков дефектам

Дефект	Информативные признаки	Весовые коэффициенты
d_1	p_i, p_j, p_k	$\alpha_{d_1,i}, \alpha_{d_1,j}, \alpha_{d_1,k}$
...
d_r	p_1, p_3	$\alpha_{d_r,1}, \alpha_{d_r,3}$
...
d_M	p_q	$\alpha_{d_M,q}$

В данной таблице индексы i, j, k, q могут принимать значения от 1 до N .

5. Множество решающих функции для каждого

дефекта, отражающих проявление или развитие дефекта. Решающие функции учитывают соотношение текущих и базовых значений информативно значимых параметров и могут быть достаточно разнообразными, например численными, логическими, на основе нечетких правил, претендентов.

Можно привести некоторые примеры таких функций.

При применении численных решающих функции на первом этапе выбирается формула, описывающая эту функцию. Такие формулы, к примеру, могут иметь вид:

$$R_{d_j} = \frac{1}{\sum_{i \in S_{d_j}} \alpha_{d_j,i}} \cdot \sum_{i \in S_{d_j}} \left(\frac{x_i - z_i}{z_i} \right) \cdot \alpha_{d_j,i}; \quad (4)$$

$$R_{d_j} = \frac{1}{\sum_{i \in S_{d_j}} \alpha_{d_j,i}} \cdot \sum_{i \in S_{d_j}} (x_i - z_i) \cdot \alpha_{d_j,i}; \quad (5)$$

$$R_{d_j} = \sum_{i \in S_{d_j}} (x_i - z_i) \cdot \alpha_{d_j,i}; \quad (6)$$

где S_{d_j} - множество номеров информативных признаков, присущих дефекту d_j ;

x_i - полученное при обследовании (текущее) значение для i -го информативного признака.

На втором этапе для конкретной численной решающей функции определяются граничные значения, соответствующие степени развития дефекта, которая может быть, например, сформулирована как очень малая (ОМ), малая (М), средняя (С), большая (Б) –

$$R_{d_j, \text{Ст.разв.деф.}} = \{R_{d_j, \text{ОМ}}, R_{d_j, \text{М}}, R_{d_j, \text{С}}, R_{d_j, \text{Б}}\}. \quad (7)$$

Сравнение, вычисленного для конкретных данных, значения решающей функции с граничными значениями из множества $R_{d_j, \text{Ст.разв.деф.}}$, позволяет установить степень развития конкретного дефекта. Целесообразно определить функцию развития дефекта Ψ_{d_j} , которая может принимать одно из значений набора {очень малая (ОМ), малая (М), средняя (С), большая (Б), очень большая (ОБ)}:

$$\Psi_{d_j,i} = \begin{cases} \text{ОМ, если } R_{d_j} < R_{d_j, \text{ОМ}} \\ \text{М, если } R_{d_j, \text{ОМ}} \leq R_{d_j} < R_{d_j, \text{М}} \\ \text{С, если } R_{d_j, \text{М}} \leq R_{d_j} < R_{d_j, \text{С}} \\ \text{Б, если } R_{d_j, \text{С}} \leq R_{d_j} < R_{d_j, \text{Б}} \\ \text{ОБ, если } R_{d_j, \text{Б}} \leq R_{d_j} \end{cases} \quad (8)$$

При построении логических решающих функций применяется следующий подход:

а) для количественных значений каждого из

информативных признаков для каждого j -го типа дефекта устанавливаются границы, соответствующие различной степени развития дефекта (например, очень малая (исправное состояние), малая, средняя, большая) - $z_{d_j,i,OM}$, $z_{d_j,i,M}$, $z_{d_j,i,C}$, $z_{d_j,i,B}$;

б) по полученному при обследовании (текущему) значению для i -го информативного признака j -го типа дефекта определяется степень развития дефекта (значение функции развития дефекта по i -му информативному признаку $F_{d_j,i}$, которая принимает одно из значений набора {очень малая (ОМ), малая (М), средняя (С), большая (Б), очень большая (ОБ)}):

$$F_{d_j,i} = \begin{cases} \text{ОМ, если } x_i < z_{d_j,i,OM} \\ \text{М, если } z_{d_j,i,OM} \leq x_i < z_{d_j,i,M} \\ \text{С, если } z_{d_j,i,M} \leq x_i < z_{d_j,i,C} \\ \text{Б, если } z_{d_j,i,C} \leq x_i < z_{d_j,i,B} \\ \text{ОБ, если } z_{d_j,i,B} \leq x_i \end{cases}; \quad (9)$$

в) для j -го типа дефекта определяется степень его развития (значение функции развития j -го типа дефекта Ψ_{d_j} , которая принимает одно из значений набора {очень малая (ОМ), малая (М), средняя (С), большая (Б), очень большая (ОБ)} и является логической функцией от функций $F_{d_j,i}$, где $i \in S_{d_j}$:

$$\Psi_{d_j} = \Phi(F_{d_j,m}, F_{d_j,n}, F_{d_j,l}, \dots), \text{ где } m, n, l, \dots \in S_{d_j}. \quad (10)$$

Такая функция может быть задана в табличной форме (см. таблицу 2).

Таблица 2 – Пример задания логической функции развития дефекта для двух диагностических признаков

$F_{d_j,i}$	$F_{d_j,j}$	Ψ_{d_j}
ОМ	ОМ	ОМ
ОМ	ОМ∨М	М
С	ОМ∨М∨С	С
ОМ∨М	Б	С
С	Б	Б
Б	Б	Б
М∨С∨Б∨ОБ	ОБ	ОБ
ОБ	М∨С∨Б∨ОБ	ОБ

Решающие функции могут быть построены и на основе нечетких правил. Вывод о том в каком состоянии находится объект в зависимости от значения характеризующего его параметра x формируется на основе значений функций принадлежности:

$$FP_{1\text{Сост.}}(x) = \exp\left(-\frac{(x-x_{\min})^2}{2(X_1-x_{\min})^2}\right), \quad x > x_{\min}; \quad (11)$$

$$FP_{j\text{Сост.}}(x) = \begin{cases} \exp\left(-\frac{(x-X_{j-1})^2}{2(X_{j-1}-X_{j-2})^2}\right), & x \leq X_{j-1} \\ \exp\left(-\frac{(x-X_{j-1})^2}{2(X_j-X_{j-1})^2}\right), & x > X_{j-1} \end{cases}; \quad (12)$$

$$FP_{5\text{Сост.}}(x) = \begin{cases} \exp\left(-\frac{(x-X_4)^2}{2(X_4-X_3)^2}\right), & x \leq X_4 \\ 1, & x > X_4 \end{cases}; \quad (13)$$

где x - параметр, для которого определяется значение функции принадлежности; $j=2,3,4$.

X_j - значения параметра x , которые соответствуют определенным состояниям, причем $x_{\min} < X_1 < X_2 < X_3 < X_4$. Их количественные значения выбираются на основе экспертных оценок и накапливаемых экспериментальных данных.

Решение о том, что объект находится в состоянии j принимается в том случае, если $FP_{j\text{Сост.}}(x)$ имеет максимальное значение из всех вычисленных $FP_{k\text{Сост.}}(x)$.

Если решение о состоянии объекта принимается на основании нескольких параметров, то в этом случае обобщенная функция принадлежности для J -го состояния имеет вид:

$$FP_{J\text{Сост.}} = \sum_{i=1}^N \alpha_i \cdot FP_{J\text{Сост.}}(x_i), \quad (14)$$

где $FP_{J\text{Сост.}}(x_i)$ - значение функции принадлежности к J -му состоянию по параметру x_i ; α_i - весовой коэффициент значимости параметра x_i ; N - число параметров, значение которых учитывается при принятии решения.

Вывод о том, что объект находится в состоянии J , делается в том случае, если $FP_{J\text{Сост.}}$ имеет максимальное значение из функций принадлежности, вычисленных для всех возможных состояний объекта. Если окажется, что для нескольких состояний функции принадлежности $FP_{J\text{Сост.}}$ имеют одинаковое значение, то отнесение объекта к определенному состоянию осуществляется на основании функции принадлежности для параметра, который имеет наибольший весовой коэффициент. В такой ситуации, когда все параметры равнозначны, следует руководствоваться дополнительными

правилами по отнесению объекта к определенному состоянию в зависимости от целевых и стоимостных критериев.

6. Множество рекомендаций по устранению обнаруженного дефекта и предотвращению его развития. Для каждого дефекта формулируется некоторое множество рекомендаций по его ликвидации или снижению его влияния:

$$C_{d_j} = \{c_{d_j,1}, c_{d_j,2}, c_{d_j,3}, \dots, c_{d_j,K}\}. \quad (15)$$

Примеры таких рекомендаций: заменить смазку в подшипнике; провести балансировку вала; заменить подшипник; проводить ежедневное измерение параметров вибрации механизма и т.п.

Результатом работы системы вибрационной диагностики является набор рекомендаций для пользователя по устранению выявленного дефекта. Выбор таких рекомендаций есть некоторый функционал φ от функции развития дефекта Ψ_{d_j} и множества рекомендаций по его ликвидации C_{d_j} , результатом которого есть некоторое подмножество множества C_{d_j} в пределе с ним совпадающее:

$$E_{d_j} = \varphi(\Psi_{d_j}, C_{d_j}), \quad E_{d_j} \in C_{d_j}. \quad (16)$$

ЗАКЛЮЧЕНИЕ

Применение систем вибрационного мониторинга позволяет получать данные об изменении технического состояния механизмов и агрегатов, планировать мероприятия по техническому обслуживанию и ремонтам оборудования и, таким образом, оптимизировать затраты на их проведение. Стационарные системы мониторинга помимо этого ориентированы на предотвращение серьезных аварий.

Использование компьютера в качестве базового узла системы обработки вибрационных данных обеспечивает быструю настройку под условия применения, гибкость и расширяемость функциональных возможностей.

Задачами вибрационной диагностики является обнаружение зарождающихся дефектов, наблюдение за их развитием и прогнозирование изменения технического состояния. Однако не следует ожидать очень быстрого экономического эффекта от их внедрения, так как требуется время на адаптацию системы под конкретное производство, а также подготовку специалистов и изменение психологии их поведения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Барков, 2000] Барков, А.В. Мониторинг и диагностика роторных машин по вибрации / А.В. Барков, Н.А. Баркова, А.Ю. Азовцев. – СПб. : Изд. центр СПбГМТУ, 2000. – 169 с.

[Бранцевич, 2008] Бранцевич, П.Ю. ИВК «Лукомль-2001» для вибрационного контроля / П.Ю. Бранцевич // Энергетика и ТЭК. – 2008. – № 12(69). – с.19-21.

ORGANIZATION OF INTELLECTUAL SYSTEM OF THE ASSESSMENT TECHNICAL CONDITIONS COMPOSITE MECHANISMS

Brancevich P.J. *, Li.Y. **

*Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

branc@bsuir.edu.by

**Shandong university, Jinan, P.R.China

liyibing@sdu.edu.cn

This article notes the importance of technical equipment maintenance while their operation, according to its actual technical state. It can be achieved by implementation of modern monitoring and technical condition diagnostics systems. The article covers main components of evaluation state system for complex mechanisms with rotary motion, basing on vibration parameters.

INTRODUCTION

On vibration state analysis basis for similar mechanisms functioning in different modes and technical conditions in long terms, we can justify and formulate diagnostic criteria's for further localization of vibration sources. This gives ability to construct automatic intelligent technical condition assessment systems, which facilitate engineering and technical personnel work.

MAIN PART

Each type of equipment has its own specific characteristics and can be described with a certain set of parameters. In this regard, it is appropriate to create a generalized model of the technical condition assessment and diagnosis system, which can be customized for specific applications. Such system consists of the next elements, which interact with each other.

Informative parameters set, which characterize the object technical state. Defects set, which need to be detected. Informative parameters initial values. Changes in subset of informative parameters, which indicate appearance of corresponding defects. Set of decision functions for each defect type.

Result of vibration diagnostics system work is a set of recommendations for user how to resolve identified defects.

CONCLUSION

The objectives of vibration diagnostics are: incipient defects detection, defect progress monitoring and predicting technical condition changes. However, we can't expect a very rapid economic effect from their implementation, as it takes time to adapt the system to a specific production, as well as training and change in the psychology of behavior.



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 519.7

ЗАДАЧИ НА ГРАФАХ В ТЕОРИИ ПРОЕКТИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ

Поттосин Ю.В. *, Поттосина С.А. **

* *Объединенный институт проблем информатики НАН Беларуси,
г. Минск, Республика Беларусь*
pott@newman.bas-net.by

** *Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*
pottosina@sum-solution.net

Рассматриваются задачи на графах, имеющие приложение в проектировании дискретных устройств управления. Дается обзор методов и алгоритмов решения этих задач. Приводятся примеры использования графовых моделей для решения задач логического проектирования.

Ключевые слова: логическое проектирование; раскраска графа; полный подграф; двудольный подграф.

ВВЕДЕНИЕ

Объектно-ориентированный подход к решению проблем логического проектирования дискретных устройств предполагает построение базы знаний. База знаний содержит структурированную информацию и правила вывода, которые покрывают библиотечными элементами данную область знаний. При логическом проектировании дискретных устройств такими элементами могут служить логико-комбинаторные задачи и алгоритмы их решений, сформулированные в терминах теории графов.

Методы теории графов широко применяются в логическом проектировании дискретных управляющих и вычислительных устройств. О приложениях теории графов в логическом проектировании и близких к нему областях написано довольно много работ, перечислить которые было бы сложно, поэтому ограничимся только указанием некоторых монографий [Мелихов, 1971], [Мелихов и др., 1974], [Нечепуренко и др., 1990] и обзорных статей [Закревский и др., 1986], [Поттосин, 2001].

На этапе логического проектирования дискретных устройств управления возникают разнообразные логико-комбинаторные задачи [Агибалов, 1981], [Закревский и др., 2007], [Поттосин, 2011], решение которых приводит к оптимальным вариантам логического проекта. В

настоящей работе предлагается краткий обзор тех задач и методов теории графов, которые находят широкое применение в логическом проектировании.

В рассматриваемых задачах понятие графа используется в разных интерпретациях. Чаще всего это графическое представление бинарного отношения типа совместимости, т.е. отношения, обладающего свойствами рефлексивности и симметричности. Такое отношение представляется произвольным неориентированным графом. В некоторых случаях граф представляет отношение квазипорядка или отношение следования между некоторыми событиями. В отдельных задачах граф представляет логическую сеть. Рассматриваемые в этих случаях графы являются ориентированными.

1. Раскраска графа

Значительное место в логическом проектировании занимает задача раскраски графа. Раскраска графа может быть использована при решении задачи декомпозиции булевых функций [Perkowski, 1995], [Закревский, 2000], [Бибило, 1987], занимающей важное место в синтезе комбинационных схем в базе сверхбольших интегральных схем (СБИС). Из всех задач логического проектирования, пожалуй, именно она наиболее часто сводится к раскраске графа.

Одна из постановок задачи декомпозиции формулируется следующим образом. Задана система булевых функций как векторная булева

функция $y = f(x)$, и заданы векторные булевы переменные w и z , составленные из компонент векторной переменной x . Необходимо представить заданную систему в виде $f(x) = h(w, g(z))$. Функции h и g должны быть более простыми, чем f . В данном случае это означает, что число s компонент вектора z и сумма чисел компонент векторов w и $u = g(z)$ меньше n . Обычно стремятся получить минимум этой суммы.

Общий метод решения данной задачи предполагает построение графа G , вершинами которого являются значения z^* векторной переменной z , а ребра определяются следующим образом. Пусть $x^*_{i_1}$ – значение векторной переменной x , а $z^*_{i_1}$ и $w^*_{i_1}$ – значения векторных переменных z и w соответственно, причем компоненты x^* совпадают с соответствующими компонентами $z^*_{i_1}$ и $w^*_{i_1}$. Вершины $z^*_{i_1}$ и $z^*_{i_2}$ графа G связаны ребром, если найдется значение w^* вектора w такое, что значения функции f при соответствующих $x^*_{i_1}$ и $x^*_{i_2}$ различны.

Легко доказывается, что заданную векторную функцию можно представить в виде $f(x) = h(w, u)$, где векторный аргумент u имеет k компонент, тогда и только тогда, когда хроматическое число графа G не превосходит 2^k .

Для получения окончательного решения достаточно получить минимальную раскраску графа G и закодировать полученные цвета булевыми векторами, которые представят значения функции g при соответствующих значениях вектора z .

Ввиду того, что задача раскраски графа является NP-трудной и точное ее решение не всегда практически достижимо, в работе [Закревский и др., 2007] предлагается эвристический метод, который позволяет получать раскраску, довольно близкую к минимальной за практически приемлемое время. Алгоритм, реализующий данный метод, позволяет в отдельных случаях по характеру его выполнения показывать, что полученное решение является точным.

2. Кодирование графа. Двудольные подграфы

Как было сказано выше, при декомпозиции булевых функций в конечном счете требуется закодировать булевыми или троичными векторами цвета, полученные после раскраски графа. Данные векторы считаются значениями векторной переменной u , обладающей тем свойством, что каждому ребру графа должна соответствовать ее компонента, принимающая противоположные значения (0 и 1) на концах этого ребра. Таким образом, каждой компоненте вектора u можно поставить в соответствие двудольный подграф рассматриваемого графа, а данную задачу можно рассматривать как задачу покрытия графа двудольными подграфами. Такой подход к решению задачи декомпозиции булевых функций описан в работах [Закревский, 2000], [Бибило, 1987].

Задача покрытия графа двудольными подграфами возникает при поиске оптимального кодирования состояний синхронного автомата, описанном в работе [Закревский и др., 2007]. Предполагается задание входа и выхода автомата в структурном алфавите. Такая модель названа в работе [Закревский, 1981] автоматом с абстрактным состоянием. Автомат представляется в виде двух троичных матриц U и V . Столбцам матрицы U соответствуют входные булевы переменные x_1, x_2, \dots, x_n , столбцам матрицы V – выходные булевы переменные y_1, y_2, \dots, y_m . Строки обеих матриц помечены состояниями автомата так, что пара их одноименных строк представляет переход из состояния, являющегося меткой строки матрицы U , в другое состояние, являющееся меткой строки матрицы V . Строка матрицы U представляет при этом условие перехода, а строка матрицы V – двоичные выходные сигналы, сопровождающие этот переход.

Метод заключается в следующем. Для некоторого столбца u_i матрицы V строится граф G , вершинам которого соответствуют состояния заданного автомата. В данном столбце u_i отыскиваются пары элементов с противоположными значениями (0 и 1), которым соответствуют пары неортогональных строк матрицы U . Вершины графа G , соответствующие меткам строк всякой такой пары, связываются ребром и затем кодируются, как показано выше. Полученные коды переносятся в матрицы U и V в виде значений вновь введенных внутренних переменных z_1, z_2, \dots, z_k . Эта процедура повторяется для каждого столбца матрицы V , причем каждый раз выбирается столбец с минимальным числом пар указанных элементов. Когда граф G для каждого столбца оказывается пустым (после введения новых переменных строки становятся ортогональными), процесс заканчивается, а преобразованные матрицы U и V представляют систему булевых функций, описывающую комбинационную часть логической сети, реализующей заданный автомат. Некоторые совместимые состояния оказываются закодированными неортогональными или даже одинаковыми кодами. В этом случае фактическое число состояний автомата оказывается уменьшенным.

В работе [Поттосин, 1985] описан метод выделения в заданном графе G основного двудольного подграфа (т.е. такого подграфа, множество вершин которого совпадает с множеством вершин заданного графа) с множествами вершин V^0 и V^1 и максимальным числом ребер. Алгоритм, основанный на данном методе, состоит из двух этапов. На первом этапе находятся все циклы нечетной длины, на втором этапе решается задача покрытия множества полученных циклов ребрами графа. Ребра, составляющие найденное покрытие, затем удаляются, в результате чего получается искомым подграф. В зависимости от конкретной реализации

данного метода решение может быть как точным, так и приближенным.

Эвристический метод решения этой задачи предложен в работе [Бибило, 1987] и уточнен в статье [Закревский, 2000]. Согласно этому методу начальное значение V^0 представляет одноэлементное множество, а V^1 содержит все остальные вершины. Затем вершины из множества V^1 последовательно переносятся в множество V^0 до тех пор, пока не перестанет увеличиваться число ребер, связывающих вершины из V^0 с вершинами из V^1 .

3. Независимые множества. Полные подграфы

Задачи логического проектирования, формулируемые как задачи разбиения заданного множества на совместимые в некотором смысле подмножества, удается свести к раскраске графа, если попарной совместимости элементов подмножества достаточно для их групповой совместимости. В противном случае приходится вводить какие-то дополнительные условия одноцветности вершин. Другой подход к решению таких задач предполагает нахождение в заданном графе наибольшего независимого множества, т.е. множества попарно несмежных вершин, а затем группирование около них остальных вершин заданного графа.

С помощью этого подхода можно решать такие задачи, как минимизация длины кода состояний асинхронного автомата, сжатие таблицы переходов автомата (по строкам и столбцам), параллельная декомпозиция автомата, упрощение системы булевых функций, описывающих асинхронный автомат. Этот прием особенно подходит для решения таких задач абстрактного синтеза автомата, как минимизация числа состояний синхронного автомата и его декомпозиция. В этих задачах искомая совокупность совместимых подмножеств должна обладать свойством замкнутости, которое заключается в том, что наличие каких-либо элементов в одном совместимом подмножестве требует включения некоторых других элементов в другое совместимое подмножество из получаемой совокупности. Двойственной задачей по отношению к задаче нахождения независимых множеств является задача нахождения полных подграфов, поскольку полный подграф некоторого графа порождается независимым множеством его дополнения.

Для получения всех максимальных независимых множеств можно с успехом применять классический метод Полла-Ангера построения всех максимальных совместимых множеств состояний автомата. Алгоритм, основанный на этом методе, описан в терминах теории графов в работе [Закревский и др., 2007]. В нем используется прием, при котором заданный граф разлагается на последовательность подграфов и решение данной

задачи для предыдущего подграфа преобразуется в решение для последующего подграфа. В этих же работах описан алгоритм получения всех максимальных независимых множеств, использующий лексикографический перебор.

4. Кодирование состояний автомата. Полный булев граф

В теории булевых функций используется представление булева пространства в виде графа, в котором вершинами являются всевозможные булевы вектора заданной размерности (элементы булева пространства) и две вершины связаны ребром тогда и только тогда, когда соответствующие векторы отличаются друг от друга значением только одной компоненты. В теории графов такой граф называется полным булевым графом и обозначается символом Q_k , где k – размерность соответствующего булева пространства.

Данная модель используется для решения задачи кодирования внутренних состояний дискретного автомата. Это необходимо для перехода от поведенческой модели автомата к его структурной модели. То есть каждое состояние автомата, представленное абстрактным символом в поведенческой модели, должно быть заменено булевым вектором, который, в свою очередь, представляет набор состояний двоичных элементов памяти в логической схеме, реализующей данный автомат. Критерием оптимизации при кодировании состояний автомата служит площадь кристалла, на котором размещается схема, или оценка величины энергии, которую может потреблять проектируемое устройство.

В работе [Закревский, 2005] процесс кодирования состояний автомата представляется как размещение состояний в булевом пространстве. Этот же процесс описан в работе [Закревский и др., 2007] как построение полного булева графа Q_k , напоминающее сборку некоторой простой механической конструкции.

Пусть задан автомат, множество состояний которого $Q = \{q_1, q_2, \dots, q_r\}$. Вершины графа Q_k , являющиеся первоначально вершинами некоторого пустого графа (без ребер), заранее поставлены в соответствие состояниям автомата. На парах этих вершин $\langle q_i, q_j \rangle$ задана функция целочисленная w_{ij} , которая определяется в зависимости от применяемого критерия оптимизации. Если надо получить минимум площади кристалла, то функция w_{ij} определяется как в работе [Armstrong, 1962], и состояния в графе Q_k должны располагаться тем ближе, чем больше величина w_{ij} . Если надо получить минимум потребляемой энергии, то значение функции w_{ij} должно быть связано с частотой переходов между состояниями q_i и q_j .

Построение графа Q_k , который в теории булевых функций называется k -мерным гиперкубом, представляется как последовательность k шагов. На

p -м шаге рассматривается множество $(p - 1)$ -мерных гиперкубов (графов Q_{p-1}), они объединяются в пары, и из каждой пары получается один p -мерный гиперкуб (граф Q_p) путем соответствующего добавления ребер. При этом по возможности для соединения ребрами выбираются те пары вершин, которым соответствуют наибольшие значения величины w_{ij} . Вершинам полученного графа Q_k приписываются k -компонентные булевы векторы с соблюдением отношения соседства, представленного ребрами графа Q_k .

5. Задачи на графах в синтезе структур СБИС

Довольно значительное многообразие задач на графах встречается при оптимизации структур СБИС. Задачи оптимизации структур СБИС лежат на стыке этапов логического и технического проектирования. При оптимизации структуры программируемой логической матрицы (ПЛМ) – одной из структур СБИС – особое место занимает задача свертки ПЛМ, которая заключается в использовании одной и той же линии (горизонтальной или вертикальной) для укладки различных проводников, что приводит к сокращению площади кристалла, занимаемой проектируемой схемой.

К раскраске графа в работе [Бибилло, 1992] сводится получение многократной свертки ПЛМ, при которой на одной линии укладывается более чем два проводника, а внешние соединения переносятся в "третье измерение". Вершины рассматриваемого графа соответствуют проводникам, и две вершины связаны ребром, если и только если соответствующие проводники не могут быть уложены на одной линии. Раскраска графа указывает способ укладки проводников.

При заданном разбиении множества внешних сигналов на два подмножества, соответствующие сигналам, подводимым к ПЛМ с разных сторон, задача нахождения максимального числа свертываемых пар сводится к задаче нахождения максимального паросочетания в двудольном графе [Бибилло, 1992]. Доли этого графа соответствуют упомянутым подмножествам, и две вершины из разных долей связаны ребром, если соответствующие сигналы образуют пару свертки.

При свертывании горизонтальных линий ПЛМ также решается задача размыкания контуров в двудольном орграфе, после чего надо должным образом упорядочить вертикальные шины. Это упорядочение сводится к задаче топологической сортировки вершин бесконтурного орграфа [Бибилло, 1992]. Решением этой задачи является последовательность вершин, в которой для каждой дуги вершина, являющаяся концом этой дуги, расположена после вершины, являющейся ее началом.

В последнее время проектировщики электронной техники и исследователи в области

проектирования дискретных устройств стали уделять много внимания сокращению расхода энергии при эксплуатации проектируемого устройства. Это обусловлено стремлением увеличить время действия источника энергии в портативных приборах, а также снизить остроту проблемы отвода тепла при проектировании сверхбольших интегральных схем. Поэтому одним из основных критериев оптимизации при проектировании дискретных устройств является величина потребляемой энергии схемы. Потребляемая мощность схемы, построенной на основе КМОП-технологии, пропорциональна интенсивности переключений узлов схемы. Это дает возможность частично решать данную проблему на уровне логического проектирования. Если не учитывать задержку сигналов в логических элементах, то повлиять на интенсивность переключений в фиксированной схеме невозможно, но использовать эту величину для снижения потребления энергии можно при покрытии ее библиотечными элементами. Покрывать надо так, чтобы узлы схемы с наибольшей интенсивностью переключений по возможности оказались внутри библиотечных элементов. Действительно, сокращение длины соединительных проводников в схеме ведет к уменьшению паразитных емкостей схемы, на перезарядку которых в процессе переключений расходуется энергия питания. В статье [Поттосин, 2011] предлагается метод покрытия логической схемы библиотечными элементами, основанный на сведении данной задачи к изоморфному вложению графов.

Заключение

Приведенный обзор не охватывает многочисленных работ по приложениям теории графов на всех этапах проектирования дискретных устройств, но показывает, насколько широко методы теории графов могут быть использованы для решения задач на всех стадиях логического проектирования. Большинство из рассмотренных задач теории графов представлено в курсе дискретной математики для студентов Белорусского государственного университета информатики и радиоэлектроники, специализирующихся в области проектирования дискретных устройств.

Библиографический список

- [Агибалов, 1981] Агибалов, Г.П. Технология решения комбинаторно-логических задач методом сокращенного обхода дерева поиска / Г.П. Агибалов, В.А. Беляев. – Томск : Изд-во Томского ун-та, 1981. – 126 с.
- [Бибилло, 1987] Бибилло, П.Н. Синтез комбинационных схем методами функциональной декомпозиции / П.Н. Бибилло, С.В. Енин. – Минск : Наука и техника, 1987. – 189 с.
- [Бибилло, 1992] Бибилло, П.Н. Синтез комбинационных ПЛМ-структур для СБИС / П.Н. Бибилло. – Минск : Наука и техника, 1992. – 232 с.
- [Закревский, 1981] Закревский, А.Д. Логический синтез каскадных схем / А.Д. Закревский. – М. : Наука, 1981. – 416 с.
- [Закревский, 1986] Закревский, А.Д. Приложения теории графов к задачам логического проектирования дискретных устройств / А.Д. Закревский [и др.] // Исследования по

прикладной теории графов. – Новосибирск : Наука, Сибирское отделение, 1986. – С. 3-9.

[Закревский, 2000] Закревский, А.Д. Раскраска графов при декомпозиции булевых функций / А.Д. Закревский // Логическое проектирование. – Минск : Ин-т техн. кибернетики НАН Беларуси, 2000. – Вып. 5. – С. 83-97.

[Закревский, 2005] Закревский, А.Д. Об оптимальном размещении графа в булевом пространстве / А.Д. Закревский // Вестник ТГУ. Приложение. – 2005. – № 14. – С. 13-17.

[Закревский и др., 2007] Закревский, А.Д. Логические основы проектирования дискретных устройств / А.Д. Закревский [и др.]. – М. : Физматлит, 2007. – 592 с.

[Мелихов, 1971] Мелихов, А.Н. Ориентированные графы и конечные автоматы / А.Н. Мелихов. – М. : Наука, 1971. – 416 с.

[Мелихов и др., 1974] Мелихов, А.Н. Применение графов для проектирования дискретных устройств / А.Н. Мелихов [и др.]. – М. : Наука, 1974. – 304 с.

[Нечепуренко и др., 1990] Нечепуренко, М.И. Алгоритмы и программы решения задач на графах и сетях / М.И. Нечепуренко [и др.]. – Новосибирск : Наука, 1990. – 515 с.

[Поттосин, 1985] Поттосин, Ю.В. Выделение максимальной двудольной части в неориентированном графе / Ю.В. Поттосин // Автоматизация решения логико-комбинаторных задач. – Минск : Ин-т техн. кибернетики АН БССР, 1985. – С. 22-28.

[Поттосин, 2001] Поттосин, Ю.В. Задачи теории графов в логическом проектировании / Ю.В. Поттосин // Логическое проектирование, вып.6. – Минск : Ин-т техн. кибернетики НАН Беларуси, 2001. – С. 106-130.

[Поттосин, 2011] Поттосин, Ю.В. Основы теории проектирования цифровых устройств / Ю.В. Поттосин. – Saarbrücken : LAP LAMBERT Academic Publishing, 2011. – 336 с.

[Armstrong, 1962] Armstrong, D.B. A programmed algorithm for assigning internal codes for sequential machines / D.B. Armstrong. – IRE Trans., EC-11, 1962. – N 4. – P. 466-472.

[Perkowski, 1995] Perkowski, M. A survey of literature on functional decomposition / M. Perkowski, S. Grygiel. – Portland State University, Department of Electrical Engineering, 1995 (Technical report). – 188 p.

THE PROBLEMS ON GRAPH IN THE THEORY OF DESIGN OF DIGITAL DEVICES

Pottosin Yu.V. *, Pottosina S.A. **

*United Institute of Informatics Problems, NAS of Belarus, Minsk, Republic of Belarus

pott@newman.bas-net.by

** Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

pottosina@sum-solution.net

The problems on graphs are considered, which have applications in the design of discrete control devices. A review of methods and algorithms for solving these problems is given. Examples of applications of graph models for solving logical design problems are given.

INTRODUCTION

The object-oriented approach to solving the problems of logical design of discrete devices supposes constructing a knowledge base. It contains structured information and inference rules that cover a given knowledge area by library elements. In logical design of discrete devices, such elements may be logical combinatorial problems and algorithms to solve them that are formulated in graph theory terms.

Graph theory methods are widely used in logical design of discrete control and computation devices.

There are many works on applications graph theory in logical design and areas close to it, that are difficult to be enumerated. Therefore, we are restricted to mention some monographs [Мелихов, 1971], [Мелихов и др., 1974], [Нечепуренко и др., 1990] and reviews [Закревский и др., 1986], [Поттосин, 2001].

At the stage of logical design of discrete devices, various logical combinatorial problems arise [Агибалов, 1981], [Закревский и др., 2007], [Поттосин, 2011], whose solutions lead to optimal variants of logical projects. In this paper, we give a brief review of the methods and problems of graph theory that have wide application in logical design.

The concept of graph in the considered problems is used in different interpretations. A graph is often a graphic representation of a binary relation of compatibility, i.e. a relation having the properties of reflexivity and symmetry. Such a relation is represented by an undirected graph. In some cases, a graph represents a quasi-order relation or a consecution relation between some events. In some problems, a graph represents a logic network. The graphs are directed in these cases.

MAIN PART

The problem of graph coloring takes a considerable place in logical design. The graph coloring can be used in solving the problem of decomposition of Boolean functions that takes an important place in the synthesis of combinational circuits in VLSI basis. The problem of decomposition of Boolean functions may be set as follows. Given a system of Boolean functions as a vector function $y = f(x)$, and Boolean vector variables w and z composed of the components of vector variable x . The given system of Boolean functions must be represented as $f(x) = h(w, g(z))$. A method to solve this problem supposes constructing graph G whose vertices are values z^* of z , and there is an edge between z^*_i and z^*_j if and only if function f has different values at the same value of w . It is proved that f can be represented as $f(x) = h(w, u)$ where u has k components if and only if the chromatic number of G is at most 2^k .

As it was said above, to solve the problem of decomposition of Boolean functions, one should encode the obtained colors by Boolean or ternary vectors. These vectors are the values of vector variable u that has the property that each edge of the graph must correspond to a component of u having opposite values (0 and 1) at the ends of the edge. So, a bipartite subgraph can be put in correspondence to each component of u , and the problem can be considered as the problem of covering a graph by bipartite subgraphs. This problem arises in the search for optimal state assignment of a synchronous automaton. The specification of input and output is assumed to be in the structural alphabet. This model is called automaton with abstract state. An automaton is given by two ternary matrices, U and V . The columns of U correspond to input Boolean variables x_1, x_2, \dots, x_n and the columns of V binary output signals y_1, y_2, \dots, y_m . The rows of both matrices are marked by the states of the automaton so that a pair of their rows of the same

name represents transition from the state, which is the mark of the row of U , to the state, which is the mark of the row of V . At that, the row of U represents the transition condition, and the row of V shows the output signals accompanying this transition. The method is following. Graph G is constructed for some column y_i of V . The vertices of G correspond to the states of the given automaton. Pairs of elements with opposite values (0 and 1) are looked for in column y_i that correspond to pairs non-orthogonal rows of U . The vertices of G corresponding to the marks of the rows of any such pair are connected with an edge and than are encoded as it is shown above. The obtained codes are put in the matrices U and V as the values of the introduced internal variables z_1, z_2, \dots, z_k . This procedure is repeated for every column of V and each time a column with minimum of pairs of mentioned elements is selected. When the graph G is empty (after introducing new variables the rows of U become orthogonal), the process finishes and the transformed matrices U and V give the system of Boolean functions describing the combinational part of the logical circuit that implements the given automaton. Some compatible states can be encoded by non-orthogonal or the same codes. In this case the number of states of the automaton becomes reduced.

The problems of logical design formulated as problems of partitioning a given set into compatible subsets in a certain sense can be reduced to graph coloring if pair-wise compatibility of the set elements is enough for their group compatibility. Otherwise, one should introduce some additional conditions for the elements to be of the same color. Another approach to solving such problems assumes finding in a given graph the largest independent set, i.e. the largest set of pair-wise nonadjacent vertices, and then grouping at them the rest of vertices of the graph. This approach can be used to solve such problems as minimization of code length of states of an asynchronous automaton, compaction of the flow table of an automaton (by rows and columns), parallel decomposition of an automaton, simplification of the system of Boolean functions describing an asynchronous automaton. This technique suits especially for solving such problems of abstract synthesis of automata as state minimization and decomposition of a synchronous automaton. In these problems, the desired family of compatible sets must be closed, i.e. existence of some elements in a compatible subset demands including some other elements in another compatible subset belonging the same family. Finding complete subgraphs is a dual problem with respect to finding independent sets, because a complete subgraph of a graph is induced by an independent set of the complement of the graph. To find all maximal independent sets the classical Paull-Unger method for finding all maximal compatible sets of states of an automaton can be used. It uses a technique where the given graph is decomposed into the sequence of subgraphs, and the solution for preceding subgraph is transformed into the solution for the next subgraph. Another algorithm for finding all maximal independent sets is suggested that uses the lexicographical search.

The model of complete Boolean graph Q_k is used to

solve the problem of state assignment of discrete automata that must be solved to transit from behavior description of an automaton to structural one. The optimization criterion is the chip area or power consumption. The process of state assignment can be considered as assembling a graph Q_k whose vertices are Boolean vectors (elements of Boolean space). The process of constructing the graph Q_k , which is called k -dimensional hyper-cube, is a sequence of k steps. At the p -th step, the set of $(p-1)$ -dimensional hyper-cubes (graphs Q_{p-1}) is considered. They are joint in pairs, and adding edges transforms each pair into one p -dimensional hyper-cube (graph Q_p). At that, as far as it is possible, those pairs of vertices are chosen for connecting with edges, which maximal values of some integral function w_{ij} determined at the pairs of states correspond to. The vertices of the obtained graph Q_k are assigned k -component Boolean vectors keeping neighbor relation between the vertices represented by the edges of graph Q_k .

Rather considerable variety of graph problems is in optimization of the structures of very large scale integration (VLSI). The problems of optimization of VLSI structures are at the turn of logical and technological designs. In the process of optimization of the structure of a programmable logical array (PLA), which is one of VLSI structures, the problem of folding PLA, i.e. using the same line (horizontal or vertical) to embed different wires. This results in reducing the chip area. In [Бибило, 1992], finding a multiple folding of a PLA, when more than two wires are embedded on the same line and outer connection is carried to "the third dimension", is reduced to graph coloring. The vertices of the graph correspond to the wires and two vertices are adjacent if and only if the correspondent wires cannot be embedded on the same line. The graph coloring indicates the way of wire embedding. Given a partition of the set of external signals into two subsets corresponding to the signals supplied to the PLA from different sides, the problem of finding the maximal number of folded pairs is reduced to the problem of finding the maximal matching in a bipartite graph. The parts of this graph correspond to the mentioned subsets, and there is an edge between two vertices of different parts if the corresponded signals form a pair of folding.

CONCLUSION

This review does not include many works on applications of graph theory at all the stages of designing digital devices, but it shows how wide the graph theory methods can be used in logical design. The majority of the considered problems are present in the course of discrete mathematics for students of Byelorussian State University of Informatics and Radio-Electronics.



УДК 004.822:514

DECOMPOSIBILITY AND SEARCH FOR ALL SOLUTIONS OF A SYSTEM OF BOOLEAN FUNCTIONS

Taghavi S.A.* and Pottosin Y.V.*

* *United Institute of Informatics Problems of the NAS of Belarus,
Minsk, Republic of Belarus*

{taghavi,pott}@newman.bas-net.by

Abstract: The problem of series two-block decomposition of completely specified Boolean functions is considered. Analysis and investigation of such systems are very important in logical design context. Recently, a method for solving this problem was suggested based on using the ternary matrix cover approach. Using this method a computer program developed. This paper is focused on decomposability of a system of Boolean functions. In decomposable systems, the number of solutions and time elapsed to achieve them was investigated.

Keywords: Boolean functions, decomposition, cover map, compact table

1. INTRODUCTION

The problem of decomposition of Boolean functions is one of the most important problems of logical design that makes it an object of great attention by many researchers in this field. The survey [Perkowski, 1995] shows a considerable number of papers are already published on this topic. It is important to find a successful solution for this problem because it has a direct influence on the quality and cost of digital devices designed. We consider the problem of decomposition of a system of Boolean functions in the following statement. A system of completely specified Boolean functions $y=f(x)$ is given where $y=(y_1, y_2, \dots, y_m)$, $x=(x_1, x_2, \dots, x_n)$, $f(x)=(f_1(x), f_2(x), \dots, f_m(x))$. The superposition $y=\varphi(w, z_2)$, $w=g(z_1)$ where z_1 and z_2 are vector variables whose components are Boolean variables in the subsets Z_1 and Z_2 respectively that form a partition of the set $X=\{x_1, x_2, \dots, x_n\}$ of arguments. At that, the number of components of the vector variable w must be less than that of z_1 . Such a kind of decomposition is called two-block disjoint decomposition by [Zakrevskij, 2009]. The subsets Z_1 and Z_2 are called bound and free sets respectively. Only a few papers deal with the search for the partition $\{Z_1, Z_2\}$, at which this problem has a solution. Among the papers considering this question, we can point out [Bibilo, 2009], [Jóźwiak, 2000], [Perkowski, 1995] and [Zakrevskij, 2007].

Searching for a solution of this kind is NP-complete problem because it has been proved that this problem equivalent with well-known set covering problem (SCP). While to be aware of decomposability of the

given system of Boolean functions, finding only one such a pair is satisfying, due to analyzing of the task and search for the best solution we were motivated to find all possible solutions. For that we used the ternary matrix cover approach [Pottosin, 2010]. Using a compact table one can find rather easily the existence of a solution of the problem for a given system of functions, and if it does exist, the corresponding superposition can be easily found.

2. DEFINITIONS AND SETTING THE PROBLEM

Let a system of completely specified functions $y=f(x)$, where $y=(y_1, y_2, \dots, y_m)$, $x=(x_1, x_2, \dots, x_n)$ and $f(x)=(f_1(x), f_2(x), \dots, f_m(x))$, be given by matrices U, V that are the matrix representation of the system of disjunctive normal forms (DNFs) of the given functions [Zakrevskij, 2009]. Matrix U is a ternary matrix of $l \times n$ dimension where l is the number of terms in the given DNFs. The columns of U are marked with the variables x_1, x_2, \dots, x_n , and the rows represent the terms of the DNFs (the intervals of the space of the variables x_1, x_2, \dots, x_n). The matrix V is a Boolean matrix. Its dimension is $l \times m$, and its columns are marked with the variables y_1, y_2, \dots, y_m . The ones in this columns point out the terms in the given DNFs. A row u in U absorbs a Boolean vector a if a belongs to the interval represented by u .

The task considered is set as follows. Given a system of completely specified Boolean functions $y=f(x)$, the superposition $y=\varphi(w, z_2)$, $w=g(z_1)$ must be found where z_1 and z_2 are vector variables whose components are Boolean variables in the subsets of the set $X=\{x_1, x_2, \dots, x_n\}$, Z_1 and Z_2 respectively such that

$X = Z_1 \cup Z_2$ and $Z_1 \cap Z_2 = \emptyset$. At that, the number of components of the vector variable \mathbf{w} must be less than that of z_1 . The main attention is paid to the search for subsets Z_1 and Z_2 such that the task would have a solution. It is clear that the subset Z_1 should have at least two members while the subset Z_2 can have only one.

3. INTRODUCING COVER MAP AND COMPACT TABLE

Any family π of different subsets (blocks) of a set L whose union is L , is called a *cover* of L . Let $L = \{1, 2, \dots, l\}$ be the set of numbers of rows of a ternary matrix U . A cover π of L is called a *cover of the ternary matrix U* if for each value x^* of the vector variable \mathbf{x} there exists a block in π containing all the numbers of those and only those rows of U , which absorb x^* . Block \emptyset corresponds to the value x^* , which is absorbed by no row of U . Other subsets are not in π .

Let $t(x^*, U)$ be the set of numbers of those rows of U , which absorb x^* . For every block π_j of π , we define the Boolean function $\pi_j(x)$ having assumed that $\pi_j(x^*) = 1$ for any $x^* \in \{0,1\}^n$ if $t(x^*, U) = \pi_j$, and $\pi_j(x^*) = 0$ otherwise.

Let us define an operation $\vee(\pi_i, V)$ over the rows of a binary matrix V , the result of which is the vector y^* ($y^* = \vee(\pi_i, V)$) obtained by component-wise disjunction of rows V whose numbers are in the block π_i . If $\pi_i = \emptyset$, all the components of y^* are equal to 0. It is shown in [Pottosin, 2006] that $f(x^*) = y^* = \vee(\pi_i, V)$ if $\pi_i(x^*) = 1$.

There is a convenient way to construct the cover of a ternary matrix U when the number of arguments is not large. This technique uses the cover map that has the structure of the Karnaugh map. In any cell of a cover map of U corresponding to a vector x^* , there is the set $t(x^*, U)$, which is a block of the cover of U .

Let a pair of matrices, U and V , give a system of completely specified Boolean functions $\mathbf{y} = \mathbf{f}(\mathbf{x})$, and let the matrix U_1 be composed of the columns of U , marked with the variables from the set Z_1 and the matrix U_2 from the columns marked with the variables from Z_2 . The covers of U_1 and U_2 are $\pi^1 = \{\pi^1_1, \pi^1_2, \dots, \pi^1_r\}$ and $\pi^2 = \{\pi^2_1, \pi^2_2, \dots, \pi^2_s\}$. Let us construct a table M . Assign the blocks $\pi^1_1, \pi^1_2, \dots, \pi^1_r$ and the Boolean functions $\pi^1_1(z_1), \pi^1_2(z_1), \dots, \pi^1_r(z_1)$ to the columns of M , and $\pi^2_1, \pi^2_2, \dots, \pi^2_s$ and $\pi^2_1(z_2), \pi^2_2(z_2), \dots, \pi^2_s(z_2)$ to the rows of M . At the intersection of the i -th column, $1 \leq i \leq r$ and the j -th row, $1 \leq j \leq s$, of M , we put the value $y^* = \vee(\pi^1_i \cap \pi^2_j, V)$. The table M is called the *compact table*. It gives the system of Boolean functions $\mathbf{y} = \mathbf{f}(\mathbf{x})$ in the following way: the value of the vector Boolean function $\mathbf{f}(x^*)$ is $\vee(\pi^1_i \cap \pi^2_j, V)$ at any set argument values x^* , for which $\pi^1_i(z_1) \wedge \pi^2_j(z_2) = 1$.

Having the compact table for a system of functions $\mathbf{y} = \mathbf{f}(\mathbf{x})$, it is easy to construct the desired systems $\mathbf{y} = \boldsymbol{\varphi}(\mathbf{w}, \mathbf{z}_2)$ and $\mathbf{w} = \mathbf{g}(\mathbf{z}_1)$. The columns of the compact table are encoded with binary codes; equal columns

may have the same codes. The length of the code is equal to $\lceil \log_2 r \rceil$ where r is the number of different columns of the table and $\lceil a \rceil$ is the least integer, which is not less than a . So, the system of functions $\mathbf{w} = \mathbf{g}(\mathbf{z}_1)$ is defined. The value of the vector variable \mathbf{w} at any set of values of the vector variable \mathbf{z}_1 turning the function $\pi^1_i(z_1)$ into 1 is the code of the i -th column, $1 \leq i \leq r$. Naturally, there is no solution to this task at the given partition $\{Z_1, Z_2\}$ of the set X of arguments if the length of the code is not less than the length of \mathbf{z}_1 . Otherwise, the compact table whose columns are assigned with the values of the variable \mathbf{w} can be considered as a form of representation of the other desired system of functions $\mathbf{y} = \boldsymbol{\varphi}(\mathbf{w}, \mathbf{z}_2)$. The value of \mathbf{y} at the value of \mathbf{w} assigned to the i -th column, $1 \leq i \leq r$, and at any value of \mathbf{z}_2 turning $\pi^2_j(z_2)$ into 1, $1 \leq j \leq s$, is the vector that is at the intersection of the i -th column and the j -th row [Pottosin, 2010].

Example 1. Let a system of completely specified functions $\mathbf{y} = \mathbf{f}(\mathbf{x})$ was given by the following pair of matrices:

$$U = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 1 & - \\ 0 & 1 & 0 & 0 & - \\ 0 & 1 & - & 0 & 1 \\ 0 & - & 0 & 0 & - \\ 0 & 0 & - & 0 & 1 \\ 1 & 1 & 0 & 1 & - \\ 1 & 1 & - & 1 & 1 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} \end{matrix}, \quad V = \begin{matrix} & y_1 & y_2 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} \end{matrix}.$$

For the partition of the set of arguments into subsets $Z_1 = \{x_1, x_2, x_3\}$ and $Z_2 = \{x_4, x_5\}$, we have the following matrices:

$$U_1 = \begin{matrix} & x_1 & x_2 & x_3 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & - \\ 0 & - & 0 \\ 0 & 0 & - \\ 1 & 1 & 0 \\ 1 & 1 & - \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} \end{matrix}, \quad U_2 = \begin{matrix} & x_4 & x_5 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{bmatrix} 1 & - \\ 0 & - \\ 0 & 1 \\ 0 & - \\ 0 & 1 \\ 1 & - \\ 1 & 1 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} \end{matrix}.$$

To find the length of \mathbf{w} in the superposition $\mathbf{y} = \boldsymbol{\varphi}(\mathbf{w}, \mathbf{z}_2)$, $\mathbf{w} = \mathbf{g}(\mathbf{z}_1)$ where $\mathbf{z}_1 = (x_1, x_2, x_3)$ and $\mathbf{z}_2 = (x_4, x_5)$, we construct the covers of the ternary matrices U_1 and U_2 : $\pi^1 = \{\emptyset, \{3\}, \{5\}, \{7\}, \{6, 7\}, \{1, 4, 5\}, \{2, 3, 4\}\}$ and $\pi^2 = \{\{1, 6\}, \{2, 4\}, \{1, 6, 7\}, \{2, 3, 4, 5\}\}$ (In continue we will discuss how to obtain this covers). The corresponding compact table is represented in Table 1 that has seven different columns.

Table 1 – The compact table for the system of functions in Example 1 with $\mathbf{z}_1 = (x_1, x_2, x_3)$ and $\mathbf{z}_2 = (x_4, x_5)$

	\emptyset	3	5	7	6,7	1,4,5	2,3,4
1,6	00	00	00	00	01	10	00
2,4	00	00	00	00	01	01	11
1,6,7	00	00	00	01	01	10	00
2,3,4,5	00	10	01	00	00	11	11

Clearly, this task has no solution at the given subsets Z_1 and Z_2 , because to encode the columns of the compact table with the values of w , three variables are needed that is not less than the length of z_1 .

4. SEARCH FOR APPROPRIATE PARTITION

To search for an appropriate partition of the set of arguments we use ternary matrix covers and compact tables induced by them. Let a few free variables be to find that constitute the set Z_2 (then the set of bound variables would be $Z_1 = X \setminus Z_2$). To do this, we use the operation of dividing a ternary matrix cover by the cover of a column of the matrix. Let us determine the operation to divide the cover π of a ternary matrix U by the cover π^i of its i -th column as:

$$\pi / \pi^i = \pi^1 \times \pi^2 \times \dots \times \pi^{i-1} \times \pi^{i+1} \times \dots \times \pi^n$$

This operation can be easily fulfilled using the *cover map*, which, as well as Karnaugh map, has the lines of symmetry related to the variables of the Boolean space represented by this map [Zakrevskij, 2007]. To transform the cover map of a ternary matrix U into that of the matrix obtained from U by deleting the i -th column, one should superpose pair-wise the entries that are symmetric with regard to the lines relative to x_i , and put the unions of the superposed entries into the obtained entries. The obtained cover map would represent the desired cover [Pottosin, 2010].

Example 2. Figure 1 shows the cover map of the ternary matrix U from Example 1.

Figure 1 – The cover map of matrix U from Example 1

Figure 2 – The cover map obtained by dividing π by the cover of the column x_4 .

Figure 3 – The cover map obtained by dividing π by the covers of the column x_4 and x_5 .

The cover of U is $\pi = \{\emptyset, \{1\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{2,4\}, \{4,5\}, \{6,7\}, \{2,3,4\}\}$. As it can be seen from Figure 2 the division of π by the cover of the column x_4 will be $\{\emptyset, \{3\}, \{5\}, \{6\}, \{7\}, \{1,4\}, \{2,4\}, \{6,7\}, \{1,4,5\}, \{2,3,4\}\}$. Having transformed this map by the described way with regard to x_5 , we obtain $\{\emptyset, \{3\}, \{5\}, \{7\}, \{6,7\}, \{1,4,5\}, \{2,3,4\}\}$ as a result of dividing π by the covers of the columns x_4 and x_5 (see Figure 3).

The method used for the search for an appropriate partition consists in fulfilling the lexicographical enumeration and testing by the above way every variant of the set Z_2 if it would provide a solution of the task.

Example 3. Let the system of completely specified Boolean functions from Example 1 be given. Consider this variant that $Z_2 = \{x_2, x_4\}$, $Z_1 = \{x_1, x_3, x_5\}$. For this variant with the cover map in Figure 1, we obtain the cover map shown in Figure 4 and then we obtain Figure 5 from Figure 4.

Figure 4 – The cover map obtained by dividing π by the cover of the column x_2 .

Figure 5 – The cover map obtained by dividing π by the covers of the column x_2 and x_4 .

The compact table for the covers $\pi^1 = \{\emptyset, \{6\}, \{7\}, \{3,5\}, \{6,7\}, \{1,2,4\}, \{1,2,3,4,5\}\}$ and $\pi^2 = \{\{1\}, \{4,5\}, \{6,7\}, \{2,3,4\}\}$ is represented by Table 2 that have four different columns. To encode these columns, two variables are sufficient. The codes of the columns are shown at the bottom of Table 2.

Table 2 – The compact table for the partition from Example 3

	\emptyset	6	7	3,5	6,7	1,2,4	1,2,3,4,5
1	00	00	00	00	00	10	10
4,5	00	00	00	01	00	01	01
6,7	00	01	01	00	01	00	00
2,3,4	00	00	00	10	00	11	11
	00	01	01	10	01	11	11

To construct the system of functions $y = \varphi(w, z_2)$ and $w = g(z_1)$ that are the solution of the task, the functions connected with the blocks of the covers obtained must be constructed. The DNFs of the functions connected with the blocks of π^1 can be obtained from the cover map in Figure 5: $\pi^1_1(z_1) = x_3 \bar{x}_5$, $\pi^1_2(z_1) = x_1 \bar{x}_3 \bar{x}_5$, $\pi^1_3(z_1) = x_1 x_3 x_5$, $\pi^1_4(z_1) = \bar{x}_1 x_3 x_5$, $\pi^1_5(z_1) = x_1 \bar{x}_3 x_5$, $\pi^1_6(z_1) = \bar{x}_1 \bar{x}_3 \bar{x}_5$, $\pi^1_7(z_1) = \bar{x}_1 \bar{x}_3 x_5$.

Similarly, the DNFs $\pi^2_1(z_2) = \bar{x}_2 x_4$, $\pi^2_2(z_2) = \bar{x}_2 \bar{x}_4$, $\pi^2_3(z_2) = x_2 x_4$, $\pi^2_4(z_2) = x_2 \bar{x}_4$ are obtained. As a result of simple minimization we obtain the following matrices representing the desired superposition $y = \varphi(w, z_2)$, $w = g(z_1)$:

$$\begin{bmatrix} w_1 & w_2 & x_2 & x_4 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & - & 1 & 0 \\ 1 & - & 0 & 0 \\ 1 & 1 & - & 0 \end{bmatrix}, \begin{bmatrix} y_1 & y_2 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}; \quad \begin{bmatrix} x_1 & x_3 & x_5 \\ 0 & 0 & - \\ - & 0 & 0 \\ 1 & - & 1 \end{bmatrix}, \begin{bmatrix} w_1 & w_2 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

5. IMPLEMENTATION AND RESULTS

We designed and developed special computer program on MSVC++ to find all solutions of systems of Boolean functions. Our program based on ternary matrix cover approach and the general scheme of our algorithm summarized in Figure 6. The experiments run on a Pentium 2.26GHz CPU with 3 GByte of main memory. As a benchmark, we generate many systems of completely specified Boolean functions using a prepared library explained in [Romanov, 1997], [Romanov, 2001] and [Romanov, 2005]. We considered three parameters for these systems; number of rows of matrix U that indicate number of conjunctions, number of columns of matrix U or number of arguments and number of columns of matrix V or number of functions. For every system and after generating matrices U and V as SOP (Some Of Product), first of all we expand matrix U to obtain corresponding matrix without don't cares. The rows which have don't cares will replace with several suitable rows and consequently number of DNFs of system will increase exponentially.

Then we begin to provide cover map; for that we used gray code encoding system. On contrary to our example in section 4 that cover map is a two dimensional table, due to simplicity to store in computer memory and referring for it and also in the future calculations of compact table, we implemented it as a one dimensional array. An example of our approach with three variables is represented in Figure 5. The order of replacement of the variables on the array is important and this can be extended for any number of variables.

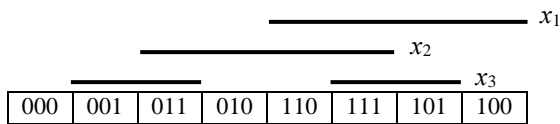


Figure 5 – The cover map model for three variables using gray code encoding system

In the array we store the values explained in section 4 and gray codes in the array of Figure 5 are symbolically shown to represent the correctness of the approach, but we also save the gray codes in the other list. In fact, the method of storing information in the mentioned array is as follows. Each row in matrix U , numbered with integers started from one. We compare the value of each row with gray codes list until the equal value to be founded. Then we add the row

number of compared row in corresponding element of the array. We continue this way until all rows to be compared and the row numbers to be added to the array elements. At the final, we sweep the array and put the empty set to the elements with no value added.

-Algorithm for one system of Boolean functions

- $Con \leftarrow ConValue$
- $Arg \leftarrow ArgValue$
- $Fun \leftarrow FunValue$
- Generate **SOP** (Con, Arg, Fun)
(i.e Matrices U and V)
- Expand Matrix U
(i.e removing all don't cares in Matrix U with replacing the according rows with new ones)
- Compute Cover Map
(Generate Gray Codes with Length 2^n and fill out the Cover Map Array According to the Algorithm Rules)
- $C \leftarrow 0$ (To Count Number of Solutions)
- for $k \leftarrow 2$ to $n-1$
 - »Combination Generator(n, k)
 - »for each combination of $\binom{n}{k}$
 - Check Current Partition
 - 1-Divide Cover Map Over Z_1
 - 2-Divide Cover Map over Z_2
 - 3-Compute Compact Table
 - 4-Compute Number of Different Columns (r) of Compact Table
 - 5-Encode the Columns of the Compact Table
 - 6-if $\log(r) \leq 2^{k-1}$ then
 - Solution Founded
 - (Produce the Solution i.e Matrices φ, w, y and x)
 - »if Solution Founded then
 - Add this partition to the set of Solutions and $C \leftarrow C + 1$
- if $C=0$ then
 - Declare the system is not decomposable
- else
 - Print C (which is the number of all solutions)

Figure 6 – Implemented algorithm for determining decomposability of system of Boolean Functions and find total number of solutions of system

To find all solutions of the task anyone should enquire into all possible partitions which constructing Z_1 and Z_2 . The relatively simple method to address the appropriate partition can be done by lexicographical enumeration. After computing cover map of the current system of Boolean functions, in each stage we used Donald E. Knuth algorithm [Knuth, 2011] to generate all combinations of the arguments and of course for each partition we check whether it is a solution of the task or not. To obtain all k -element subsets of an n -element set, this algorithm is one of the fastest ones. Each k -element subsets is used to construct Z_1 elements and rest of the arguments will be the elements of the Z_2 .

If a partition as a solution found, the program will keep it and will calculate four matrices; Matrix Φ , Matrix Y , Matrix X and Matrix W . These matrices are the solution of the task. In fact the current system of Boolean functions convert to two new systems with less arguments; Matrices Φ and Y as U and V respectively, for the first system and also Matrices X and W as U and V respectively, for the second system. Although we obtain these matrices but they haven't influence in our results in this paper. We'll utilize them in the future works.

This manner is repeated for all partitions and if appropriate partition wasn't found, the program will declare the current system of Boolean functions isn't decomposable; otherwise the program will print number of solutions to the current system.

Now, we report experimental results for our approach in decomposition of Boolean functions, described in the previous sections. Due to space and time limitations, the results are shown refer only to the decompositions of systems with few arguments and too few functions. The results summarized in Table 3.

Table 3 – Experimental results

Con	Arg	Fun	NTP	NS	PS	ET
8	5	2	25	21	84	<1
10	6	2	56	49	87	4
10	6	3	56	13	23	5
10	6	4	56	18	32	4
20	7	3	119	25	21	21
15	8	3	246	64	26	131
15	8	4	246	40	16	134
20	8	5	246	8	3	104
40	8	10	246	8	3	201
20	9	4	501	48	10	497
30	9	5	501	10	2	688
30	10	3	1012	58	6	3634
30	10	5	1012	55	5	3197
40	10	4	1012	98	10	3100
30	12	6	4082	673	16	20413

The results show that more than 95% of generated systems are decomposable and all of them have several solutions when the system is decomposable. The first three columns in Table 3 represent the number of conjunctions (Con), number of arguments (Arg) and number of functions (Fun) respectively and these informs the parameters of a generated system of Boolean functions. The number of total partitions (NTP) counted when $2 \leq |Z_1| \leq n - 1$. So it implies that the total partitions will be $\sum_{k=2}^{n-1} \binom{n}{k}$ which it is equal to $2^n - (n + 2)$. Number of Solutions (NS) is part of results which it is found after the program was executed and the percentage of the Solutions (PS) is percent of NS to NTP. The last column represents elapsed time (ET) which is the running time of the program for each system of Boolean functions during obtaining all solutions and calculated in seconds.

CONCLUSION AND FUTURE WORK

We developed a computer program as an application to determine decomposability of system of Boolean functions via ternary matrix cover approach. The ternary matrix cover and the representation of a system of Boolean functions in the form of compact table are simple to realize and we implemented several systems with different parameters. Experimental results were interesting and show that usually a system has more solutions when it is decomposable. In the most cases the number of solutions will be high when the number of functions is small.

As a future work, optimization in encoding of compact table is proposed, because it has direct influence on quality of obtained solutions. Also it is useful to find a best partition among the solutions from the syntheses point of view which is useful in practical scene.

REFERENCES

- [Pottosin, 2010] Pottosin, Yu.V., Shestakov, E., "Choice of Free Arguments in Decomposition of Boolean Functions Using the Ternary Matrix Cover Approach", *In Proceeding of the 6th International Conference on Neural Networks and Artificial Intelligence ICNNAI*, Brest, Belarus; BSTU, 2010, pp. 123-127.
- [Bibilo, 2009] Bibilo, P.N., "Decomposition of Boolean Functions Based on Solving Logical Equations", *Byelaruskaya Navuka*, Minsk, Belarus, 2009, (In Russian).
- [Józwiak, 2000] Józwiak, L. and Chojnacki A., "An Effective and Efficient Method for Functional Decomposition of Boolean Functions Based on Information Relationship Measures", *In Proceeding of 3rd Design and Diagnostics of Electronic Circuits and Systems Workshop, DDECS*. Bratislava: Institute of Informatics, Slovak Academy of Sciences, pp. 242-249, 2000.
- [Perkowski, 1995] Perkowski, M.A., Grygiel S., "A Survey of Literature on Functional Decomposition. Version IV", *Technical report*, Department of Electrical Engineering, Portland State University, Portland, USA, 1995.
- [Pottosin, 2006] Pottosin, Yu.V., Shestakov E.A., "Tabular Methods for Decomposition of Systems of Completely Specified Boolean Functions", *Byelorusskaya Nauka*, Minsk, Belarus, 2006, (In Russian).
- [Zakrevskij, 2007] Zakrevskij, A.D., "Decomposition of Partial Boolean Functions: Testing for Decomposability According to a Given Partition", *Informatika Journal*, No. 1(13), pp. 16-21, 2007, (In Russian).
- [Zakrevskij, 2009] Zakrevskij A., Pottosin Yu. V., Cheremisinova L., "Optimization in Boolean Space", *Tallinn: TUT Press*, 2009.
- [Romanov, 1997] Romanov, V.I. and Vasilkova I.V., "Boolean Vectors and Matrices in C++", *Logic Design, Institute of Engineering Cybernetics of NASB*, Minsk, Belarus, pp. 150-158, 1997, (In Russian).
- [Romanov, 2001] Romanov, V.I., "Tools development for logic designing", *Logic Design, Institute of Engineering Cybernetics of NASB*, Minsk, Belarus, pp. 151-170, 2001, (In Russian).
- [Romanov, 2005] Romanov, V.I., "Tools for programming Boolean calculations", *Combinatorics for modern manufacturing, logistics and supply chains, Abstracts of the XVIII European Conference*, UIIP, NASB, Minsk, Belarus, C.57-58, 2005.
- [Knuth, 2011] Knuth, D. E., "The Art of Computer Programming", Vol. 4A, *Combinatorial Algorithms*, part 1, Addison-Wesley Professional, ISBN-0201038048, 2011.

РАЗЛОЖИМОСТЬ И ПОИСК ВСЕХ РЕШЕНИЙ СИСТЕМЫ БУЛЕВЫХ ФУНКЦИЙ

Тагави С.А. *, Поттосин Ю.В. *

* Объединенный институт проблем информатики НАН Беларуси, г. Минск, Республика Беларусь

{taghavi,pott}@newman.bas-net.by

Аннотация. Рассматривается задача последовательной двухблочной декомпозиции полностью определенных булевых функций. Анализ и исследование таких систем является весьма важным для логического проектирования. Ранее был предложен метод решения этой задачи, основанный на использовании покрытия троичной матрицы. Разработана компьютерная программа, использующая этот метод. Основное внимание данной работы сосредоточено на разложимости систем булевых функций. Для разложимых систем исследовалось число решений и время, затрачиваемое на их получение.

Ключевые слова: булева функция, декомпозиция, карта покрытия, компактная таблица.

ВВЕДЕНИЕ

Рассматривается задача декомпозиции системы булевых функций в следующей постановке. Пусть система полностью определенных булевых функций $y = f(x)$, где $y = (y_1, y_2, \dots, y_m)$, $x = (x_1, x_2, \dots, x_n)$ и $f(x) = (f_1(x), f_2(x), \dots, f_m(x))$ задана матрицами U и V , представляющими систему дизъюнктивных нормальных форм (ДНФ) заданных функций. Матрица U является троичной матрицей размерности $l \times n$, где l – число различных элементарных конъюнкций в заданной системе ДНФ. Столбцы матрицы U помечены переменными x_1, x_2, \dots, x_n , а строки представляют упомянутые элементарные конъюнкции (интервалы пространства переменных x_1, x_2, \dots, x_n). Матрица V является булевой матрицей размерности $l \times m$, и ее столбцы помечены переменными y_1, y_2, \dots, y_m . Требуется найти суперпозицию $y = \varphi(w, z_2)$, $w = g(z_1)$, где z_1 и z_2 – векторные переменные, компонентами которых являются булевы переменные из подмножеств Z_1 и Z_2 соответственно, образующих разбиение множества $X = \{x_1, x_2, \dots, x_n\}$ аргументов такое, что $X = Z_1 \cup Z_2$ и $Z_1 \cap Z_2 = \emptyset$. При этом число компонент u векторной переменной w должно быть меньше чем u z_1 [Zakrevskij, 2009].

ОСНОВНАЯ ЧАСТЬ

Разработана компьютерная программа нахождения всех решений рассматриваемой задачи для заданной системы булевых функций. В качестве примеров генерировались многие системы полностью определенных булевых функций с

помощью программ, описанных в работах [Romanov, 2001] и [Romanov, 2005]. Менялись три параметра этих систем: число элементарных конъюнкций, число аргументов и число функций. Затем строилась матрица покрытия, в которой использовался код Грея и которая представлялась как одномерная строка.

Для нахождения всех решений необходимо рассматривать все возможные разбиения, образованные множествами Z_1 и Z_2 . Сравнительно простой метод обращения к подходящему разбиению использует лексикографический перебор. После получения матрицы покрытия для текущей системы булевых функций на каждом этапе генерируются все сочетания аргументов, а для каждого разбиения проверяется, имеется ли для него решение или нет. Если для разбиения существует решение, программа его сохраняет. Это повторяется для всех разбиений, и если подходящее разбиение не найдено, то программа объявляет текущую систему булевых функций неразложимой. Общая схема алгоритма представлена на рис. 6.

Результаты, приведенные в табл. 3, показывают, что сгенерированные системы имеют несколько решений, если они разложимы. Первые три столбца в табл. 3 представляют соответственно число элементарных конъюнкций (Con), число аргументов (Arg) и число функций (Fun). Общее число разбиений (NTP) подсчитано, когда $2 \leq |Z_1| \leq n - 1$. Этим числом будет $\sum_{k=2}^{n-1} \binom{n}{k}$, что равно $2^n - (n + 2)$. Частью результатов является число решений (NS), получаемое в результате выполнения программы, и доля NS в процентах (PS) по отношению к NTP. Последний столбец представляет время получения всех решений (ET) в секундах.

ЗАКЛЮЧЕНИЕ И ДАЛЬНЕЙШИЕ РАБОТЫ

Разработана компьютерная программа определения разложимости систем булевых функций, использующая подход, связанный с покрытием троичной матрицы. Исследовано несколько систем с различными параметрами. Определенный интерес представляют экспериментальные результаты, показывающие, что если система разложима, то рассматриваемая задача имеет несколько решений. В большинстве случаев число решений велико при малом числе функций. В качестве дальнейшей работы предполагается исследовать оптимизацию кодирования компактной таблицы, поскольку это непосредственно влияет на качество получаемого решения. Полезно также в практическом плане с точки зрения синтеза логических схем находить наилучшее разбиение множества аргументов.

АВТОРСКИЙ УКАЗАТЕЛЬ

Аболмасов П.В.	195	Жидченко В.В.	195
Алексеев А.А.	153	Жилякова Л.Ю.	87
Алексеев А.В.	417	Житко В.А.	505
Алиев Р.М.	395	Жукевич А.И.	295
Анищенко И.С.	295	Жуков И.И.	453
Ануреев И.С.	43	Заболеева-Зотова А.В.	341, 345, 399, 497
Апанасович В.В.	111	Завгородний В.В.	103
Атучин М.М.	43	Загорулько Ю.А.	31, 265
Барлыбаев А.Б.	547, 551	Загорулько М.Ю.	159
Бекманова Г.Т.	337	Загорулько Г.Б.	265
Беляков С.Л.	413	Зуенко А.А.	165
Белякова М.Л.	413	Иванченко Н.А.	437
Бибило П.Н.	473	Иванченко Г.Ф.	437
Бобков А.С.	497, 501	Ивашенко В.П.	171
Боргест Н.М.	441, 445, 449	Кабыш А.С.	477
Бородаенко Ю.В.	107	Казаков В.А.	533
Бранцевич П.Ю.	569	Касторнов А.Ф.	461
Брэк Д.В.	375	Касторнова В.А.	461
Бурдо Г.Б.	467	Килеев В.В.	333
Вагапов Д.Р.	329	Кинтонова А.Ж.	551
Вереник Н.Л.	129	Кириллович А.В.	91
Ветров Ю.А.	407	Киселёв В.В.	517
Виноградов Г.П.	299	Клещев А.С.	25
Вишняков В.А.	107	Клюшин А.Ю.	309
Галушка И.Н.	103	Князихина Ю.Е.	449
Гатиатуллин А.Р.	329	Коварцев А.Н.	195
Гецэвіч Ю.С.	319, 375, 489, 505	Козлов О.А.	529
Глоба Л.С.	49	Колб Д.Г.	115
Голенков В.В.	55	Комарцова Л.Г.	285
Головко В.А.	477	Константинов В.М.	357
Головня А.И.	521	Коровин М.Д.	441, 445
Горбачёв Н.Н.	541	Корончик Д.Н.	125, 403
Гракова Н.В.	453	Крюков К.В.	143
Грибова В.В.	25	Кузнецов О.П.	143
Гулякина Н.А.	55	Кузнецов В.Н.	299, 309
Давыденко И.Т.	185	Кулик Б.А.	165
Дёмин В.В.	477	Кулинич А.А.	135
Дианов И.А.	357	Ландэ Д.В.	457
Дмитриев А.С.	345, 351	Ланин В.В.	97
Добров Б.В.	153	Лапшов Ю.А.	423
Додонов А.Г.	457	Лейченко А.Н.	561
Дорофеев Н.С.	399, 501	Ли И.	569
Дунец И.П.	477	Липницкий С.Ф.	229
Дунец А.П.	477	Лобанов Б.М.	505
Елисеева О.Е.	511, 517	Лукашевич Н.В.	153
Ергеш Б.Ж.	337	Лядова Л.Н.	97
Ермоленко Т.В.	369	Маджиднежад В.М.	387
Ефименко И.В.	241	Маклаев В.А.	423

Мальковский М.Г.	147	Татур М.М.	129
Мамчич А.А.	229	Тельнов Ю.Ф.	533
Массель Л.В.	247	Герновой М.Ю.	49
Массель А.Г.	247	Тимченко В.А.	81
Мельникова В.В.	467	Требухин А.Г.	433
Михайлов Ю.Ф.	529	Трембач В.М.	533
Моросанова Н.А.	281	Филатова Н.Н.	433
Муканова А.С.	337	Филиппов А.А.	219
Мутовкина Н.Ю.	309	Фридман А.Я.	165
Найденова К.А.	289	Фурман О.Д.	115
Наместников А.М.	219	Харламов А.А.	369
Невзорова О.А.	91	Хейдоров И.Э.	387
Ниязова Р.С.	551	Хорошевский В.Ф.	233
Новогрудская Р.Л.	49	Худойбердиев Х.А.	383
Одинцова С.А.	449	Хусаинов А.Ф.	361
Омарбекова А.С.	547, 551	Шалфеева Е.А.	257
Орлова Ю.А.	341, 351	Шарипбаев А.А.	337, 547, 551
Пакладок Д.А.	375	Шпирко А.А.	417, 501
Палюх Б.В.	467	Шункевич Д.В.	269
Панкова Л.А.	225	Шустова Д.В.	449
Пархоменко В.А.	289	Щербак С.С.	103
Пекарь Д.В.	391	Яцкевич Н.И.	111
Петровский А.Б.	341, 497		
Пивоварчик О.В.	205		
Попова-Коварцева Д.А.	195		
Поттосин Ю.В.	573, 579		
Поттосина С.А.	573		
Прокопович Г.А.	483, 489		
Прокопчук Ю.А.	251		
Пронина В.А.	225		
Родченко В.Г.	295		
Розалиев В.Л.	357, 399		
Романов В.И.	473		
Русецкий К.В.	511		
Савельева М.Н.	413		
Садов В.С.	391		
Сидоркина И.Г.	333		
Сидорова Е.А.	159		
Скопинава А.М.	319		
Смирнов С.В.	191		
Соловьев В.И.	555		
Соловьев С.Ю.	147		
Солошенко А.Н.	351		
Солошич С.Н.	103		
Соснин П.И.	423		
Степура Л.В.	229		
Субхангулов Р.А.	219		
Сулейманов Д.Ш.	329, 361		
Суховеров В.С.	143		
Сычѳв В.А.	489		
Тагави С.А.	579		

AUTHOR INDEX

Abolmasov P.V.	195	Ivanchenko N.	437
Alekseev A.A.	153	Ivashenko V.P.	171
Alekseev A.V.	417	Kabush A.S.	477
Aliyev R.M.	395	Kastornov A.F.	461
Anishchenko I.S.	295	Kastornova V.A.	461
Anureev I.S.	43	Kazakov V.A.	533
Apanasovich V.V.	111	Kharlamov A.A.	369
Atuchin M.M.	43	Kheidorov I.	387
Barlybayev A.B.	547, 551	Khoroshevsky V. F.	233
Bekmanova G.T.	337	Khudoyberdiev Kh. A.	383
Belyakov S.L.	413	Khusainov A.F.	361
Belyakova M.L.	413	Kileev V.V.	333
Bibilo P.N.	473	Kintonova A.J.	551
Bobkov A.S.	497, 501	Kirillovich A.V.	91
Borgest N.M.	443, 445, 449	Kiselev V.V.	517
Borodaenko J.V.	107	Kleschev A.	25
Brancevich P.J.	569	Klyushin A. Yu.	309
Brek D.V.	375	Knyazihina Y.E.	449
Burdo G.B.	467	Kolb D.G.	115
Davydenko I.T.	185	Komartsova L.G.	285
Demin V.V.	477	Konstantinov V.M.	357
Dianov I.A.	357	Koronchik D. N.	125, 403
Dmitriev A.S.	345, 351	Korovin M.D.	441, 445
Dobrov B.V.	153	Kovartsev A.N.	195
Dodonov A.G.	457	Kozlov O.A.	529
Dorofeev N.S.	399, 501	Kryukov K.V.	143
Dunets A.P.	477	Kulik B.A.	165
Dunets I.P.	477	Kulinich A. A.	135
Efimenko I. V.	241	Kuznetsov O.P.	143
Filatova N.N.	433	Kuznetsov V.N.	299, 309
Filippov A.A.	219	Lande D.V.	457
Fridman A.Ya.	165	Lanin V.V.	97
Furman O.D.	115	Lapshov Y.A.	423
Galushka I.N.	103	Leichanka N.	561
Gatiatullin A. R.	329	Li.Y.	569
Globa L.	49	Lipnitsky S.F.	229
Golenkov V.V.	55	Lobanov B.M.	505
Golovko V.A.	477	Loukachevitch N.V.	153
Golovnja A.I.	521	Lyadova L.N.	97
Gorbachev N.N.	541	Majidnezhad V.	387
Grakova N.V.	453	Maklaev V.A.	423
Gribova V.	25	Malkovsky M.G.	147
Guliakina N.A.	55	Mamchich A.A.	229
Hetsevich Y.S.	319, 375, 489, 505	Massel A.G.	247
Ivanchenko G.	437	Massel L.V.	247

Melnikova V.V.	467	Subkhangulov R.A.	219
Mikhailov J.F.	529	Suhoverov V.S.	143
Morosanova N.A.	281	Suleymanov D.S.	329, 361
Mukanova A.S.	337	Sychou U.A.	489
Mutovkina N.Yu	309	Taghavi S.A.	579
Naidenova X.	289	Tatur M.M.	129
Namestnikov A.M.	219	Telnov Yu.F.	533
Nevzorova O.A.	91	Ternovoy M.,	49
Niyazova R.S.	551	Timchenko V.A.	81
Novogrudska R.	49	Trebukhin A.G.	433
Odintsova S.A.	449	Trembach V.M.	533
Omarbekova A.S.	547, 551	Vagapov D.R.	329
Orlova Y.A.	341, 351	Verenik N.L.	129
Pakladok D.A.	375	Vetrov Y.A.	407
Palyukh B.V.	467	Vinogradov G.P.	299
Pankova L.A.	225	Vishniakou U.A.	107
Parkhomenko V.	289	Yatskevich M.I.	111
Pekar D. V.	391	Yeliseyeva O.E.	511, 517
Petrovsky A.B.	341, 497	Yergesh B.Zh.	337
Pivovarchyk O.V.	205	Yermolenko T.V.	369
Popova-Kovartseva D.A.	195	Zaboleeva-Zotova A.V.	341, 345, 399, 497
Pottosin Yu.V.	573, 579	Zagorulko G.B.	265
Pottosina S.A.	573	Zagorulko M.Yu.	159
Prakapovich R.A.	483, 489	Zagorulko Yu.A.	31, 265
Prokopchuk I.A.	251	Zavgorodniy V.V.	103
Pronina V.A.	225	Zhidchenko V.V.	195
Rodchenko V.G.	295	Zhilyakova L.Yu.	87
Romanov V.I.	473	Zhitko V.A.	505
Rozaliev V.L.	357, 399	Zhukau I.I.	453
Rusetski K.V.	511	Zhukevich A.I.	295
Sadov V. S.	391	Zuenko A.A.	165
Savelyeva M.N.	413		
Shalfeeva E.	257		
Sharipbayev A.A.	337, 547, 551		
Shcherbak S.S.	103		
Shpirko A.A.	417, 501		
Shunkevich D.V.	269		
Shustova D.V.	449		
Sidorkina I.G.	333		
Sidorova E.A.	159		
Skopinava A.M.	319		
Smirnov S.V.	191		
Soloshenko A.N.	351		
Soloshish S.N.	103		
Soloviev S.Y.	191		
Soloviev V.I.	555		
Sosnin P.I.	423		
Stepura L.V.	229		

Научное издание

**ОТКРЫТЫЕ СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ
ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ**

OSTIS-2013

OPEN SEMANTIC TECHNOLOGIES FOR INTELLIGENT SYSTEMS

Материалы

III Международной научно-технической конференции

(Минск, 21 – 23 февраля 2013 года)

В авторской редакции

Ответственный за выпуск *В. В. Голенков*
Компьютерная верстка *Н. В. Гракова*

Подписано в печать 13.02.2013. Формат 60×84 1/8. Бумага офсетная. Гарнитура «Гаймс».
Отпечатано на ризографе. Усл. печ. л. . Уч.-изд. л. . Тираж 120 экз. Заказ 42.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0494371 от 16.03.2009. ЛП №02330/0494175 от 03.04.2009.
220013, Минск, П. Бровки, 6

http://ostis.net



OSTIS12

(Open Semantic Technologies for Intelligent Systems)
(Open Semantic Technology for Intelligent Systems)

Это открытый проект, направленный на создание массовой семантической технологии компонентного проектирования интеллектуальных систем различного назначения.

Цели проекта OSTIS

- Создать массовую, комплексную и активно развивающуюся технологию проектирования интеллектуальных систем, включающую в себя теоретические и практические, программные и аппаратные аспекты
- Создать инфраструктуру, обеспечивающую сочетание научной и учебной, инженерной и коммерческой деятельности в области искусственного интеллекта

Особенности проекта OSTIS

- Является открытым комплексным проектом, состоит из большого числа частных проектов и предоставляет полный пакет документации по всем компонентам предлагаемой технологии (включая исходные тексты соответствующих программных средств)
- Ориентирован на широкий контингент разработчиков прикладных интеллектуальных систем (на массовое распространение предлагаемой технологии)
- Ориентирован на существенное сокращение сроков проектирования интеллектуальных систем

ПРИГЛАШАЕМ СТАТЬ

УЧАСТНИКАМИ НАШЕГО ПРОЕКТА

Проект OSTIS заинтересован в существенном и постоянном расширении контингента участников. Любой желающий может войти в число участников нашего проекта. Для этого вам необходимо зарегистрироваться на сайте <http://ostis.net> и сообщить конкретную область своих интересов.



OSTIS-2014

IV Международная научно-техническая конференция «Открытые семантические технологии проектирования интеллектуальных систем»

Open Semantic Technologies for Intelligent Systems

20 – 22 февраля 2014 г. Минск. Республика Беларусь

ИНФОРМАЦИОННОЕ СООБЩЕНИЕ

Приглашаем принять участие в IV Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2014).

Конференция пройдет в период с **20 по 22 февраля 2014** года в Белорусском государственном университете информатики и радиоэлектроники, г. Минск, Республика Беларусь.

Рабочие языки конференции: русский, белорусский, английский.

ОСНОВНЫЕ ОРГАНИЗАТОРЫ КОНФЕРЕНЦИИ

- Российская ассоциация искусственного интеллекта (РАИИ)
- Белорусский государственный университет информатики и радиоэлектроники (БГУИР)
- Государственное учреждение «Администрация Парка высоких технологий» (Республика Беларусь)
- Объединённый институт проблем информатики Национальной академии наук Беларуси (ОИПИ НАН Беларуси)
- Тверской государственный технический университет
- Научно-исследовательский институт «Прикладная семиотика» АН РТ

ПРОГРАММНЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

Боргест Н.М., к.т.н., доц., РФ
Гаврилова Т.А., д.т.н., проф., РФ
Глоба Л.С., д.т.н., проф., Украина
Голенков В.В., д.т.н., проф., РБ
Головко В.А., д.т.н., проф., РБ
Гордей А.Н., д. фил. н., проф., РБ
Грибова В.В., д.т.н., РФ
Гулякина Н.А., к.ф.-м.н., доц., РБ
Еремеев А.П., д.т.н., проф., РФ
Ефименко И.В., к. фил. н., РФ
Заболеева-Зотова А.В., д.т.н., РФ
Загорюлько Ю.А., к.т.н., доц., РФ
Клешев А.С., д.т.н., проф., РФ
Кобринский Б.А., д.мед.н., РФ
Козлов О.А., д.п.н., проф., РФ
Комарцова Л.Г., д.т.н., РФ
Кузнецов О.П., д.т.н., проф., РФ
Курейчик В.М., д.т.н., проф., РФ
Лобанов Б.М., д.т.н., проф., РБ
Найденова К.А., к.т.н., РФ

Невзорова О.А., к.т.н., доцент, РФ
Осипов Г.С., д.ф.-м.н., проф., РФ
Палюх Б.В., д.т.н., проф., РФ
Петровский А.А., д.т.н., проф., РБ
Плесневич Г.С., к.ф.-м.н., РФ
Родченко В.Г., к.т.н., доц., РБ
Рычкова Л.В., к. фил. н., доц., РБ
Смирнов С.В., д.т.н., проф., РФ
Соловьёв С.Ю., д.ф.-м.н., проф., РФ
Соснин П.И., д.т.н., проф., РФ
Сулейманов Д.Ш., академик АН Татарстана, РФ
Тарасов В.Б., к.т.н., доц., РФ
Тельнов Ю.Ф., д.э.н., проф., РФ
Тузиков А.В., д.ф.-м.н., проф., РБ
Харламов А.А., д.т.н., РФ
Хорошевский В.Ф., д.т.н., проф., РФ
Чернявский А.Ф., академик НАН Беларуси
Шарипбаев А.А., д.т.н., проф., Казахстан
Щербак С.С., к.т.н., доц., Украина

СТАТЬИ НА КОНФЕРЕНЦИЮ БУДУТ ПРИНИМАТЬСЯ ПО СЛЕДУЮЩИМ НАПРАВЛЕНИЯМ:

1. *Принципы построения и структура технологии компонентного проектирования интеллектуальных систем*
2. *Базовые семантические модели интеллектуальных систем и их реализация*
3. *Семантические модели, средства и методы компонентного проектирования баз знаний и пакетов программ, ориентированных на обработку знаний*
4. *Семантические модели, средства и методы компонентного проектирования решателей задач*
5. *Семантические модели, средства и методы компонентного проектирования пользовательских интерфейсов интеллектуальных систем*
6. *Комплексные методы, онтология и менеджмент компонентного проектирования интеллектуальных систем*
7. *Логико-семантические модели прикладных интеллектуальных систем*

ЦЕЛЬ И ФОРМАТ ПРОВЕДЕНИЯ КОНФЕРЕНЦИИ

Целью конференции является обсуждение проблем создания открытой комплексной технологии компонентного проектирования интеллектуальных систем. Этим определяется и формат её проведения, предполагающий (1) ориентацию на круглые столы, посвященные обсуждению различных вопросов создания указанной технологии, (2) обеспечение возможности всем авторам не только выступить с докладами, но и продемонстрировать свои результаты на выставочных стендах.

Важнейшей задачей конференции является привлечение к её работе не только учёных и аспирантов, но и студенческой молодежи, интересующейся проблемами искусственного интеллекта, а также коммерческих организаций, готовых сотрудничать с научными коллективами, работающими над созданием современных технологий проектирования интеллектуальных систем.

ПОРЯДОК ПРЕДСТАВЛЕНИЯ СТАТЕЙ

Статьи (только по перечисленным выше направлениям) представляются в готовом для публикации виде. Текст статьи должен быть логически законченным и содержать новые научные и практические результаты.

Крайний срок получения статей Оргкомитетом – *1 декабря 2013.*

Минимальный объем статьи, включая иллюстрации, библиографический список и заключительную аннотацию (англоязычную аннотацию, в случае русскоязычной или белорусской статьи или русскоязычную аннотацию, в случае англоязычной статьи) – 4 полностью заполненные страницы. Статьи, занимающие объем более 6 страниц подлежат согласованию с Программным комитетом.

Переписка с авторами будет вестись только по электронной почте. Адрес электронной почты Оргкомитета: ostisconf@gmail.com. Для переписки необходимо зарегистрироваться на сайте конференции <http://conf.ostis.net>.

Шаблон оформления статей размещен на сайте конференции.

ФОРМИРОВАНИЕ ПРОГРАММЫ КОНФЕРЕНЦИИ

Программа конференции формируется Программным комитетом по результатам рецензирования, представленных статей.

ПУБЛИКАЦИЯ МАТЕРИАЛОВ КОНФЕРЕНЦИИ

Оргкомитет конференции предполагает публикацию отобранных Программным комитетом статей в Сборнике материалов конференции и на официальном сайте конференции <http://conf.ostis.net>. Неимущественные права принадлежат авторам статей, поэтому публикация и распространение материалов статей на иных информационных ресурсах допускается только с согласия авторов статей.

УСЛОВИЯ УЧАСТИЯ В КОНФЕРЕНЦИИ

Участие в конференции не предполагает организационного взноса.

СВЯЗЬ С ОРГАНИЗАТОРАМИ КОНФЕРЕНЦИИ

Сайт: <http://conf.ostis.net>

Email: ostisconf@gmail.com