

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ
АДМИНИСТРАЦИЯ ПАРКА ВЫСОКИХ ТЕХНОЛОГИЙ

**Открытые семантические технологии
проектирования интеллектуальных систем**

OSTIS-2012

Open Semantic Technologies for Intelligent Systems

**МАТЕРИАЛЫ
II МЕЖДУНАРОДНОЙ
НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ**

(Минск, 16 – 18 февраля 2012 года)

Минск
БГУИР
2012

УДК 004.8

ББК 32.0813

О-83

Редакционная коллегия:

В. В. Голенков (отв. ред.), Т. А. Гаврилова, Л. С. Глоба, В. А. Головки, В. В. Грибова, Н. А. Гулякина, А. П. Еремеев, П. П. Ермолов, И. В. Ефименко, А. В. Заболеева-Зотова, Ю. А. Загорюлько, А. В. Зубов, А. А. Иванов, В. С. Ижуткин, А. С. Клецев, Б. А. Кобринский, О. А. Козлов, Л. Г. Комарцова, О. П. Кузнецов, С. О. 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, И. В. Роберт, В. Г. Родченко, В. С. Смородин, П. И. Соснин, В. Л. Стефанюк, Д. Ш. Сулейманов, В. Б. Тарасов, А. А. Харламов, И. Э. Хейдоров, В. Ф. Хорошевский, А. Ф. Чернявский, С. С. Щербак

Организаторы:

Белорусский государственный университет информатики и радиоэлектроники
Российская ассоциация искусственного интеллекта
Государственное учреждение "Администрация Парка высоких технологий" (Республика Беларусь)
Институт информатизации образования Российской академии образования
Компания "Мелсофт"
Компания "Речевые Технологии"
Компания "СКЭНД"
Компания "Эксиджен Сервисис"
Компания "Дженерэйшн-Пи Консалтинг"
Компания "Итранзишэн"
Компания "ВирусБлокАда"

Organized:

Belarusian State University of Informatics and Radioelectronics
Russian Association of Artificial Intelligence
Belarus Hi-Tech Park Administration
Institution Of The Russian Academy Of Education "Institute Of Informatization Of Education"
Melsoft
Speech Technology Ltd.
Scand Ltd.
Exigen Services Ltd.
Generation_P Consulting Ltd.
Itransition Ltd.
VirusBlokAda Ltd.

Техническая и информационная поддержка:

Журнал "ЭЛЕКТРОНИКА инфо"
Научно-практический журнал "Речевые технологии"
Международный журнал "Программные продукты и системы"
Открытый проект "Ogre"
Открытый проект "MyGui"

Technical Co-Sponsorship:

"Electronics info" Magazine
"Speech Technology" Journal
"SOFTWARE & SYSTEMS" Journal
Open Source Project "Ogre"
Open Source Project "MyGui"

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

**Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2012): материалы II Междунар. научн.-техн. конф. (Минск, 16-18 февраля 2012 г.) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2012. – 548 с.
ISBN 978-985-488-683-1.**

Сборник включает прошедшие рецензирование доклады II 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-2012 и печатаются в виде, представленном авторами.

**УДК 004.8
ББК 32.0813**

ISBN 978-985-488-683-1

© УО «Белорусский государственный университет информатики и радиоэлектроники», 2012

ПРОГРАММНЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

- Кузнецов О.П., д.т.н., проф., ИПУ РАН, г. Москва – председатель
- Гаврилова Т.А., д.т.н., проф., Высшая Школа менеджмента СПбГУ, г. Санкт-Петербург
- Глоба Л.С., д.т.н., проф., КПИ, г. Киев
- Голенков В.В., д.т.н., проф., БГУИР, г. Минск
- Головкин В.А., д.т.н., проф., БрГТУ, г. Брест
- Грибова В.В., д.т.н., ИАПУ ДВО РАН, г. Владивосток
- Гулякина Н.А., к.ф.-м.н., доц., БГУИР, г. Минск
- Еремеев А.П., д.т.н., проф., МЭИ ТУ, г. Москва
- Ермолов П.П., к.т.н., СевНТУ, г. Севастополь
- Ефименко И.В., к. фил. н., ГУ-ВШЭ, г. Москва
- Заболеева-Зотова А.В., д.т.н., проф., ВолгГТУ, г. Волгоград
- Загоруйко Ю.А., к.т.н., доц., Институт систем информатики имени А.П.Ершова СО РАН, г. Новосибирск
- Зубов А. В., д. фил. н., проф., МГЛУ, г. Минск
- Иванюк А.А., д.т.н., доц., БГУИР, г. Минск
- Ижуткин В.С., д.ф.-м.н., проф., НИУ(МЭИ), г. Москва
- Клещев А.С., д.т.н., проф., ИАПУ ДВО РАН, г. Владивосток
- Кобринский Б.А., д.мед.н., проф., РНИМУ им. Н.И.Пирогова, г. Москва
- Козлов О.А., д.п.н., проф., ИИО РАО, г. Москва
- Комарцова Л.Г., д.т.н., проф., Калужский филиал МГТУ им. Н.Э. Баумана, г. Калуга
- Кузнецов С.О., д.ф.-м.н., проф., ГУ-ВШЭ, МФТИ, г. Москва
- Курейчик В.М., д.т.н., проф., ТТИ ЮФУ, г. Таганрог
- Лобанов Б.М., д.т.н., проф., ОИПИ, г. Минск
- Найденова К.А., к.т.н., ВМА, г. Санкт-Петербург
- Невзорова О.А., к.т.н., доцент, ИПС АН РТ, г. Казань
- Осипов Г.С., д.ф.-м.н., проф., ИСА РАН, г. Москва
- Палюх Б.В., д.т.н., проф., ТГТУ, г. Тверь
- Петровский А.А., д.т.н., проф., БГУИР, г. Минск
- Плесневич Г.С., к.ф.-м.н., проф. МЭИ, г. Москва
- Роберт И.В., д.п.н., проф., ИИО РАН, г. Москва
- Родченко В.Г., к.т.н., доц., ГрГУ им. Я. Купалы, г. Гродно
- Смородин В.С., д.т.н., ГГУ им. Ф. Скорины, г. Гомель
- Соснин П.И., д.т.н., проф., УГТУ, г. Ульяновск
- Стефанюк В.Л. д.т.н., проф., ИППИ, г. Москва
- Сулейманов Д.Ш., академик АН Татарстана, ИПС АН Татарстана, г. Казань
- Тарасов В.Б., к.т.н., доц., МГТУ им. Н. Э. Баумана, г. Москва
- Харламов А.А., д.т.н, ИВНД и НФ РАН, г. Москва
- Хейдоров И.Э., к.ф.-м.н., доц., БГУ г. Минск
- Хорошевский В.Ф., д.т.н., проф., ВЦ РАН, г. Москва
- Чернявский А.Ф., академик НАН Беларуси, д.т.н., проф., БГУ, г. Минск
- Щербак С.С., к.т.н., доц., КНУ им. М.Остроградского, г. Кременчуг

PROGRAM COMMITTEE

- **Kuznetsov O.** d.of t.s, RF proff., **chair**
- **Chernyavskiy A.** NSA of Belarus academician, d.of t.s., proff.
- **Gavrilova T.** d.of t.s., RF proff.
- **Globo L.** d.of t.s, Ukraine proff.
- **Golenkov V.** d.of t.s., RB proff.
- **Golovko V.** d.of t.s., RB proff.
- **Gribova V.** d.of t.s. RF
- **Gulyakina N.** c.of ph.-m.s., RB ass.proff.
- **Eremeev A.** d.of t.s, RF proff.
- **Ermolov P.** c.of t.s., Ukraine ass.proff.
- **Harlamov A.** d.of t.s., RF
- **Heydorov I.** c.of ph.-m.s., RB
- **Horoshevskiy V.** d.of t.s, RF proff.
- **Ivanuk A.** d.of t.s, RB ass.proff.
- **Izhutkin V.** d.of ph.-m.s., RF proff.
- **Kleshev A.** d.of t.s, RF proff.
- **Kobriniski B.** d.of med.s, RF proff.
- **Komartsova L.** d.of t.s, RF proff.
- **Kozlov O.** d.of teach.s, RF proff.
- **Kureychik V.** d.of t.s, RF proff.
- **Kuznetsov S.** d.of ph.-m., RF proff.
- **Lobanov B.** d.of t.s, RB proff.
- **Naydenova K.** c.of t.s, RF
- **Nevzorova O.** c.of t.s, RF ass.proff.
- **Osipov G.** c.of ph.-m.s., RF proff.
- **Palyuh B.** d.of t.s, RF proff.
- **Petrovskiy A.** d.of t.s, RB proff.
- **Plesnevich G.** c.of ph.-m.s., RF proff.
- **Robert I.** d.of teach.s, RF proff.
- **Rodchenko V.** c.of t.s., RB ass.proff.
- **Scherbak S.** c.of t.s., Ukraine ass.proff.
- **Smorodin V.** d.of t.s, RB
- **Sosnin P.** d.of t.s, RF proff.
- **Stefanyuk V.** d.of t.s, RF proff.
- **Suleymanov D.** SA of Tatarstan academician, d.of t.s, RF proff.
- **Tarasov V.** c.of t.s, RF ass.proff.
- **Yefimenko I.** c.of phyl.s, RF
- **Zaboleeva-Zotova A.** d.of t.s, RF proff.
- **Zagorulko Y.** c.of t.s, RF ass.proff.
- **Zubov A.** d.of phyl.s, RB proff.

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
- **Томилин Владимир Федорович,**
заместитель проректора Белорусского государственного университета информатики и радиоэлектроники по НИЧ
- **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
- **Leaid 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
- **Vladimir Tomilin,**
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	19
СЕКЦИЯ 1. БАЗОВЫЕ УНИВЕРСАЛЬНЫЕ СЕМАНТИЧЕСКИЕ МОДЕЛИ ПРЕДСТАВЛЕНИЯ И ОБРАБОТКИ ЗНАНИЙ И ИХ ПРОГРАММНАЯ И АППАРАТНАЯ РЕАЛИЗАЦИЯ	21
В.В. Голенков, Н.А. Гулякина ГРАФОДИНАМИЧЕСКИЕ МОДЕЛИ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ЗНАНИЙ: ПРИНЦИПЫ ПОСТРОЕНИЯ, РЕАЛИЗАЦИИ И ПРОЕКТИРОВАНИЯ	23
Б.А. Кобринский ОБРАЗНЫЕ ПРЕДСТАВЛЕНИЯ СПЕЦИАЛИСТА И ПРОБЛЕМА ИХ ОТРАЖЕНИЯ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ	53
В.А. Тимченко МОДЕЛЬ КЛАССОВ СЕМАНТИЧЕСКИХ СЕТЕЙ И ИХ ПРЕОБРАЗОВАНИЙ	63
Л.Ю. Жиликова ДИСКРЕТНЫЕ МОДЕЛИ РАССЕЯНИЯ НА ГРАФАХ	71
М.Г. Мальковский, С.Ю. Соловьев ТЕРМИНОЛОГИЧЕСКИЕ СЕТИ	77
В.И. Бодякин АВТОМАТИЧЕСКОЕ ПОСТРОЕНИЕ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ПРОИЗВОЛЬНЫХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ НА БАЗЕ НЕЙРОСЕМАНТИЧЕСКОГО ПОДХОДА	83
А.Е. Баранович, Н.О. Никитин О НЕКОТОРЫХ ОБЛАСТЯХ ПРИЛОЖЕНИЙ АЛГЕБРАИЧЕСКОЙ МОДЕЛИ К- ГИПЕРПРОСТРАНСТВА СХ-ГИПЕРТОПОГРАФОВ	91
Л.С. Глоба, М.Ю. Терновой, Е.С. Штогрин ПОДХОД К ХРАНЕНИЮ БАЗ НЕЧЕТКИХ ЗНАНИЙ	99
Ю.Н. Молчанов, Л.С. Глоба, Н.А. Алексеев ДЕТЕКТИРОВАНИЕ ИЗМЕНЕНИЙ В INTERNET-ДОКУМЕНТАХ	103
Д.Г. Колб WEB-ОРИЕНТИРОВАННАЯ РЕАЛИЗАЦИЯ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	111
А.А. Иванюк СОВРЕМЕННОЕ СОСТОЯНИЕ И ТЕНДЕНЦИИ РАЗВИТИЯ ТЕХНОЛОГИЙ РЕКОНФИГУРИРУЕМЫХ ЦИФРОВЫХ УСТРОЙСТВ	123
Н.В. Абрамов, А.А. Иванюк РЕКОНФИГУРИРУЕМЫЕ ВЫЧИСЛЕНИЯ КАК СРЕДСТВО ОПТИМИЗАЦИИ РЕШЕНИЙ ПОДКЛАССА ЗАДАЧ	129
С.А. Байрак, Д.Н. Одинец, М.М. Татур, Ф. Филипов, Марио Мунос ПАРАЛЛЕЛЬНЫЕ ПРОЦЕССОРЫ ДЛЯ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	135

СЕКЦИЯ 2. СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ, ПРОГРАММ И ПАКЕТОВ ПРОГРАММ, ОРИЕНТИРОВАННЫХ НА ОБРАБОТКУ ЗНАНИЙ	141
В.Ф. Хорошевский СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ: ОЖИДАНИЯ И ТРЕНДЫ	143
О.П. Кузнецов О ВОЗМОЖНОСТИ ОРГАНИЗАЦИИ ЗНАНИЙ НА ОСНОВЕ КОГНИТИВНОЙ СЕМАНТИКИ	159
Г.С. Плесневич ФОРМАЛЬНЫЕ ОНТОЛОГИИ	163
А.П. Еремеев, А.А. Еремеев ИНТЕГРАЦИЯ АППАРАТА ХРАНИЛИЩ ДАННЫХ И ТЕМПОРАЛЬНЫХ МОДЕЛЕЙ	169
А.А. Горкина ИНТЕГРАЦИЯ ЛОГИКИ ДВИЖЕНИЯ И БИНАРНОЙ МОДЕЛИ ДАННЫХ И ЗНАНИЙ	175
Ю.А. Загоруйко, О.И. Боровикова, Г.Б. Загоруйко ПОСТРОЕНИЕ МНОГОЯЗЫЧНЫХ ТЕЗАУРУСОВ СРЕДСТВАМИ СЕМАНТИЧЕСКОЙ ТЕХНОЛОГИИ	181
Н.А. Моросанова ДОСТОВЕРНОСТЬ ПРАВИЛ В ЭКСПЕРТНЫХ СИСТЕМАХ	189
В.П. Ивашенко СЕМАНТИЧЕСКИЕ МОДЕЛИ И СРЕДСТВА ИНТЕГРАЦИИ И ОТЛАДКИ БАЗ ЗНАНИЙ	193
В.В. Диковицкий ИЗВЛЕЧЕНИЕ ЗНАНИЙ ПОЛЬЗОВАТЕЛЯ И ВЕРИФИКАЦИЯ ЗНАНИЙ САМООРГАНИЗУЮЩИХСЯ ИНФОРМАЦИОННЫХ СИСТЕМ С ОБРАТНОЙ СВЯЗЬЮ	205
В.А. Катаев РАЗРАБОТКА ЭКСПЕРТНЫХ СИСТЕМ В СРЕДЕ MULTISTUDIO	207
В.В. Грибова, А.С. Клещев ОНТОЛОГИЧЕСКАЯ ПАРАДИГМА ПРОГРАММИРОВАНИЯ	213
Н.А. Гулякина, О.В. Пивоварчик, Д.А. Лазуркин ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ, ОРИЕНТИРОВАННЫЕ НА ОБРАБОТКУ СЕМАНТИЧЕСКИ СЕТЕЙ	221
И.С. Волегов, Е.Б. Замятина ОНТОЛОГИЧЕСКИЙ ПОДХОД К ИНТЕГРАЦИИ КОМПОНЕНТОВ ИМИТАЦИОННОЙ МОДЕЛИ В TRIAD.NET	229

СЕКЦИЯ 3. СЕМАНТИЧЕСКИЕ МОДЕЛИ ИНФОРМАЦИОННОГО ПОИСКА И РЕШЕНИЯ ЗАДАЧ. СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННО-ПОИСКОВЫХ МАШИН И РЕШАТЕЛЕЙ ЗАДАЧ	237
<hr/>	
Ю.В. Рогошина СЕМАНТИЧЕСКИЙ ПОИСК КАК СОСТАВЛЯЮЩАЯ УПРАВЛЕНИЯ ЗНАНИЯМИ В SEMANTICWEB	239
<hr/>	
Е.В. Биряльцев, М.Р. Галимов, Н.Г. Жильцов, О.А. Невзорова ПОДХОД К СЕМАНТИЧЕСКОМУ ПОИСКУ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ В НАУЧНЫХ ТЕКСТАХ	245
<hr/>	
В.С. Дюндюков, В.Б. Тарасов 3.3. ФОРМИРОВАНИЕ МНОГОАГЕНТНЫХ СИСТЕМ С ПОМОЩЬЮ РЕСУРСНО-ЦЕЛЕВЫХ СЕТЕЙ	257
<hr/>	
В.Б. Тарасов, А.П. Калуцкая, М.Н. Святкина ГРАНУЛЯРНЫЕ, НЕЧЕТКИЕ И ЛИНГВИСТИЧЕСКИЕ ОНТОЛОГИИ ДЛЯ ОБЕСПЕЧЕНИЯ ВЗАИМОПОНИМАНИЯ МЕЖДУ КОГНИТИВНЫМИ АГЕНТАМИ	267
<hr/>	
О.О. Варламов МИВАРНЫЕ ТЕХНОЛОГИИ СОЗДАНИЯ ЛОГИЧЕСКОГО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА: ЛОГИЧЕСКАЯ ОБРАБОТКА С ЛИНЕЙНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТЬЮ БОЛЕЕ 3 МИЛЛИОНОВ ПРОДУКЦИЙ И ВОЗМОЖНОСТЬ ПОНИМАНИЯ СМЫСЛА ЧЕРЕЗ МОДЕЛИРОВАНИЕ БОЛЬШОГО КОНТЕКСТА	279
<hr/>	
Ю.М. Сметанин ВЕРоятностная логика и ортогональный базис силлогистики	289
<hr/>	
С.С. Заливако, Д.В. Шункевич СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ РЕШАТЕЛЕЙ ЗАДАЧ	297
<hr/>	
Л.Г. Комарцова, Ю.Н. Лавренков, О.В. Антипова, Д.С. Кадников ДИНАМИЧЕСКИЕ МОДЕЛИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ АДАПТИВНОГО РЕЗОНАНСА	315
<hr/>	
А.И. Жукевич, Е.В. Олизарович, В.Г. Родченко ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЙ ПРИ ПОСТРОЕНИИ СИСТЕМ РАСПОЗНОВАНИЯ ОБРАЗОВ	321

СЕКЦИЯ 4. СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ МУЛЬТИМОДАЛЬНЫХ И ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ИНТЕРФЕЙСОВ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	325
<hr/>	
С.С. Курбатов, А.П. Лобзин, К.А. Найденова, Г.К. Хахалин ГИБРИДНАЯ СХЕМА АНАЛИЗА ИЗОБРАЖЕНИЙ	327
<hr/>	
Н.М. Боргест, Р.В. Чернов, Д.В. Шустова РАЗРАБОТКА ИНТЕРФЕЙСА ИНТЕЛЛЕКТУАЛЬНОГО ПОМОЩНИКА ПРОЕКТАНТА	335
<hr/>	
Д.Н. Корончик СЕМАНТИЧЕСКИЕ МОДЕЛИ МУЛЬТИМОДАЛЬНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ И СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ ИХ ПРОЕКТИРОВАНИЯ	339
<hr/>	
А.В. Заболеева-Зотова, Ю.А. Орлова, В.Л. Розалиев, А.С. Бобков ЗАДАЧА СОЗДАНИЯ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО РАСПОЗНАВАНИЯ ЭМОЦИЙ	347
<hr/>	
А.В. Заболеева-Зотова, Ю.А. Орлова, В.Л. Розалиев, А.С. Бобков РЕШЕНИЕ ЗАДАЧИ ПРЕДСТАВЛЕНИЯ ДВИЖЕНИЙ В ВИДЕ НЕЧЕТКИХ ТЕМПОРАЛЬНЫХ ВЫСКАЗЫВАНИЙ ДЛЯ ОПРЕДЕЛЕНИЯ ЭМОЦИОНАЛЬНЫХ РЕАКЦИЙ ЧЕЛОВЕКА	351
<hr/>	
В.В. Киселёв, А.Г. Давыдов, А.В. Ткаченя СИСТЕМА ОПРЕДЕЛЕНИЯ ЭМОЦИОНАЛЬНОГО СОСТОЯНИЯ ДИКТОРА ПО ГОЛОСУ	355
<hr/>	
А.Г. Киселёва, Г.Д. Киселёв ОБРАБОТКА КОНТЕКСТНЫХ ДАННЫХ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ	359
<hr/>	
Е.Б. Козеренко КОГНИТИВНО-ЛИНГВИСТИЧЕСКИЕ ПРЕДСТАВЛЕНИЯ ЯЗЫКОВЫХ СТРУКТУР В СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И МАШИННОГО ПЕРЕВОДА	365
<hr/>	
И.П. Кузнецов, М.М. Шарнин, Е.Б. Козеренко, А.Г. Мацкевич, В.Г. Николаев, Н.В. Сомин ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ИЗВЛЕЧЕНИЯ СУЩНОСТЕЙ, ОСНОВАННЫЕ НА РАСШИРЕННЫХ СЕМАНТИЧЕСКИХ СЕТЯХ	373
<hr/>	
И.В. Ефименко ОНТОЛОГИЯ АВТОРИТЕТА: МЕХАНИЗМЫ АПЕЛЛЯЦИИ К ЧУЖИМ МНЕНИЯМ В ТЕКСТАХ РАЗЛИЧНЫХ ЖАНРОВ	381
<hr/>	

А.А. Харламов, Т.В. Ермоленко АВТОМАТИЧЕСКОЕ ФОРМИРОВАНИЕ НЕОДНОРОДНОЙ СЕМАНТИЧЕСКОЙ СЕТИ НА ОСНОВЕ ВЫЯВЛЕНИЯ КЛЮЧЕВЫХ ПРЕДИКАТНЫХ СТРУКТУР ПРЕДЛОЖЕНИЙ ТЕКСТА	385
Д.Ш. Сулейманов, А.Р. Гатиатуллин СЕМАНТИЧЕСКИЕ УНИВЕРСАЛИИ В СИСТЕМЕ ОПИСАНИЯ ЗНАЧЕНИЙ МОРФЕМ	391
А.А. Шарипбаев, Г.Т. Бекманова, Б.Ж. Ергеш, А.К. Бурибаева, М.Х. Карабалаева ИНТЕЛЛЕКТУАЛЬНЫЙ МОРФОЛОГИЧЕСКИЙ АНАЛИЗАТОР, ОСНОВАННЫЙ НА СЕМАНТИЧЕСКИХ СЕТЯХ	397
С.А. Гецевич, Ю.С. Гецевич, О.Е. Елисеева, В.А. Житко, А.А. Кузьмин СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ БЕЛОРУССКО- И РУССКОЯЗЫЧНЫХ ЕЯ-ИНТЕРФЕЙСОВ ВОПРОСНО-ОТВЕТНЫХ СИСТЕМ	401
Ю.А. Кім, В.Я. Елісеєва ФАРМАЛІЗАЦЫЯ СЕМАНТЫКІ ПРАГМАТЫЧНЫХ ФРАЗЕМ БЕЛАРУСКАЙ МОВЫ	413
А.И. Головня НЕКОТОРЫЕ АСПЕКТЫ СЕМАНТИЧЕСКОГО КОДИРОВАНИЯ: СПОСОБЫ ВЫРАЖЕНИЯ ПРИЧИННОСТИ В ЯЗЫКЕ-СИСТЕМЕ	419
СЕКЦИЯ 5. МЕТОДОЛОГИИ И МЕНЕДЖМЕНТ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ НА ОСНОВЕ СЕМАНТИЧЕСКИХ ТЕХНОЛОГИЙ В РАМКАХ OPEN SOURCE ПРОЕКТОВ	425
В.А. Маклаев, П.И. Соснин ПРЕЦЕДЕНТНО-ОРИЕНТИРОВАННАЯ БАЗА ОПЫТА ПРОЕКТНОЙ ОРГАНИЗАЦИИ	427
А.С. Клещев, Е.А. Шалфеева ОНТОЛОГИЯ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ	437
Т.А. Гаврилова, Н.Е. Мельников ОБ ОСОБЕННОСТЯХ УПРАВЛЕНИЯ ЗНАНИЯМИ В РОССИЙСКИХ СОФТВЕРНЫХ КОМПАНИЯХ	443
Л.С. Глоба, Р.Л. Новогрудская МОДЕЛИ И МЕТОДЫ ИНТЕГРАЦИИ ИНФОРМАЦИОННЫХ И ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ	447
В.В. Бахтизин, С.Н. Неборский ОЦЕНКА КАЧЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ	453
И.Т. Давыденко КОМПЛЕКСНАЯ МЕТОДИКА ПРОЕКТИРОВАНИЯ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ИНТЕЛЛЕКТУАЛЬНЫХ СПРАВОЧНЫХ СИСТЕМ	457

СЕКЦИЯ 6. ПРИКЛАДНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ, ОСНОВАННЫЕ НА СЕМАНТИЧЕСКИХ СЕТЯХ	467
Ю.И. Майборода МЕТОДИКА РЕШЕНИЯ ЗАДАЧИ ЧАСТОТНО-ТЕРРИТОРИАЛЬНОГО ПЛАНИРОВАНИЯ НА ОСНОВЕ ПРИМЕНЕНИЯ ПРОДУКЦИОННЫХ ЭКСПЕРТНЫХ СИСТЕМ И ЭВРИСТИЧЕСКИХ АЛГОРИТМОВ	469
А.С. Клещев, М.Ю. Черняховская, Е.А. Шалфеева АНАЛИЗ МЕДИЦИНСКОЙ ИНТЕЛЛЕКТУАЛЬНОЙ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ С ТОЧКИ ЗРЕНИЯ ВОЗМОЖНОСТЕЙ ЕЕ АВТОМАТИЗАЦИИ	473
Ю.М. Сметанин, М.Ю. Сметанин. МЕДИЦИНСКАЯ ДИАГНОСТИКА И ОРТОГОНАЛЬНЫЙ БАЗИС СИЛЛОГИСТИКИ	479
А.И. Вовк, Ю.Я. Рубан, Д.А. Гирнык. РАЗРАБОТКА СИСТЕМ ТЕСТИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ WEB- ОРИЕНТИРОВАННОЙ НОТАЦИИ МАТЕМАТИЧЕСКИХ ТЕКСТОВ, СОХРАНЯЮЩЕЙ СЕМАНТИКУ	489
А.Я. Гладун, Ю.В. Рогошина. КОГНИТИВНЫЕ СЕТИ И ОНТОЛОГИЧЕСКИЙ АНАЛИЗ В ПОВЫШЕНИИ АДАПТИВНОСТИ И КАЧЕСТВА ОБСЛУЖИВАНИЯ В ГЕТЕРОГЕННОЙ БЕСПРОВОДНОЙ СРЕДЕ	493
Л.С. Глоба, И.А. Кузин, К.С. Мочалкина, Р.Л. Новогрудская. МОДЕЛЬ ИНТЕРНЕТ-ПОРТАЛА НАЦИОНАЛЬНОГО ЦЕНТРА АНТАРКТИЧЕСКИХ ДАННЫХ	501
О.А. Козлов, Ю.Ф. Михайлов ПОСТРОЕНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ФОРМИРОВАНИЯ ИНДИВИДУАЛЬНОЙ ТРАЕКТОРИИ ИЗУЧЕНИЯ СТУДЕНТОМ НЕКОТОРОЙ ПРЕДМЕТНОЙ ОБЛАСТИ ЗНАНИЙ НА ОСНОВЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ ТЕОРИИ АДАПТИВНОГО РЕЗОНАНСА	507
О.Е. Елисева, Е.А. Ерченко КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ИНТЕРАКТИВНЫХ МЕТОДОВ ОБУЧЕНИЯ ИНОСТРАННОМУ ЯЗЫКУ: ПОСТАНОВКА ЗАДАЧИ	513
С.А. Самодумкин ИНТЕЛЛЕКТУАЛЬНЫЕ ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ	521
А.В. Паркалов ПРИМЕНЕНИЕ НЕЙРОННЫХ И СЕМАНТИЧЕСКИХ СЕТЕЙ ДЛЯ СЕГМЕНТАЦИИ РАСТРОВЫХ ИЗОБРАЖЕНИЙ ЗЕМНОЙ ПОВЕРХНОСТИ	527
А.П. Кузнецов, В.Н. Никонов, А.С. Шмарловский, М.В. Силивонец. АЛГОРИТМЫ УПРАВЛЕНИЯ ЛИФТАМИ С ИСПОЛЬЗОВАНИЕМ СЕМАНТИЧЕСКОЙ ИНФОРМАЦИИ	531
Д.Л. Шилин АНАЛИЗ ФАЗОВОГО ДЕТЕКТОРА ТИПА «ВЫБОРКА-ЗАПОМИНАНИЕ»	537
АВТОРСКИЙ УКАЗАТЕЛЬ	541

TABLE OF CONTENTS

FOREWORD	20
SECTION 1. BASIC UNIVERSAL SEMANTIC MODELS OF KNOWLEDGE REPRESENTATION AND KNOWLEDGE PROCESSING BOTH THEIR PROGRAM AND HARDWARE IMPLEMENTATION	21
V. V. Golenkov, N. A. Guliakina GRAPHODYNAMICAL MODELS OF PARALLEL KNOWLEDGE PROCESSING	23
B.A. Kobrinskiy IMAGE REPRESENTATIONS OF THE EXPERT AND PROBLEM THEIR REFLECTIONS IN INTELLECTUAL SYSTEMS	53
V.A. Timchenko THE MODEL OF SEMANTIC NETWORK CLASSES AND THEIR TRANSFORMATIONS	63
L. Yu. Zhilyakova DISCRETE MODELS OF DIFFUSION ON GRAPHS	71
M.G. Malkovsky, S.Y. Soloviev TERMINOLOGICAL NETWORKS	77
V.I. Bodyakin AUTOMATIC CONSTRUCTION OF SEMANTIC MODELS OF ARBITRARY SUBJECT DOMAINS ON BASE OF NEURONSEMANTIC APPROACH	83
A.E. Baranovich, N.O. Nikitin ABOUT SOME APPLICATION AREAS OF THE ALGEBRAIC MODEL FORK–HYPERSPACE OF SH-HYPERTOPOGRAPHS	91
L.S. Globa, M.Y. Ternovoy, O.S. Shtogrina APPROACH FOR STORING FUZZY KNOWLEDGE BASES	99
Y.N. Molchanov, L. S. Globa, N. A. Alexeyev CHANGE DETECTION OF INTERNET DOCUMENTS	103
D.G. Kolb THE WEB-BASED IMPLEMENTATION SEMANTIC MODELS OF INTELLIGENT SYSTEMS	111
A.A. Ivaniuk CURRENT STATE AND DEVELOPMENT TRENDS OF TECHNOLOGIES OF RECONFIGURABLE DIGITAL DEVICES	123
N. V. Abramov, A.A. Ivaniuk RECONFIGURABLE COMPUTING AS MEANS OF SOLUTIONS' OPTIMIZATION OF TASKS OF COMMON FIELD	129
S. Bairak, D. Adzinets, M. Tatur, P. Philipoff, M. Mucoz PARALLEL PROCESSORS FOR INTELLIGENT SYSTEMS DEVELOPMENT	135

SECTION 2. SEMANTIC TECHNOLOGIES OF DESIGNING OF KNOWLEDGE BASES, PROGRAMS AND THE SOFTWARE PACKAGES ORIENTED ON KNOWLEDGE PROCESSING	141
V.F. Khoroshevsky SEMANTIC TECHNOLOGIES: EXPECTATIONS AND TRENDS	143
O. P. Kuznetsov THE POSSIBILITY OF KNOWLEDGE STRUCTURING BASED ON COGNITIVE SEMANTICS	159
G.S. Plesniewicz FORMAL ONTOLOGIES	163
A.P. Ereemeev, A.A. Ereemeev INTEGRATION OF THE DATA WAREHOUSES TOOLS AND THE TEMPORAL MODELS	169
A.A. Gorkina INTEGRATION OF THE LOGIC OF MOTION AND BINARY DATA AND KNOWLEDGE MODEL	175
Yu.A. Zagorulko, O.I. Borovikova, G.B. Zagorulko DEVELOPMENT OF MULTILINGUAL THESAURUS BY MEANS OF SEMANTIC TECHNOLOGY	181
N. A. Morosanova RULES UNCERTAINTY IN RULE-BASED EXPERT SYSTEMS	189
V.P. Ivashenko SEMANTIC MODELS AND TOOLS FOR KNOWLEDGE BASE INTEGRATION AND DEBUGGING	193
V.V. Dikovitsky EXTRACTION OF USER KNOWLEDGE AND VERIFICATION OF KNOWLEDGE SELF-ORGANIZING INFORMATION SYSTEMS WITH FEEDBACK	205
V.A. Kataev EXPERT SYSTEMS DEVELOPMENT IN THE MULTI STUDIO ENVIRONMENT	207
V. Gribova, A. Kleshev ONTOLOGICAL PROGRAMMING PARADIGM	213
N.A. Guliakina, O.V. Pivovarchyk, D.A. Lazurkin INTELLECTUAL SYSTEM FOR INFORMATION SERVICE OF SCP-PROGRAMMES' DEVELOPERS	221
I.S. Volegov, E.B. Zamyatina THE INTEGRATION OF SIMULATION MODEL COMPONENTS INTO TRIAD.NET USING ONTOLOGICAL APPROACH	229

SECTION 3. SEMANTIC MODELS OF INFORMATION SEARCH AND PROBLEM SOLVING. SEMANTIC TECHNOLOGIES OF DESIGN OF INTELLIGENT INFORMATION RETRIEVAL ENGINES AND INTELLIGENT PROBLEM SOLVERS	237
<hr/>	
J.V. Rogushina SEMANTIC SEARCH AS A COMPONENT OF KNOWLEDGE MANAGEMENT IN SEMANTIC WEB	239
<hr/>	
E.V. Birialtsev, M.R. Galimov, N.G. Zhiltsov, O.A. Nevzorova A NOVEL APPROACH TO SEMANTIC SEARCH OF MATHEMATICAL EXPRESSIONS IN THE SCIENTIFIC DOCUMENTS	245
<hr/>	
V.S. Dyundyukov, V.B. Tarassov GENERATION OF MULTI-AGENT SYSTEMS BY USING RESOURCE-GOAL NETWORKS	257
<hr/>	
V.B. Tarassov, A.P. Kalutskaya, M.N. Svyatkina GRANULAR, FUZZY AND LINGUISTIC ONTOLOGIES TO ENABLE MUTUAL UNDERSTANDING BETWEEN COGNITIVE AGENTS	267
<hr/>	
O.O. Varlamov THE MIVAR TECHNOLOGIES OF LOGICAL ARTIFICIAL INTELLIGENCE CREATING: LOGICAL INFERENCE WITH LINEAR COMPUTATIONAL COMPLEXITY WITH MORE THAN 3 MILLIONS PRODUCTION RULES AND POSSIBILITY TO UNDERSTAND A MEANING THROUGH MODELING OF ABIG CONTEXT	279
<hr/>	
Yu.M. Smetanin PROBABLISTIC LOGIC AND ORTOGONAL BASIS OF SYLLOGISTICS	289
<hr/>	
D.V. Shunkevich, S.S. Zalivako SEMANTIC TECHNOLOGY OF COMPONENTIAL DESIGNING OF INTELLECTUAL PROBLEM SOLVERS	297
<hr/>	
L.G. Komartsova, Ju.N. Lavrenkov, O.V. Antipova, D.S. Kadnikov DYNAMIC MODELS OF INTELLIGENT SYSTEMS ON THE BASE OF ADAPTIVE RESONANCE THEORY NEURAL NETWORK	315
<hr/>	
A.I. Zhukevich, E.V. Olizarovich, V.G. Rodchenko USE OF ONTOLOGY FOR THE CONSTRUCTION OF PATTERN RECOGNITION SYSTEMS	321

SECTION 4. SEMANTIC TECHNOLOGIES OF DESIGNING MULTIMODAL AND NATURAL LANGUAGE INTERFACES OF INTELLECTUAL SYSTEMS	325
S.S. Curbatov, A.P. Lobzin, K.A. Naidenova, G.K. Khakhalin THE HYBRID CIRCUIT OF THE IMAGE ANALYSIS	327
N.M. Borgest, R.V. Chernov, D.V. Shustova INTERFACE OF INTELLECTUAL ASSISTANT OF DESIGNER	335
D. N. Koronchik SEMANTIC MODELS OF MULTIMODAL USER INTERFACES AND SEMANTIC TECHNOLOGY FOR THEIR DESIGN	339
A.V. Zaboлева-Zotova, Y.A. Orlova, V.L. Rozaliev, A.S. Bobkov PPROACH TO THEPROBLEMOFCREATING AN AUTOMATED EMOTION-RECOGNITION SYSTEM	347
A.V. Zaboлева-Zotova, Y.A. Orlova, V.L. Rozaliev, A.S. Bobkov SOLVING THE PROBLEM OF FORMALIZING HUMAN MOVEMENT AS A FUZZY TEMPORAL EXPRESSIONS INORDER TO DETERMINE THE EMOTIONAL REACTIONS OF THE PERSON	351
V.V. Kyselov, A.G. Davydov, A.V. Tkachenja SYSTEM OF DETERMINATION OF THE EMOTIONAL STATE OF THE SPEAKER OF A SOFTWARE TO THE VOICE	355
A.G. Kyselova, G.D. Kyselov CONTEXT DATA PROCESSING FOR INTELLIGENT SYSTEM	359
E.B. Kozerenko COGNITIVE LINGUISTIC PRESENTATIONS OF LANGUAGE STRUCTURES IN ARTIFICIAL INTELLIGENCE AND MACHINE TRANSLATION SYSTEMS	365
I.P. Kuznetsov, M.M. Charnine, E.B. Kozerenko, A.G. Matskevich, V.G. Nikolayev, N.V. Somin INTELLIGENT SYSTEMS FOR ENTITIES EXTRACTION BASED ON EXTENDED SEMANTIC NETWORKS	373
I.V. Efimenko ONTOLOGY OF AUTHORITY: APPEALING TO EXTENAL OPINIONS IN TEXTS OF DIFFERENT GENERES	381

A.A. Kharlamov, T.V. Yermolenko AUTOMATICAL MAKING OF HETEROGENEOUS SEMANTIC NETWORKON THE BASE OF EXTRACTION OF KEY PREDICATE STRUCTURES OF TEXT SENTENCES	385
D.Sh. Suleymanov, A.R. Gatiatullin SEMANTIC UNIVERSALS IN THE SYSTEM OF DESCRIPTION OF MORPHEME MEANINGS	391
A.A. Sharipbaev, G.T. Bekmanova, B.J. Ergesh, A.K. Buribaeva, M.H. Karabalaeva THE INTELLECTUAL MORPHOLOGICAL ANALYZER BASED ON SEMANTIC NETWORKS	397
Y.S. Hetsevich, S.A. Hetsevich, O.E.Yeliseyeva, V.A. Zhitko, A.A. Kuzmin SEMANTIC TECHNOLOGY DESIGNNLINTERFACES FOR QUESTION ANSWERING SYSTEMS	401
Y.A. Kim, O. E. Yeliseyeva FORMALIZATION OF THE SEMANTICS OF PRAGMATIC PHRASEMES OF THE BELARUSIAN LANGUAGE	413
A.I. Golovnja SOME ASPECTS OF SEMANTIC CODING: METHODS OF EXPRESSION OF CAUSALITY IN LANGUAGE-SYSTEM	419
SECTION 5. METHODOLOGIES AND MANAGEMENT OF DEVELOPMENT OF INTELLECTUAL SYSTEMS ON THE BASIS OF SEMANTIC TECHNOLOGIES WITHIN THE LIMITS OF OPEN SOURCE PROJECTS	425
V.A. Maklaev, P.I. Sosnin PRECEDENT-FOCUSED BASE OF EXPERIENCE OF THE DESIGN ORGANIZATION	427
A.S. Kleshev, E.A. Shalfeeva PROFESSIONAL ACTIVITY ONTOLOGY	437
T.A. Gavrilova, N.E. Melnikov ABOUT PECULIARITIES OF KNOWLEDGE MANAGEMENT IN RUSSIAN SOFTWARE COMPANIES	443
L.S. Globa, R.L. Novogradskaja MODELS AND METHODS OF INFORMATIONAL AND COMPUTATIONAL RESOURCES INTEGRATION	447
V.V. Bakhtizin, S.N. Niaborski INTELLIGENT SYSTEMS QUALITY EVALUATION	453
I.T. Davydenko INTEGRATED METHODOLOGY OF DESIGN OF THE INTELLIGENT REFERENCE SYSTEM SEMANTIC MODEL	457

SECTION 6. THE APPLICATION-ORIENTED INTELLECTUAL SYSTEMS BASED ON SEMANTIC NETWORKS	467
Y.I. Maiboroda TECHNIQUE OF THE FREQUENCY-SPATIAL PROBLEM SOLVING BASED ON USAGE OF PRODUCTIONAL SYSTEMS AND HEURISTIC ALGORITHMS	469
A. Kleschev, M. Chernyahovskaya, E. Shalfeeva THE ANALYSIS OF MEDICAL INTELLECTUAL PROFESSIONAL ACTIVITY FROM THE VIEWPOINT OF AUTOMATION	473
Yu.M. Smetanin, M. Yu. Smetanin MEDICAL DIAGNOSTICS AND ORTHOGONAL BASIS OF SYLLOGISTIC	479
A.I. Vovk, Y.Y. Ruban, D.A. Girnyk DEVELOPMENT OF TEST SYSTEMS USING WEB-ORIENTED NOTATION, MATHEMATICAL TEXTS, WHICH PRESERVES THE SEMANTICS	489
A. Gladun, J. Rogushina COGNITIVE NETWORKS AND THE ONTOLOGICAL ANALYSIS IN ADAPTIBILITY AND SERVICE REFINEMENT OF HETEROGENEOUS WIRELESS ENVIRONMENT	493
L.S. Globa, I.A. Kuzin, K.S. Mochalkina, R.L. Novogradskaya NATIONAL ANTARCTIC DATA CENTRE INTERNET-PORTAL MODEL	501
O.A. Kozlov, J.F. Mikhailov CONSTRUCTION OF INTELLIGENT INFORMATION SYSTEMS THE FORMATION OF AN INDIVIDUAL TRAJECTORY EXAMINE THE STUDENTS SOME DOMAIN KNOWLEDGE ON THE BASIS OF ARTIFICIAL NEURAL NETWORK THEORY OF ADAPTIVE RESONANCE	507
O.E. Eliseeva, E. A. Erchenko COMPUTER MODELLING OF INTERACTIVE METHODS OF THE FOREIGN LANGUAGE STUDYING: PROBLEM STATEMENT	513
S. Samodumkin INTELLECTUAL GEOINFORMATION SYSTEMS	521
A.V. Parkalov APPLICATION OF NEURAL AND SEMANTIC NETWORKS IN THE SEGMENTATION OF THE EARTH'S SURFACE BITMAPS	527
A.P. Kuznetsov, V.N. Nikonov, A.S. Shmarlouski, M.V. Silivonets ELEVATOR CONTROL ALGORITHMS USING SEMANTIC INFORMATION	531
D.L. Shilin ANALYSIS OF SAMPLE-AND-HOLD PHASE DETECTOR	537
AUTHOR INDEX	543

ПРЕДИСЛОВИЕ

В настоящее время развитие искусственного интеллекта определяет глубокая интеграция трех составляющих:

- научных исследований, направленных на создание **общей теории** интеллектуальных систем и разработку доступной **комплексной технологии** компонентного проектирования семантически совместимых интеллектуальных систем самого различного назначения;
- практической коммерческой деятельности, направленной на массовую разработку и внедрение прикладных интеллектуальных систем в самых различных областях;
- учебной деятельности, направленной на активное вовлечение студентов, магистрантов и аспирантов в указанную выше научную и практические работы через систему согласованных Open Source проектов.

Особую актуальность всё это приобретает для постсоветского пространства. Основным препятствием на этом пути является отнюдь не отсутствие финансирования, а человеческий фактор и отсутствие опыта реального сотрудничества различных научных школ и коммерческих организаций, направленного на получение общих практически полезных результатов. Если мы хотим прийти к созданию конкурентоспособных продуктов, мы обречены на плодотворное активное сотрудничество.

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 комплексной массовой технологии компонентного проектирования (модульного, сборочного проектирования) интеллектуальных систем, в состав которой входит:

- **библиотека** типовых, многократно используемых и **семантически совместимых компонентов** интеллектуальных систем (компонентов баз знаний, интеллектуальных решателей задач, интеллектуальных пользовательских интерфейсов);
- семейство совместимых языков **семантического представления знаний** различного вида, обеспечивающих семантическую совместимость не только многократно используемых компонентов интеллектуальных систем, но и целых интеллектуальных систем;
- семейство совместимых **семантических моделей решения задач**.

Фундаментальными проблемами, решение которых лежит в основе указанной технологии, являются:

- проблема обеспечения **семантической совместимости** интеллектуальных систем и их компонентов;
- проблема обеспечения **независимости абстрактных семантических моделей интеллектуальных систем** и технологии проектирования таких моделей от различных вариантов их технической реализации.

Председатель Программного комитета конференции OSTIS-2012.

Председатель Совета Российской ассоциации искусственного интеллекта

Кузнецов О. П.

FOREWORD

Nowadays a deep integration of three components determines evolution of artificial intelligence. These components are:

- scientific researches, that are focused on the creation of common theory of intelligent systems and development of component-based technology for semantically compatible intelligent systems design;
- commercial activity, that focused on mass develop and implementation of applied intelligent systems in various areas of human activity;
- education activity, that focused on involving of students into scientific and practical activities by using a system of coordinated open source projects.

It is more actual for the post-Soviet territory especially. There are two problems on this way. The first problem is a human factor. Also there are no experience of cooperation between various scientific and commercial organizations, that focused on making practically useful results. So, if we want to create competitive products, then we are doomed to active cooperation.

Creation of conditions for expansion of cooperation between various scientific, education and commercial organizations, that focused on development of component-based technology for intelligent systems design, is the main aim of annual conference OSTIS (Open Semantic Technology for Intelligent Systems). This technology consists of:

- library of components. It contains components, which are semantically compatible and can be used for many times (components for knowledge bases, components for intelligent solver of problems, components for intelligent user interfaces);
- family of compatible semantic languages, that focused on semantic representation of knowledge's. These languages provide semantic compatibility of components and intelligent systems;
- family of compatible semantic models of problem solving.

Solution of two fundamental problems is the basis of technology. These problems are:

- ensuring of semantically compatibility of intelligent systems and their components;
- ensuring independence of abstract semantic models of intelligent systems and technology that allows to design these models from a variety of options for their technical implementation.

Programme Committee Chair

Chairman of the Council of Russian Association for Artificial Intelligence

Kuznetsov O. P.

СЕКЦИЯ № 1

БАЗОВЫЕ УНИВЕРСАЛЬНЫЕ СЕМАНТИЧЕСКИЕ МОДЕЛИ ПРЕДСТАВЛЕНИЯ И ОБРАБОТКИ ЗНАНИЙ И ИХ ПРОГРАММНАЯ И АППАРАТНАЯ РЕАЛИЗАЦИЯ

SECTION № 1

BASIC UNIVERSAL SEMANTIC MODELS OF KNOWLEDGE REPRESENTATION AND KNOWLEDGE PROCESSING BOTH THEIR PROGRAM AND HARDWARE IMPLEMENTATION

Секция 1 объединила работы, направленные на создание базовой универсальной семантической модели представления и обработки знаний, а также работы, направленные на создание различных вариантов технической реализации таких семантических моделей. Это не случайно. Такое объединение подчеркивает непосредственную связь между этими направлениями. С одной стороны, разработка базовых абстрактных семантических моделей должна осуществляться с учетом последующей их технической реализации. С другой стороны, для разработки различных вариантов такой технической реализации указанные абстрактные семантические модели являются способом строгой формальной постановки задачи, способом формального описания различных технических решений на верхнем абстрактном уровне.

Подчеркнем, что базовая универсальная семантическая модель представления и обработки знаний может иметь большое количество вариантов её технической реализации, которые можно разбить на следующие группы:

- Internet-ориентированные варианты программной реализации семантических моделей представления и обработки знаний, которые в настоящее время интенсивно разрабатываются в рамках направления Semantic Web;
- Локальные варианты программной реализации семантических моделей представления и обработки знаний на доступных компьютерах традиционного вида или на суперкомпьютерах;
- Варианты аппаратной реализации семантических моделей представления и обработки знаний с помощью реконфигурируемых вычислительных устройств, на основе которых можно рассматривать различные аппаратные решения, направленные на создание реконфигурируемой ассоциативной процессоро-памяти.

Значение базовых универсальных семантических моделей представления и обработки знаний заключается не только в том, что они являются абстрактным описанием различных вариантов их технической реализации, но и в том, что они являются фундаментом (носителем) для разработки семантических моделей более высокого уровня:

- семантических моделей представления знаний различного вида (в том числе и программ);
- различных семантических моделей решения задач;
- семантических моделей пользовательских интерфейсов.

Указанные модели рассматриваются в следующих секциях.

ВОПРОСЫ, ВЫНОСИМЫЕ НА ОБСУЖДЕНИЕ КРУГЛОГО СТОЛА СЕКЦИИ № 1:

- *Каким требованиям должна удовлетворять базовая семантическая модель представления и обработки знаний.*
- *Какие задачи необходимо решить для поддержки параллельного асинхронного взаимодействия агентов, работающих над общей семантической памятью.*
- *Базовые унифицированные логико-семантические модели проектируемых интеллектуальных систем как основа обеспечения независимости технологии проектирования интеллектуальных систем от различных платформ и вариантов их реализации.*
- *Базовые унифицированные логико-семантические модели проектируемых интеллектуальных систем как основа обеспечения семантической совместимости (интегрируемости) интеллектуальных систем и их компонентов*
- *Можно ли и нужно ли интерпретировать нейросетевые модели, генетические алгоритмы и другие модели решения задач в графодинамических моделях обработки информации.*
- *Почему разработка компьютеров пятого поколения не привела к ожидаемым результатам.*
- *Нужна ли аппаратная реализация семантических моделей интеллектуальных систем и возможно ли это в принципе.*
- *Какие подходы возможны для аппаратной реализации агентов, работающих над общей графодинамической памятью.*
- *Что такое реконфигурируемая процессоро-память.*



УДК 004.822:514

ГРАФОДИНАМИЧЕСКИЕ МОДЕЛИ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ЗНАНИЙ: ПРИНЦИПЫ ПОСТРОЕНИЯ, РЕАЛИЗАЦИИ И ПРОЕКТИРОВАНИЯ

Голенков В. В., Гулякина Н.А.

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

golen@bsuir.by

guliakina@bsuir.by

Рассматриваются принципы построения технологии проектирования интеллектуальных систем, ориентированной на семантическое представление знаний, расширение контингента разработчиков и сокращение сроков проектирования.

Ключевые слова: интеллектуальная система, технология проектирования интеллектуальных систем, семантическая сеть, язык семантических сетей, алфавит семантических сетей, ключевые узлы семантических сетей, бинарная семантическая сеть, предметная область.

ВВЕДЕНИЕ

Мы исходим из того, что основным результатом исследований в области искусственного интеллекта являются не столько разработка конкретных интеллектуальных систем, обладающих высокими интеллектуальными возможностями, сколько **разработка технологий** позволяющих быстро и в большом количестве порождать самые разнообразные интеллектуальные системы, имеющие большую практическую ценность. Очевидно, что составляющими таких технологий являются:

- формальная теория интеллектуальных систем (формальное описание того, как они устроены);
- методы проектирования интеллектуальных систем;
- инструментальные средства (средства автоматизации проектирования интеллектуальных систем);
- средства информационной поддержки (информационного обслуживания) разработчиков интеллектуальных систем;
- средства компьютерной поддержки управления коллективной разработкой интеллектуальных систем.

Современные технологии проектирования интеллектуальных систем имеют ряд недостатков:

- технологии искусственного интеллекта не ориентированы на широкий круг разработчиков интеллектуальных систем и, следовательно, не получили массового распространения;
- велики сроки разработки интеллектуальных систем и велика трудоемкость их сопровождения;

- высока степень зависимости технологий искусственного интеллекта от платформ, на которых они реализованы, что приводит к существенным изменениям технологий при переходе на новые платформы;
- для эффективной реализации даже существующих моделей представления знаний и моделей решения трудно формализуемых задач современные компьютеры оказываются плохо приспособленными, что требует разработки принципиально новых компьютеров;
- современно состояние в области проектирования интеллектуальных компьютерных систем представляет собой "вавилонское столпотворение" самых различных моделей, методов, средств, платформ;
- отсутствуют подходы, позволяющие на некоторой универсальной основе интегрировать научные и практические результаты в области искусственного интеллекта, что порождает высокую степень дублирования результатов. В частности, высока трудоемкость интеграции различных моделей представления информации, моделей обработки информации, моделей решения задач и, следовательно, различных интеллектуальных компьютерных систем.

Искусственный интеллект является междисциплинарной научной дисциплиной. Этим обусловлен большой её потенциал, так как на стыках научных направлений рождаются сильные результаты. Но этим же обусловлены и большие трудности, так как развитие искусственного интеллекта требует глубокого взаимопонимания и

сотрудничества исследователей, имеющих разный стиль мышления, разный подход к объекту и предмету исследования, разный менталитет, разные целевые установки и традиции. Современный этап развития искусственного интеллекта остро нуждается в преодолении указанных трудностей.

Важнейшей задачей искусственного интеллекта в настоящее время является построение общей комплексной теории интеллектуальных систем, в рамках которой бы сочетались самые разные направления искусственного интеллекта – и теория представления знаний, и теория решения задач (в том числе различные исчисления, эвристики, стратегии), и теория программ (процедурных, декларативных, параллельных, последовательных), и архитектуры интеллектуальных систем, в том числе детализированные до уровня аппаратной поддержки, и теория интеллектуальных пользовательских интерфейсов, и компьютерная лингвистика.

Сейчас эпицентром развития искусственного интеллекта является не столько разработка отдельных его направлений, сколько их глубокая **семантическая интеграция**, целью которой должна быть не только общая теория интеллектуальных систем, но и общая, доступная технология их комплексного проектирования.

В основе предлагаемого нами подхода к созданию технологии проектирования интеллектуальных систем, направленного на устранение выше указанных недостатков, лежат следующие принципы.

Принцип 1. Использовать опыт наиболее продвинутых технологий

В первую очередь, имеется в виду технология проектирования микросхем, которая за последнее время обеспечила существенное сокращение времени и повышение качества разработок,

- (1) благодаря созданию языковых средств формального описания проектируемых микросхем на разных уровнях детализации,
- (2) благодаря четкому разделению процесса разработки формальных описаний микросхем и процесса их реализации по заданным формальным описаниям,
- (3) благодаря созданию мощных и доступных библиотек формальных описаний типовых (многократно используемых) компонентов микросхем.

Для того, чтобы аналогичным образом построить технологию проектирования интеллектуальных систем, необходимо

- (1) создать языковые средства полного унифицированного формального описания интеллектуальных систем,
- (2) четко отделить разработку полного унифицированного формального описания

проектируемой интеллектуальной системы от разработки различных вариантов интерпретации таких формальных описаний интеллектуальных систем,

- (3) создать библиотеку формальных описаний типовых (многократно используемых) компонентов интеллектуальных систем. Но для того, чтобы такая библиотека была создана, необходимо обеспечить **интегрируемость** (семантическую совместимость) указанных компонентов интеллектуальных систем.

Принцип 2. Графодинамические модели:

В качестве формальной основы проектируемых интеллектуальных систем использовать **графодинамические модели обработки информации**.

Графодинамическая модель обработки информации трактует процесс обработки информации как процесс преобразования графовой структуры, в ходе которого меняется не только состояние элементов этой графовой структуры, но и конфигурация этой структуры (появляются или удаляются её вершины, а также связи между ними). Заметим, что для создания графодинамических моделей обработки информации недостаточно тех видов графовых структур, которые в настоящее время исследуются в теории графов. Нам потребуется не только увеличение числа компонентов инцидентных ребру (т. е. переход от ребра к **гиперребру**), но и увеличение числа компонентов инцидентных дуге (т. е. переход от дуги к **гипердуге**, которая по сути является графовой трактовкой многоместного кортежа). Нам потребуются не только ребра, гиперребра, дуги, гипердуги, компонентами которых являются вершины графовой структуры, но и ребра, гиперребра, дуги, гипердуги, компонентами которых являются другие ребра, гиперребра, дуги и гипердуги. Нам потребуются такие связующие элементы графовых структур, которые задают целые фрагменты (подструктуры) заданной графовой структуры, в состав которых входят соответствующие вершины, ребра, гиперребра, дуги, гипердуги [Попков, 1986].

Приведем общее определение **графовой структуры**, на основе которого можно строить практически полезные графодинамические модели обработки информации. Графовая структура G задается пятеркой $\langle V, C, K, M, I \rangle$, где

V – множество **вершин** (первичных элементов, терминальных элементов);

C – множество **связующих элементов** графовой структуры, каждый из которых задает некоторый фрагмент графовой структуры;

K – множество **ключевых вершин** графовой структуры, каждая из которых задает некоторый класс эквивалентных (однотипных) в определенном смысле элементов графовой структуры ($K \subset V$);

M – множество **меток** элементов (алфавит элементов) графовой структуры, каждая из которых задает некоторый базовый класс эквивалентных в определенном смысле элементов графовой структуры. К таким классам элементов, в частности, относятся:

- класс всех вершин графовой структуры;
- класс всех связующих элементов графовой структуры;
- класс всех ключевых вершин графовой структуры;
- класс всех меток графовой структуры;
- класс всех используемых в графовой структуре отношений инцидентности элементов графовой структуры.

I – множество используемых в графовой структуре **отношений инцидентности** элементов. Все эти отношения инцидентности являются бинарными ориентированными отношениями. Среди этих отношений выделим:

- отношения инцидентности вершин. Примером такого отношения является последовательность символов в строке символов;
- отношения инцидентности, каждая пара которых связывает связующий элемент графовой структуры с элементом (компонентом) того фрагмента, который задается этим связующим элементом. Подчеркнем, что компонентами связующего элемента могут быть элементы графовой структуры любого вида (вершины, связующие элементы, метки, отношения инцидентности). Подчеркнем также, что компоненты одного и того же связующего элемента могут выполнять разные роли в рамках соответствующего фрагмента графовой структуры, который задается связующим элементом. Указанные роли соответствуют разным отношениям инцидентности, входящим во множество I ;
- отношения инцидентности, каждая пара которых связывает ключевую вершину графовой структуры с тем элементом графовой структуры, который входит в класс элементов, задаваемый этой ключевой вершиной. Подчеркнем, что элементами, инцидентными ключевой вершине, могут быть элементы графовой структуры любого вида;
- отношения инцидентности, каждая пара которых связывает метку графовой структуры с тем элементом графовой структуры, который имеет указанную метку. Подчеркнем, что элементами, инцидентными метке, могут быть элементы графовой структуры любого вида.

Заметим, что связь каждого отношения инцидентности (каждого элемента множества I) с соответствующими парами инцидентности и связь

каждой пары инцидентности с элементами графовой структуры, соединяемыми этой парой инцидентности, можно условно считать неявно задаваемыми связями инцидентности более низкого уровня.

Заметим также, что каждую графовую структуру G мы будем трактовать как множество всех элементов, входящих в её состав:

$G = (V \cup C \cup K \cup M \cup I)$. Таким образом, в число элементов графовой структуры входят все её вершины (в том числе ключевые), связующие элементы, метки и отношения инцидентности.

Заметим также, что множество связующих элементов графовой структуры можно разбить на:

- множество **связок** (простых связующих элементов);
- множество **подструктур**, каждая из которых задает фрагмент графовой структуры, в состав которого входят такие связующие элементы, которые связывают элементы графовой структуры, входящие в указанный фрагмент.

В свою очередь, по признаку ориентированности множество связок можно разбить на:

- множество **ориентированных связок**, некоторые компоненты которых выполняют в рамках этих связок разные роли;
- множество **неориентированных связок**, все компоненты которых выполняют в рамках этих отношений связок одинаковые роли.

Важным частным случаем ориентированной связки является **кортеж**. Кортеж задает такое подмножество элементов графовой структуры, в котором роли всех элементов пронумерованы. То есть в рамках указанного подмножества имеется элемент, который в этом подмножестве выполняет роль первого элемента (первого компонента), имеется элемент, который в этом подмножестве выполняет роль второго элемента (второго компонента) и т. д.

Если в графовой структуре имеются кортежи, то в число её отношений инцидентности должны входить следующие отношения:

- быть первым компонентом;
- быть вторым компонентом;
- быть третьим компонентом;
- и т. д.

По количеству компонентов множество связок можно разбить на:

- множество **унарных связок** (одноместных, однокомпонентных);
- множество **бинарных связок** (двухместных, двухкомпонентных);
- множество **многокомпонентных связок** (многоместных), имеющих более двух компонентов.

Неориентированные бинарные связки будем называть **ребрами**, ориентированные – **дугами**. Неориентированные многокомпонентные связки

будем называть **гиперребрами**, а ориентированные – **гипердугами**.

Множество связующих элементов графовой структуры можно трактовать как подмножество **шкалы множеств**, заданной над множеством $(V \cup M \cup I)$.

Шкала множеств (H) над указанным множеством определяется рекурсивно:

(1) $H \supset (V \cup M \cup I)$

(2) если $hj1, hj2, \dots, hjn \in H$

то $\{hj1, hj2, \dots, hjn\} \in H$

/* т. е. любое множество, состоящее из любых элементов шкалы множеств, само также является одним из элементов шкалы множеств */

(3) если $hj1, hj2, \dots, hjn \in H; ij1, ij2, \dots, ije \in I$

то множество $hj = \{hj1, hj2, \dots, hjn\}$, между которым и множеством $\{ij1, ij2, \dots, ije\}$ задано произвольное соответствие, определяющее то или иное распределение ролей между элементами множества hj , также является элементом шкалы (т. е. $hj \in H$)

Какие вопросы ассоциируются с рассмотрением графодинамической модели (графодинамической парадигмы) обработки информации:

- Можно ли на основе теории графов построить **универсальную абстрактную модель обработки информации**, которая могла бы конкурировать с абстрактной машиной фон Неймана, лежащей в основе традиционных компьютерных систем;
- Нужно ли это делать, какими преимуществами эта модель обладает по сравнению с абстрактной машиной фон Неймана. Какими преимуществами обладают системы, создаваемые на основе этой модели. Что принципиально нового дает графодинамическая парадигма обработки информации;
- Есть ли к этому предпосылки;
- Можно ли различные модели решения задач (в том числе различные логические исчисления) формально описать в виде графодинамических моделей обработки информации и можно ли обеспечить совместимость (интегрируемость) таких графодинамических моделей.

Интерес к графодинамическим моделям обработки информации имеет достаточно длительную историю. Для подтверждения этого достаточно отметить:

- Предложенное А.Н. Колмогоровым уточнение понятия алгоритм [Колмогоров, 1958];
- Работы школы М.А. Айзермана по графодинамике [Айзерман, 1988];
- Исследования по графовым грамматикам [Петров, 1987];
- Исследования по теории программирования и

CASE-технологиям [Касьянов, 2003];

- Разработка параллельных моделей обработки информации [Котов, 1966];
- Предложенные В.Б. Борщевым и М.В. Хомяковым клубные системы и вегетативная машина [Борщев, 1983].

Для разработки графодинамических моделей обработки информации необходимо рассматривать графовую структуру с позиций семиотики и трактовать её как **знаковую структуру** (текст), представляющую собой систему взаимосвязанных знаков. Такая трактовка графовых структур позволяет "вдохнуть" семантику в теорию графов.

Действительно, почему тексты обязательно должны быть линейными (т. е. цепочками символов). Но, как только мы введем понятие **графового языка** (т. е. языка, текстами которого являются в общем случае графовые структуры различной конфигурации), возникают следующие вопросы:

- В чем преимущество графовых языков по сравнению с традиционными линейными языками, текстами которых являются цепочки (строки) символов;
- Можно ли построить **универсальный графовый язык**, обеспечивающий представление информации (знаний) любого семантического вида;
- Можно ли в универсальном графовом языке сделать так, чтобы множество всех меток, используемых во всех графовых структурах, являющихся текстами универсального графового языка, было конечным.

Говоря о графовых языках, следует подчеркнуть то, что графовые структуры, являющиеся текстами таких языков, представляют собой **абстрактные** математические структуры, не уточняющие (не детализирующие) способ их материального представления (например, способ кодирования в компьютерной памяти, способ графического изображения, ориентированного на человеческое восприятие). То есть графовая структура как абстрактный математический объект и её, например, графическое представление – это принципиально разные вещи. Из этого, в частности, следует, что каждому графовому языку может соответствовать несколько языков, использующих разные способы представления (кодирования, изображения) текстов этого графового языка.

Накопленный опыт развития и применения теории графов и все полученные в ней результаты становятся хорошим математическим фундаментом для разработки различных графовых языков и различных графодинамических моделей обработки информации, а также для создания теории таких языков и моделей. На стыке теории графов и семиотики может появиться очень интересный раздел семиотики – **графовая семиотика**.

Принцип 3. Параллельные асинхронные графодинамические модели:

В качестве формальной основы проектируемых интеллектуальных систем использовать графодинамические модели специального вида, ориентированные на **параллельную и асинхронную** обработку информации.

Почему акцентируется внимание на **параллельных** графодинамических моделях. Во-первых, потому, что без организации параллельной обработки информации невозможно рассчитывать на необходимую производительность подавляющего числа практически полезных интеллектуальных систем. Во-вторых, потому, что целый ряд исследований [Котов, 1966] показал перспективность создания параллельных моделей обработки информации именно на основе графодинамического подхода.

Почему отдается предпочтение **асинхронному** варианту управления обработкой информации. Потому, что асинхронные модели обработки информации являются более гибкими, их легче интегрировать и наращивать новыми функциональными возможностями.

Графодинамическая модель параллельной асинхронной обработки информации, которую будем также называть **графодинамической параллельной асинхронной машиной**, трактуется нами как абстрактная **многоагентная система**, состоящая из:

- абстрактной **графодинамической памяти**, в которой хранятся обрабатываемые графовые структуры;
- **коллектива агентов**, работающих над общей для них графодинамической памятью и обменивающихся информацией только через эту память (в т. ч. и для координации своих действий).

Графодинамическая память носит реконфигурируемый, структурно перестраиваемый характер, поскольку процесс обработки графовых структур в конечном счете сводится к генерации и удалению различных элементов графовых структур, а также к генерации и удалению пар инцидентности между этими элементами. Другими словами, процесс обработки информации в графодинамической памяти сводится не только к изменению состояния элементов памяти, но и к изменению конфигурации связей между ними.

Агенты, работающие над общей графодинамической памятью, делятся на три вида:

- внутренние агенты, каждый из которых реагирует на определенного вида ситуации или события в графодинамической памяти и осуществляет изменение состояния графодинамической памяти, соответствующее своему функциональному назначению;
- рецепторные агенты, каждый из которых реагирует на определенные события во внешней

среде и осуществляет первичное отражение этих событий в графодинамической памяти;

- эффекторные агенты, каждый из которых реагирует на определенного вида команды, формируемые внутренними агентами в графодинамической памяти, и осуществляет соответствующее изменение материального (физического) состояния интеллектуальной системы, которое определенным образом влияет на изменение её внешней среды.

Агенты могут работать параллельно, если одновременно возникают условия инициирования агентов.

Асинхронность деятельности внутренних агентов заключается в том, что наличие условия инициирования агента ещё не означает начала его работы. То есть время реакции каждого внутреннего агента в известной мере субъективно и достаточно произвольно. В этом смысле указанные агенты обладают:

- свободой выбора момента начала реакции на условие инициирования;
- свободой выбора последовательности обработки условий инициирования, если в текущий момент таких условий возникло несколько.

Для обеспечения эффективного взаимодействия агентов, работающих над общей графодинамической памятью, наряду с представляемой им свободой, необходима разработка таких правил их поведения, которые гарантируют безопасность и производительность каждого из них. В конечном счете, эти правила сводятся к двум положениям:

- позаботиться о своей безопасности, точнее, об обеспечении безопасного выполнения своей задачи;
- не навреди другим агентам (помни о том, что ты не один, не создавай для других "аварийных" ситуаций).

Для обеспечения безопасного выполнения своей задачи агент блокирует некоторые элементы графовой структуры, которая хранится в общей графодинамической памяти. Блокировка – это запрет, установленный заданным агентом и адресованный другим агентам, на выполнение тех или иных действий над заданным элементом хранимой графовой структуры. Таким образом, существует несколько видов таких блокировок. Приведем некоторые из них:

- запрет на удаление заданного элемента графовой структуры;
- запрет на удаление **всех** элементов хранимой графовой структуры, инцидентных заданному (блокируемому) элементу;
- запрет на удаление заданного вида элементов хранимой графовой структуры, которые связаны с блокируемым элементом выходящей из него (или входящей в него) парой инцидентности,

принадлежащей заданному отношению инцидентности;

- запрет на генерацию заданного вида элементов в хранимой графовой структуре, которые связаны с блокируемым элементом выходящей из него (или входящей в него) парой инцидентности, принадлежащей заданному отношению инцидентности.

Приведем некоторые правила поведения агента, работающего в коллективе агентов над общей графодинамической памятью:

- не нарушать блокировочные запреты, сформированные другими агентами;
- самому заблокировать тот фрагмент обрабатываемой графовой структуры, целостность которого необходимо сохранить до завершения своей работы;
- не "жадничать" – не блокировать больше, чем надо;
- снимать свои блокировки как можно быстрее, как только в них отпадает необходимость (т. е. желательно это делать до завершения своей работы);
- удалять сгенерированные для своей работы вспомогательные структуры (информационные "леса") как можно быстрее, как только в них отпадает необходимость (т. е. желательно убирать информационный "мусор" по мере возможности до завершения своей работы);
- поиск фрагментов хранимой графовой структуры, являющихся условиями инициирования агента осуществлять поэтапно, начиная с поиска тех частей этих условий, которые реже появляются в памяти (это необходимо для того, чтобы скорее установить факт отсутствия условий инициирования).

В случае возникновения **конфликтов** между агентами используются внутренние агенты специального вида, реагирующие на возникновение таких конфликтов и обеспечивающие их разрешение (такие агенты будем называть метаагентами-судьями).

Заметим, что для организации своей деятельности над графодинамической памятью каждый агент "опирается" на соответствующее ему семейство постоянно присутствующих в памяти (резидентных) элементов хранимой в памяти графовой структуры. Указанные элементы будем называть ключевыми элементами агентов. Очевидно, что такие элементы соответствуют константам программ, описывающих поведение агентов.

Сложность комплексного перехода на графодинамическую парадигму параллельной асинхронной обработки информации определяется исключительно психологическими обстоятельствами – это непривычно и,

следовательно, боязно. Но накопленный человечеством опыт по созданию компьютерных систем и, в частности, интеллектуальных систем позволяет этот переход сделать достаточно быстро, так как многие проблемы, возникающие при реализации и применении графодинамических моделей, имеют достаточно близкие аналоги в традиционных компьютерных системах, но многие из них могут быть решены значительно проще и элегантнее. Кроме того, в результате перехода к графодинамическим моделям "обнажается" целый ряд проблем, которые ранее просто были не видны. Эпицентром такого перехода является формализация семантики и разработка семантически совместимых языковых средств представления различных видов знаний.

Принцип 4. Семантические модели представления и обработки знаний:

В качестве формальной основы проектируемых интеллектуальных систем, в качестве основы абстрактных логико-семантических моделей интеллектуальных систем, использовать графодинамические модели специального вида – семантические модели представления и обработки знаний, в основе которых лежат **семантические сети** [Кузнецов, 1986], [Лозовский, 1984], [Плесневич, 1982], [Скороходько, 1989], [Шенк, 1980], [Sowa, 2008].

Фактически, речь идет о создании формальных средств описания семантики различных видов знаний и формальных средств описания обработки знаний на семантическом уровне.

Семантическая сеть – это графовая структура G , задаваемая пятеркой $\langle V, C, K, M, I \rangle$ и удовлетворяющая следующим требованиям, которые дополняют свойства множеств V, C, K, M, I , указанные в определении графовой структуры:

- каждая вершина $v_j \in V$ является знаком одного из объектов, описываемых семантической сетью;
- каждая ключевая вершина $k_j \in K$ является знаком соответствующего класса элементов графовой структуры G ;
- каждая метка $m_j \in M$ также является знаком соответствующего класса элементов графовой структуры G ;
- каждая пара инцидентности, принадлежащая любому отношению инцидентности $ij \in I$, является парой принадлежности, связывающей знак некоторого множества элементов семантической сети с одним из этих элементов;
- в семантической сети вершины $v_j, v_k \in V$ могут быть инцидентны друг другу, но только в том случае, если по крайней мере одна из них (например, v_k) является ключевой ($v_k \in K$), а вторая является вершиной, принадлежащей множеству, обозначаемому ключевой вершиной v_k (т.е. $v_j \in v_k$);
- каждый элемент множества I ($ij \in I$) является

- каждый связующий элемент $c_j \in C$ является знаком некоторого фрагмента графовой структуры G , а точнее знаком некоторого подмножества множества всех элементов графовой структуры G ;
- среди элементов графовой структуры G нет пар, синонимичных друг другу знаков, т.е. знаков, обозначающих один и тот же объект (одну и ту же сущность) – либо один и тот же внешний описываемый объект, либо одно и то же множество элементов графовой структуры;
- среди элементов графовой структуры G нет омонимичных знаков, которые в разных контекстах, в разных обстоятельствах могут обозначать разные сущности.

Следовательно, все (!) элементы (атомарные фрагменты) семантической сети являются знаками различных сущностей (объектов). Такими сущностями могут быть всевозможные внешние описываемые объекты, а также различные множества, состоящие их элементов (атомарных фрагментов) этой же семантической сети.

Таким образом, семантическая сеть – это абстрактная знаковая конструкция "рафинированного вида", в которой нет ничего кроме знаков и инцидентности этих знаков. В частности, в семантической сети отсутствуют элементарные незначащие фрагменты (символы), имена описываемых объектов, слова, из которых эти имена состоят, всевозможные разделители и ограничители, обеспечивающие структуризацию текста. В отличие от текстов традиционного вида, семантическая сеть имеет в общем случае нелинейный характер, поскольку каждый элемент семантической сети может быть инцидентен более чем двум другим элементам.

Семантическую сеть можно трактовать как абстрактный текст, который является семантическим инвариантом соответствующего максимального множества семантически эквивалентных текстов, принадлежащих всевозможным языкам.

На основе понятия семантической сети вводится понятие языка семантических сетей в заданном алфавите и с заданным набором ключевых узлов.

Семантические сети как модели представления знаний известны давно. Но, в отличие от фреймовых, продукционных и логических моделей, для семантических сетей не были разработаны достаточно удобные и практически используемые языки представления знаний, достаточно удобные

языки программирования, специально ориентированные на обработку семантических сетей. И, как следствие этого, не были созданы широко используемые комплексные технологии проектирования интеллектуальных систем, в основе которых лежат семантические сети. Причин тому много. Одна из них – это не совсем привычный, нетрадиционный характер таких моделей и возникший на этой основе миф о сложности их реализации. Если такие графодинамические семантические модели реализовывать "в лоб" и без ориентации на последующую аппаратную поддержку, то, конечно, это будет неэффективно. Но жизнь берет свое. И на фоне бурного развития микроэлектронных технологий подобного рода мифы выглядят все менее и менее убедительными. Более того, развитие Internet-технологий привело к необходимости формализации семантики информации, обрабатываемой в сети Internet, что вызывало бурное развитие целого направления – Semantic Web.

Чем же хороши семантические сети и в чем достоинство семантических моделей обработки информации:

- Представление знаний в виде семантических сетей позволяет существенно упростить процедуру интеграции знаний и свести эту процедуру к выявлению и склеиванию синонимичных элементов интегрируемых семантических сетей;
- Специфика обработки баз знаний заключается в том, что порождаемые (генерируемые) новые фрагменты знаний необходимо не просто построить, но и погрузить, интегрировать в текущее состояние базы знаний, т.к. в этих порождаемых фрагментах знаний могут появиться знаки, синонимичные тем, которые уже присутствуют в текущем состоянии базы знаний. Таким образом, процедура интеграции порождаемых фрагментов обрабатываемой базы знаний является процедурой, постоянно используемой в ходе обработки знаний. Следовательно, представление знаний в виде семантических сетей, благодаря упрощению процедуры интеграции знаний, позволяет упростить не только ввод новых знаний из вне, но и интеграцию в состав текущего состояния базы знаний новых знаний, порождаемых в ходе решения задач;
- База знаний интеллектуальной системы, представленная в виде корректно построенной семантической сети, полностью исключает дублирование информации в рамках такой базы знаний – каждая информация, представленная соответствующим фрагментом семантической сети, должна находиться в рамках этой семантической сети там и только там, где она должна находиться, и нигде больше;
- Представление знаний в виде семантических сетей позволяет существенно упростить процедуру ассоциативного доступа к различным

видам фрагментов хранимой базы знаний, а также существенно расширить типологию запросов (вопросов) к базе знаний;

- Семантические модели обработки знаний не просто хорошо приспособлены к поддержке параллельной асинхронной обработки информации, но и обеспечивают обмен информацией через общую графодинамическую память между различными параллельно (одновременно) протекающими (выполняемыми) процессами, что может существенно ускорить каждый из этих процессов. Примером такого взаимодействия параллельно протекающих процессов является одновременная реализация разных стратегий и тактик, направленных на поиск пути решения заданной нетривиальной задачи;
- С помощью семантических моделей представления и обработки знаний можно проинтерпретировать все известные виды моделей представления обработки знаний (фреймовые, продукционные, логические), а также все известные модели решения задач различного вида и все известные модели рассуждений. Это дает возможность рассматривать перечисленные модели не как альтернативные, а как дополняющие друг друга модели, которые могут сосуществовать в разных сочетаниях в разных интеллектуальных системах.

Семантическая модель обработки знаний представляет собой абстрактную многоагентную систему, состоящую из **абстрактной семантической памяти** в которой хранятся семантические сети, и из множества агентов, ориентированных на обработку семантических сетей, хранимых в указанной семантической памяти.

Семантическую память можно трактовать как абстрактную семантическую модель памяти интеллектуальной системы.

Семейство абстрактных агентов, работающих над семантической памятью вместе с этой семантической памятью можно трактовать как **семантическую модель решения задач**, используемую в соответствующей интеллектуальной системе, или как операционную семантику этой интеллектуальной системы. Подчеркнем, что семантическую модель обработки информации можно построить для любой компьютерной системы (как для интеллектуальной системы, так и для компьютерной системы традиционного вида), обеспечивая, тем самым, семантическую совместимость (на абстрактном уровне) не только интеллектуальных систем, но и компьютерных систем любого уровня интеллектуальности.

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.

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 (сюда, в том числе входит и описание блокировок, задаваемых разными процессами в семантической памяти).

Семантическая модель базы знаний интеллектуальной системы – это, образно говоря, формальная трактовка "семантического пространства" в котором "живет" эта интеллектуальная система, а, точнее, такого фрагмента указанного "семантического пространства", который в текущий момент указанной интеллектуальной системе известен.

В целом логико-семантическая модель интеллектуальной системы включает в себя:

- семантическую модель базы знаний этой интеллектуальной системы;
- семантическую машину обработки знаний этой интеллектуальной системы, которая, в свою очередь, состоит из:
 - семантической памяти;
 - коллектива агентов над семантической памятью.

Принцип 5. Унификация семантического представления знаний:

Определить структуру унифицированных семантических сетей, обеспечивающих представление самых различных видов знаний.

Это предполагает разработку соответствующего стандарта, выделяющего из всего многообразия абстрактных языков семантических сетей определенный базовый универсальный язык семантических сетей, который мы назвали SC-кодом (Semantic Computer code).

Для того, чтобы перейти от семантических сетей произвольного вида к текстам SC-кода уточним направления такого перехода, т.е. задаваемые нами критерии качества разрабатываемого языка семантических сетей (SC-кода). К таким критериям мы отнесем:

- (1) Переход от семантических сетей, имеющих унарные и многокомпонентные (многоместные) связки, к семантическим сетям, имеющим только бинарные связки. Такие сети будем называть бинарными семантическими сетями [Плесневич, 2008], [Карабеков и др., 2008];
- (2) Минимизация алфавита, т.е. минимизация числа меток, используемых в семантических сетях;
- (3) Универсальность разрабатываемого языка, т.е. возможность представления любых знаний в виде текстов этого языка.

Семантическую сеть G_b задаваемую пятеркой, $\langle V_b, C_b, K_b, M_b, I_b \rangle$ будем называть **бинарной семантической сетью** в том и только в том случае, если:

- все связующие элементы, входящие во множество C_b , являются бинарными связками, каждая из которых имеет только два компонента, которыми могут быть элементы множества V_b , элементы множества C_b и элементы множества K_b ;
- $I_b = \{i_1, i_2\}$, где i_1 – бинарное ориентированное отношение инцидентности, связывающее бинарную связку с её компонентом; i_2 – бинарное ориентированное отношение инцидентности, связывающее бинарную ориентированную связку с ее вторым компонентом. Очевидно, что $i_1 \subset i_2$.

Для любой семантической сети G , задаваемой пятеркой $\langle V, C, K, M, I \rangle$, можно построить семантически эквивалентную ей бинарную семантическую G_b , задаваемую пятеркой $\langle V_b, C_b, K_b, M_b, I_b \rangle$ следующим образом:

- во множество V_b включаются все вершины (V) семантической сети G и все небинарные связующие элементы (C_n) семантической сети G ($C_n \subset G$);
- $I_b = \{i_1, i_2\}$ – это отношения инцидентности связывающие бинарные связки (в том числе из множества C) с их компонентами;
- $M_b = (M \cup \{re\})$. Если в алфавит M не входила метка отношения принадлежности (re), то в алфавит M_b она дополнительно вводится;
- $K_b = K \cup (I \setminus (I_b \cup \{re\}))$. Т.е. знаки отношений инцидентности, связывающих (в рамках G) небинарные связующие элементы с их компонентами, становятся в рамках G_b ключевыми узлами;
- во множество C_b включаются (1) все бинарные связки семантической сети G , (2) все ориентированные пары отношения принадлежности (re), связывающие ключевые вершины семантической сети G с инцидентными им элементами семантической сети, (3) все ориентированные пары всех отношений инцидентности (I), связывающих небинарные связующие элементы (из множества C) с их компонентами, (4) все пары принадлежности, явно связывающие ключевые узлы семантической сети G_b , соответствующие различным отношениям инцидентности (I) семантической сети G , с парами принадлежности которые явно связывают небинарные связующие элементы с их компонентами, выполняющими в рамках этих связующих элементов роли, обозначаемые указанными отношениями инцидентности.

Нетрудно заметить, что приведение семантической сети к бинарному виду приводит также к минимизации числа отношений инцидентности.

Нетрудно также заметить, что все пары инцидентности, связывающие небинарные связующие элементы семантической сети G с их компонентами, в бинарной семантической сети C_b "превращаются" в бинарные ориентированные связки, которые принадлежат отношению принадлежности и которые, следовательно, должны быть помечены меткой re ($re \in M_b$), являющейся знаком отношения принадлежности.

Таким образом, каждый небинарный связующий элемент семантической сети G в семантической сети G_b трактуется как множество связываемых им элементов семантической сети, связь которого с его элементами представляется явно – не в виде пар инцидентности, а в виде дополнительно вводимых связок принадлежности. Эти связки принадлежности, в свою очередь, могут быть вторыми компонентами других связок принадлежности, связывающих указанные связки с ключевыми узлами, обозначающими различные роли компонентов небинарных связующих элементов исходной семантической сети.

Бинарные связки легко изображать графически (в виде линий, каждая из которых соединяет графические изображения двух связываемых элементов семантической сети). Использование только бинарных связок существенно упрощает машинное кодирование семантических сетей и, в частности, упрощает разработку специальной памяти для хранения семантических сетей.

В основе **минимизации алфавита** элементов лежит следующее свойство семантических сетей. Метки, входящие в состав алфавита элементов семантической сети, и ключевые узлы этой семантической сети семантически эквивалентны в том смысле, что просто являются синтаксически различными способами выделения (задания) различных классов элементов семантической сети. При этом заметим, что в отличие от алфавита символов линейного текста, все элементы алфавита (все метки) семантической сети, как и все ее ключевые узлы, имеют семантическую интерпретацию на описываемой предметной области. Таким образом, метки элементов семантической сети без какого-либо изменения семантики этой семантической сети можно "превращать" (преобразовывать) в ее ключевые узлы. При этом семантическая интерпретация каждого такого ключевого узла будет совпадать с семантикой соответствующей преобразованной метки.

Если мы имеем дело с корректно (правильно) построенной семантической сетью и записанной в некотором языке семантических сетей, то перевод этой семантической сети на любой другой язык семантических сетей не требует больших усилий, т.к. имеет место большое сходство синтаксического и семантического устройства всех языков семантических сетей. Фактически эти языки отличаются своими алфавитами, элементы которых, как было показано выше, легко преобразуются в ключевые узлы семантических сетей.

Итак, число меток семантической сети можно уменьшать ценой расширения множества её ключевых узлов. Вопрос в том, до какого предела это можно делать и как выглядит минимальный алфавит универсального языка семантических сетей. Универсальным языком семантических сетей будем называть такое множество семантических сетей, элементами которых являются семантические сети, представляющие любую информацию о любой описываемой предметной области.

Из сказанного ранее следует, это в состав минимального алфавита универсального языка семантических сетей, по крайней мере, должна входить метка *re*, обозначающая **отношение принадлежности**. Без этой метки невозможно описать связи ключевых узлов семантической сети с элементами обозначаемых ими классов, а также невозможно осуществить переход к бинарным семантическим сетям.

Текст, принадлежащий SC-коду, т.е. **sc-текст** (sc-структура, sc-конструкция) является семантической сетью частного вида, имеющей следующие особенности:

- все связки sc-текстов являются бинарными связками, которые будем называть **sc-коннекторами**. Неориентированные sc-коннекторы будем называть **sc-ребрами**, ориентированные – **sc-дугами**.
- множество меток элементов sc-текстов (алфавит sc-элементов, алфавит SC-кода) включает в себя:
 - метку **sc-узлов** (вершин sc-текстов);
 - метку **sc-ребер**;
 - метку **sc-дуг общего вида**;
 - метку **sc-дуг принадлежности**;
 - метку **sc-дуг основного вида**.
- множество отношений инцидентности элементов sc-текстов состоит из двух следующих отношений:
 - **быть компонентом sc-коннектора** (sc-ребра или sc-дуги);
 - **быть вторым компонентом sc-дуги**;
- множество ключевых узлов sc-кода (ключевых узлов sc-текстов) вместе с метками sc-элементов задает базовую семантическую типологию sc-элементов, т.е. базовую онтологию SC-кода.

К числу ключевых узлов SC-кода, определяющих разбиение множества sc-элементов по признаку **константности**, относятся:

- ключевой sc-узел, обозначающий множество всевозможных константных sc-элементов (**sc-констант**). Каждая sc-константа является обозначением некоторого конкретного фиксированного объекта;
- ключевой sc-узел, обозначающий множество всевозможных переменных sc-элементов (**sc-переменных**), каждая из которых обозначает некоторый произвольный, нефиксированный объект из некоторого дополнительно уточняемого множества объектов. Используются

sc-переменные в логических формулах (в т.ч. в высказываниях), в программах (в обобщенных описаниях способов решения различных классов задач), в формулировках вопросов.

К числу ключевых узлов SC-кода, определяющих разбиение множества sc-элементов по **структурному признаку**, относятся:

- ключевой sc-узел, обозначающий множество всевозможных **sc-коннекторов**, т.е. атомарных связок sc-элементов. Более детальное разбиение множества sc-коннекторов по структурному признаку осуществляется с помощью меток sc-элементов (на **sc-ребра**, **sc-дуги общего вида**, **sc-дуги принадлежности**, **sc-дуги основного вида**);
- ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторое связующее множество sc-элементов. Указанные sc-элементы будем называть **связующими sc-узлами**. К числу ключевых узлов SC-кода, определяющих более детальную структурную типологию связующих sc-узлов, относятся:
 - ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторую неатомарную связь между sc-элементами, т.е. связь, не являющуюся sc-коннектором. Такие sc-узлы будем называть **неатомарными sc-связками**. Более детальная структурная типология неатомарных sc-связок задается такими ключевыми узлами SC-кода, как быть **унарной sc-связкой**, быть **бинарной неатомарной sc-связкой**, быть **многокомпонентной sc-связкой**, быть **ориентированной неатомарной sc-связкой**, быть **неориентированной неатомарной sc-связкой**;
 - ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторую структуру из sc-элементов. Такие структуры будем называть обозначениями sc-структур или просто **sc-структурами**.
- ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторый класс sc-элементов. Такие sc-узлы будем называть обозначениями sc-понятий или, просто, **sc-понятиями**. Более детальная структурная типология sc-понятий задается следующими ключевыми узлами SC-кода: быть **отношением** (классом однотипных связок), быть **бинарным отношением**, быть **унарным отношением**, быть **многоместным отношением**, быть **ориентированным отношением**, быть **неориентированным отношением**, быть **ролевым отношением** (т.е. отношением, которое является подмножеством отношения принадлежности), быть **классом структур**, быть **классом терминальных sc-узлов** (первичных sc-узлов, которые не являются обозначениями множеств sc-элементов).

- ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторый объект, который не является множеством sc-элементов. Такие sc-узлы будем называть **терминальными sc-узлами** (первичными sc-узлами). Более детальная структурная типология терминальных sc-узлов задается следующими ключевыми узлами sc-кода:

- быть **предметным sc-узлом**, каждый из которых обозначает некоторый реальный (материальный, физический) или вымышленный внешний объект некоторой предметной области;
- быть **sc-ссылкой**, каждая из которых 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;
- быть **терминальным элементом шкалы или шаблона** (это sc-элементы, для которых трудно установить обозначаемые ими объекты, поскольку эти sc-узлы просто являются терминальными элементами каких-либо шкал, шаблонов, типовых структур, с которыми устанавливаются соответствия, с которыми сравниваются, сопоставляются различные объекты и структуры).

К числу ключевых узлов SC-кода, уточняющих семантику sc-дуг принадлежности, относятся:

- ключевые узлы SC-кода, определяющие разбиение множества sc-дуг принадлежности по признаку **позитивности**:
 - быть **sc-дугой позитивной принадлежности**;
 - быть **sc-дугой негативной принадлежности**;
 - быть **sc-дугой нечеткой принадлежности** (т.е. sc-дугой позитивность или негативность которой в текущий момент не установлена);
- ключевые узлы SC-кода, определяющие разбиение множества sc-дуг принадлежности по признаку **стационарности**:
 - быть **sc-дугой стационарной принадлежности**, семантический тип которой является постоянным, не изменяющимся во времени;
 - быть **sc-дугой нестационарной принадлежности**, семантический тип которой изменяется во времени;

Заметим, что **sc-дуги основного вида**, которые выделяются с помощью соответствующей метки, семантически трактуются как **sc-дуги позитивной стационарной принадлежности**.

Перечислим основные особенности и достоинства SC-кода.

Унифицированные семантические сети (sc-тексты) – это абстрактная семантическая модель

знаний, являющаяся инвариантом различных способов представления и кодирования этих же знаний (в том числе и самих семантических сетей). Наличие такого инварианта необходимо для решения проблемы интеграции самых различных видов знаний. На основе унифицированных семантических сетей можно строить семантические модели различных компьютерных систем и решить проблему интеграции таких систем.

SC-код является абстрактным языком в том смысле, что способ изображения (материализации) его текстов не уточняется. Следовательно, можно разрабатывать различные графические уточнения SC-кода (например, SCg-код), различные варианты изображения sc-текстов в виде строк символов (например, SCs-код), различные варианты машинного представления sc-конструкций в адресной памяти традиционных компьютеров, а также в специальной структурно перестраиваемой ассоциативной памяти будущих компьютеров, ориентированных на обработку баз знаний.

Все sc-элементы, кроме терминальных sc-узлов, являются обозначениями множеств, состоящих из sc-элементов (множеств sc-элементов). Такие sc-элементы будем называть **вторичными sc-элементами**. Из этого следует то, что SC-код имеет базовую теоретико-множественную семантическую интерпретацию.

SC-код представляет собой достаточно простой компьютерный код семантических сетей, который является не "инородным" представлением семантических сетей, а их представлением тоже в виде семантических сетей, но максимально простого вида – с минимальным алфавитом и с бинарными связками.

SC-код ориентирован на представление информации в компьютерной памяти и может рассматриваться как основа модели структурно перестраиваемой ассоциативной памяти будущих компьютеров, ориентированных на обработку семантических сетей. Т.е. SC-код можно рассматривать как универсальную основу машинного кодирования знаний в памяти будущих компьютеров, ориентированных на обработку семантических сетей. В такой памяти биты и байты "уступят место" sc-дугам, sc-ребрам и sc-узлам.

На базе SC-кода можно создавать целое семейство совместимых специализированных языков, ориентированных на представление самых разных видов знаний (логических формул и высказываний, программ, вопросов, поведенческих целей, различных видов моделей динамических систем и т.п.), таким образом, чтобы тексты всех этих специализированных языков полностью соответствовали SC-коду (т.е. были sc-текстами). Такие специализированные языки, общим носителем которых является SC-код, будем называть **sc-языками**. Каждый sc-язык определяется своим расширением множества ключевых узлов SC-кода.

SC-код представляет собой ядро **универсального открытого языка семантических сетей**, являющегося результатом интеграции всевозможных языков семантических сетей, построенных на основе SC-кода, и задаваемого:

- фиксированным алфавитом (алфавитом SC-кода);
- постоянно расширяемым (открытым) семейством ключевых узлов, в состав которого входят все ключевые узлы всех интегрируемых языков.

SC-код представляет собой **единство языка и метаязыка**. Так, например, в виде sc-конструкций можно описать синтаксис, семантику и онтологию SC-кода. С формальной точки зрения SC-код можно трактовать как метаязык базовой семантической спецификации sc-элементов с помощью специального набора ключевых узлов SC-кода.

Единство языка и метаязыка в SC-коде проявляется в том числе и на самом низком уровне – на уровне sc-дуг принадлежности $\langle si, ei \rangle$, в которых сам sc-элемент ei , а не обозначаемый им объект, является элементом множества, обозначаемого sc-узлом si .

SC-код позволяет описать структуру любой информационной конструкции, не принадлежащей SC-коду, на любом уровне (на любом этапе синтаксического и семантического анализа). В частности, первичную синтаксическую структуру любой информационной конструкции можно представить в виде изоморфной sc-конструкции. Следовательно, SC-код может быть использован в качестве метаязыка для описания любого внешнего языка, т.е. языка, тексты которого не являются sc-конструкциями.

SC-код хорошо приспособлен к использованию в условиях так называемых **не-факторов** - нестационарности, неточности, противоречивости, неактуальности знаний, а также неполноты знаний (нечеткости, несформированности множеств, несформированности внешних информационных конструкций) [Нариньяни, 1994].

Информационные конструкции SC-кода (sc-конструкции) легко **визуализируются**.

Принцип 6. Унификация структуризации баз знаний:

Трактовать семантическую структуру **базы знаний** интеллектуальной системы как отражение иерархической системы взаимосвязанных друг с другом **предметных областей**, представляемых в базе знаний. Это предполагает

- (1) уточнение понятия предметной области;
- (2) разработку языковых средств описания структуры предметных областей с помощью унифицированных семантических сетей;
- (3) разработку языковых средств описания типологии предметных областей и различных видов связей между ними.

Структуризация базы знаний, выделение в ней различных связанных между собой подструктур

необходимы по целому ряду причин. В частности, это необходимо для дидактических целей (человеку усваивающему некоторые знания, желательно иметь, своего рода оглавление или "карту" этих знаний, что позволяет планировать их усвоение и рассматривать их с различной степенью детализации), а также для организации распределения работ по проектированию баз знаний (когда разным исполнителям поручается разработка разных фрагментов базы знаний, имеющих достаточно четкие границы).

Таким образом, база знаний рассматривается нами как система взаимосвязанных между собой интегрируемых структур, которые будем называть **фрагментами базы знаний**. Связи между фрагментами базы знаний могут быть самыми различными. Каждый фрагмент и вся база знаний в целом может иметь несколько вариантов декомпозиции на подфрагменты (частные фрагменты).

По структурно-семантическому принципу можно выделить следующие типы фрагментов баз знаний:

- база фактов некоторой предметной области, которую, сокращенно, будем называть просто **предметной областью** и которая представляет собой результат интеграции всех известных в текущий момент фактографических высказываний, являющихся истинными для указанной предметной области;
- **иерархическая система нескольких предметных областей**, которые нецелесообразно объединять (интегрировать) в одну предметную область, так как в результате этого получается "сборная солянка";
- **семантическая окрестность заданного объекта**;
- **связная семантическая окрестность конечного множества заданных объектов**, представляющая сравнение (сравнительный анализ) и связи объектов из заданного конечного множества, т. е. описывающая сходства (аналогии), отличия заданных объектов, а также "близкие" связи между ними.

При структуризации базы знаний некоторым её фрагментам приписывается статус **разделов базы знаний**, которые именуются, нумеруются и входят в состав (оглавление) базы знаний.

Рассмотрим формальное уточнение понятия **предметной области** с помощью SC-кода. Если в рамках **sc-модели базы знаний** явно вводится некоторая предметная область, то она трактуется как некоторая sc-структура, для которой в базе знаний явно вводится обозначающий её sc-узел, который, в свою очередь, связывается входящей в него sc-дугой основного вида с sc-узлом, обозначающим класс sc-структур, являющихся предметными областями. После этого в указанной sc-структуре необходимо явно задать роли

некоторых узлов, входящих в состав этой sc-структуры. К числу таких ролей относятся:

- быть **максимальным классом исследуемых объектов**, т. е. множеством всех исследуемых объектов и только их. В каждой предметной области существует только один ключевой узел, выполняющий такую роль;
- быть **классом исследуемых объектов**. Каждая предметная область может иметь любое число таких классов;
- быть **классом вторичных объектов**, построенных на основе исследуемых;
- быть **классом вспомогательных объектов**, через связи с которыми описываются некоторые характеристики исследуемых объектов;
- быть **отношением, каждая связка которого связывает только исследуемые объекты** или вторичные объекты, построенные на основе исследуемых;
- быть **отношением, каждая связка которого связывает исследуемые объекты со вспомогательными**.

Такое явное указание ролей ключевых элементов предметных областей есть не что иное, как их семантическая спецификация, уточняющая то, какие объекты описываются (исследуются) в данной предметной области, и о каких характеристиках, связях исследуемых объектов в данной предметной области идет речь.

Можно говорить о достаточно богатой типологии предметных областей. В частности, 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 сеть заданного фрагмента базы знаний;
- текст формальной теории, описывающей свойства и закономерности заданной предметной области. Классами объектов исследования такой предметной метаобласти являются: (1) класс логических формул и, в частности, высказываний интерпретируемых на заданной предметной области, (2) класс элементов заданной предметной области, используемых в качестве констант в указанных логических формулах, (3) класс переменных, используемых в указанных логических формулах и возможными значениями которых являются соответствующие элементы заданной 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;
- логическая система фрагментов баз знаний, связывающая каждый фрагмент базы знаний с теми фрагментами, в которых (1) даются определения понятий, используемых в заданном фрагменте и (2) вводятся и доказываются используемые утверждения. На основании такой логической системы строятся различные варианты последовательности изучения (прочтения) разделов баз знаний.

Заметим, что некоторым из перечисленных классов предметных областей может соответствовать одинаковый (унифицированный, фиксируемый) набор используемых в них ключевых понятий. Унификация (стандартизация) таких наборов понятий является важнейшим средством более глубокой семантической совместимости (интегрируемости) различных фрагментов базы знаний. Результатом такой унификации фактически является разработка средств SC-кода, ориентированных на представление предметных областей соответствующего класса. Такие языковые средства будем называть специализированным **sc-языком**.

К числу таких специализированных sc-языков можно отнести:

- **Теоретико-множественный sc-язык**, обеспечивающий описание теоретико-множественных характеристик и связей заданного семейства объектов. С помощью такого языка, в частности, могут быть представлены предметные области, являющиеся теоретико-множественными онтологиями других предметных областей;
- **Терминологический sc-язык**, обеспечивающий построение терминологических сетей;
- **Логический sc-язык**, обеспечивающий построение sc-текстов формальных теорий.

Таким образом, **SC-код** является ядром целого семейства самых различных **sc-языков**, ориентированных на описание различных классов предметных областей, в каждый из которых входят предметные области с разными множествами исследуемых объектов, но с одинаковыми предметами исследования.

Каждому такому специализированному sc-языку ставится в соответствие **множество ключевых узлов**, обозначающих различные классы

исследуемых объектов, различные отношения и алгебраические операции, заданные на множестве исследуемых объектов.

SC-язык, являющийся объединением всевозможных специализированных sc-языков будем называть **языком SCK** (Semantic Code Knowledge). Этот язык рассматривается нами как интегрированный язык представления знаний. Язык SCK является открытым (расширяемым) языком, поскольку его всегда можно пополнить новым sc-языком, описывающим структуры нового вида предметных областей.

Построение **семантической структуры базы знаний** интеллектуальной системы требует не только явного представления спецификаций каждой описываемой предметной области в виде sc-текста, но и явного описания всевозможных связей между этими предметными областями.

Переходя к рассмотрению отношений, заданных на множестве предметных областей, мы фактически переходим к некоторой предметной метаобласти, объектами исследования которой являются всевозможные предметные области (в том числе и сама эта предметная метаобласть).

Обобщая понятия гомоморфизма и изоморфизма алгебраических систем, можно говорить о гомоморфизме и изоморфизме предметных областей, что дает хорошую основу для выявления глубоких нетривиальных аналогий между предметными областями.

Различные предметные области могут пересекаться. То есть элементы одной предметной области могут быть также и элементами другой предметной области. При этом возможны самые различные варианты такого пересечения. Это может быть строгое пересечение, строгое включение. Общие элементы пересекающихся предметных областей могут в рамках этих областей выполнять как одинаковые, так и разные роли. Так, например, первичные элементы одной предметной области могут входить в состав другой предметной области в качестве вторичных элементов, в качестве ключевых элементов. Объекты исследования одной предметной области могут входить в состав другой предметной области и в качестве вспомогательных элементов.

В качестве примеров отношений, заданных на множестве предметных областей рассмотрим несколько вариантов выделения частных предметных областей:

- выделение частной предметной области, на основе выделения подмножества из максимального класса исследуемых объектов. Таким способом из предметной области геометрии, объектами исследования которой являются геометрические точки, фигуры и семейства фигур, можно выделить (1) предметную область планиметрии, изучающую планарные фигуры и планарные семейства фигур

(т. е. семейства фигур, лежащих в одной плоскости) и (2) предметную область стереометрии, которая изучает непланарные фигуры и непланарные семейства фигур, которые могут состоять как из непланарных, так и из планарных фигур;

- выделение частной предметной области, на основе выделения подмножества из семейства классов исследуемых объектов. Таким способом из предметной области, изучающей треугольники, можно выделить (1) предметную область, изучающую остроугольные, тупоугольные и прямоугольные треугольники, а также (2) предметную область, изучающую равносторонние, разносторонние и равнобедренные треугольники;
- выделение частной предметной области, на основе выделения подмножества из семейства отношений, заданных на исследуемых объектах. Таким способом из геометрической предметной области можно выделить (1) предметную область, объектами исследования которой являются геометрические фигуры, а предметом исследования – их числовые характеристики и (2) предметную область, объектами исследования которой являются геометрические фигуры, а предметом исследования – различные виды их конгруэнтности (движений).

Предметная область позволяет рассматривать исследуемые объекты на разных уровнях детализации. Детализацию рассмотрения исследуемых объектов можно осуществлять как в рамках исходной (заданной) предметной области, расширяя эту предметную область в соответствующих направлениях, а можно переходить к системе самостоятельных, но связанных между собой предметных областей.

Первым и важнейшим этапом проектирования базы знаний является уточнение структуры описываемой предметной области или нескольких взаимосвязанных предметных областей. Уточнение такой структуры – это, прежде всего, уточнение класса исследуемых объектов, уточнение предмета исследования, уточнение всего семейства ключевых узлов семантической сети, представляющей предметную область. При этом для заданного класса исследуемых объектов и заданного предмета исследования можно построить более качественную и менее качественную предметную область.

Рассмотрим еще один тип фрагментов баз знаний – **семантические окрестности**. В общем случае семантическая окрестность заданного объекта – это описание некоторых, числовых характеристик, свойств и связей заданного объекта. Частными видами семантических окрестностей являются:

- описание характеристик, свойств или связей заданного объекта, однозначно (!) определяющих (устанавливающих) заданный объект (для

понятий – это определение или любое другое высказывание определяющего типа);

- полная (интегрированная) семантическая окрестность заданного объекта, содержащая все известные в **текущий момент** сведения об этом объекте в рамках заданного раздела базы знаний или в рамках всего текущего состояния базы знаний;
- описание числовых характеристик (параметров, признаков) заданного объекта;
- описание свойств заданного объекта (это те истинные нефактографические высказывания, в которых знак этого объекта используется в качестве константы);
- описание теоретико-множественных связей заданного объекта с другими объектами;
- описание разноязычных терминов, иероглифов, пиктограмм, используемых для внешней идентификации заданного объекта (в т. ч. и описание происхождения этих идентификаторов);
- перечень отношений, соответствующих заданному объекту (отношений, в области определения которых заданный объект входит в качестве элемента; отношений, в области определения которых заданный объект входит в качестве подмножества; отношений, для областей определения которых заданный объект является надмножеством; отношений, область определения каждого из которых строго пересекается с заданным объектом).

В **sc-модели базы знаний** каждая семантическая окрестность представляется в виде соответствующего sc-текста. Для явного введения (задания) этой семантической окрестности в рамках sc-модели базы знаний необходимо:

- (1) явно ввести sc-узел, обозначающий эту семантическую окрестность;
- (2) явно связать введенный sc-узел sc-дугами основного вида со всеми элементами обозначаемого им sc-текста;
- (3) с помощью ролевого отношения **быть центром семантической окрестности** явно указать центральный элемент семантической окрестности;
- (4) явно связать введенный sc-узел sc-дугой основного вида со специальным sc-узлом, обозначающим **класс sc-структур, являющихся семантическими окрестностями**.

После этого можно описывать различные характеристики (в частности, типологию) явно введенной семантической окрестности, а также различные связи этой семантической окрестности с другими фрагментами базы знаний.

Технология проектирования унифицированных семантических моделей баз знаний рассмотрена в работе [Ивашенко, 2012].

Принцип 7. Графовые языки программирования:

Для описания способов решения задач и поведения агентов над общей графодинамической памятью использовать **графовые языки программирования**, которые ориентированы на обработку унифицированных семантических сетей и программы которых сами являются унифицированными семантическими сетями.

Если все используемые в интеллектуальной системе графовые языки программирования привести к общему унифицированному стандарту – к SC-коду (это требует представления в виде sc-текстов не только самих программ, но и обрабатываемых ими данных), то можно достаточно эффективно решать проблему формализации семантической совместимости программ, написанных не только на одном, но и на разных языках программирования.

В традиционных языках программирования синтаксическая структура и семантика хранящихся в памяти обрабатываемых данных отдается на откуп программисту, в результате чего это делается по принципу "кто во что горазд". О какой же семантической совместимости программ после этого можно говорить. В этом смысле традиционные языки программирования "хромают на одну ногу".

Наряду с применением в интеллектуальных системах множества sc-языков самого различного назначения, востребованным является использование целого семейства совместимых sc-языков программирования, которые могут иметь разный уровень, могут быть последовательными, процедурными и декларативными.

Важнейшей особенностью всех этих языков является использование ассоциативного доступа к обрабатываемым фрагментам хранимого в графодинамической памяти sc-текста.

Операционная семантика каждого такого графового языка программирования (точнее, sc-языка программирования) задается коллективом агентов над общей графодинамической памятью, которые обеспечивают интерпретацию любой программы указанного языка программирования, хранящейся вместе с обрабатываемой информацией в указанной графодинамической памяти.

Программы представленные в виде семантической сети и описывающие обработку семантических сетей, а также соответствующие им языки программирования фактически открывают новую страницу в теории программирования, которую можно назвать семантической теорией программ и языков программирования.

Основным лейтмотивом такой теории должно

быть обеспечение семантической совместимости программ и языков программирования.

Принцип 8. Унификация формального описания агентов, работающих над семантической памятью:

Из всех используемых в интеллектуальной системе графовых языков программирования (а, точнее, из всех sc-языков программирования) выделить **базовый графовый язык программирования**, ориентированный на описание агентов, работающих над общей графодинамической памятью, в которой хранятся и обрабатываются унифицированные семантические сети.

Выделение базового sc-языка программирования предназначено для унификации формального описания поведения агентов, работающих над общей графодинамической памятью.

Такой базовый sc-язык программирования будем называть **языком SCP** (Semantic Code Programming), а написанные на нем программы будем называть **scp-программами**.

Перечислим основные особенности языка SCP:

- язык SCP относится к классу **графовых языков программирования**;
- язык SCP ориентирован на обработку **унифицированных семантических сетей** (sc-текстов), хранимых в семантической памяти;
- программы языка SCP представляются также в виде унифицированных семантических сетей (sc-текстов), т. е. язык SCP принадлежит классу sc-языков;
- язык SCP ориентирован на описание **параллельной асинхронной обработки** sc-текстов, хранимых в семантической памяти;
- язык SCP использует **ассоциативный доступ** к фрагментам обрабатываемых sc-текстов;
- язык SCP является процедурным языком программирования низкого уровня, предназначенным для описания поведения агентов, работающих над семантической памятью;
- уникальной особенностью языка SCP является то, что на нем можно писать **реконфигурируемые программы**, т. е. программы, которые в процессе своего выполнения могут изменять сами себя (удалять или порождать операторы, корректировать порядок их выполнения и т. п.). Такая особенность языка SCP обусловлена не только тем, что scp-программы и обрабатываемые ими данные хранятся в общей памяти, но и тем, что они принадлежат одному и тому же, базовому языку (SC-коду), имеющему четко заданную семантическую интерпретацию.

Более подробно графовый язык программирования SCP и технология

проектирования scp-программ рассмотрены в работах [Голенков и др., 2001], [Гулякина и др., 2012].

Принцип 9. Унификация семантических моделей обработки знаний:

На основе унифицированных семантических сетей (sc-текстов) уточнить понятие **унифицированной модели обработки информации**, а также понятие унифицированной модели решения задач.

Все указанные абстрактные модели будем называть **sc-моделями обработки знаний** или **sc-машинами**, поскольку в основе их лежит использование SC-кода. Каждая такая модель (sc-машина) представляет собой многоагентную систему, состоящую:

- (1) из графодинамической памяти, в которой хранятся и обрабатываются тексты SC-кода – такую память будем называть **sc-памятью**;
- (2) из коллектива агентов, работающих над общей для них sc-памятью и взаимодействующих между собой только через эту память – такие агенты будем называть **sc-агентами**.

Очевидно, что sc-модели обработки знаний являются частным, унифицированным видом графодинамических моделей параллельной асинхронной обработки информации.

Каждый sc-агент реагирует на соответствующий ему класс ситуаций и/или событий, происходящих в sc-памяти, и осуществляет определенное преобразование sc-текста, находящегося в семантической окрестности обрабатываемой ситуации и/или события. Типология sc-агентов достаточно богата. В частности, можно выделить следующие классы sc-агентов:

- sc-агенты, обеспечивающие интерпретацию программ различных sc-языков программирования высокого уровня;
- sc-агенты информационного поиска;
- sc-агенты, обеспечивающие реализацию правил логического вывода, соответствующих самым различным логическим исчислениям;
- sc-агенты сведения задач к подзадам;
- sc-агенты анализа качества хранимой базы знаний, в частности, ее корректности, полноты;
- sc-агенты обнаружения и автоматического склеивания синонимичных sc-элементов;
- sc-агенты автоматического устранения некоторых ошибок в базе знаний;
- sc-агенты удаления информационного мусора (в частности, удаления фрагментов базы знаний, которые редко востребованы и могут быть достаточно легко восстановлены в случае их отсутствия);
- sc-агенты, обеспечивающие трансляцию вводимой информации с различных внешних языков в SC-код;
- sc-агенты, обеспечивающие трансляцию sc-текстов, вводимых пользователю на различные внешние языки;
- рецепторные sc-агенты;

- эффекторные sc-агенты.

В понятии sc-машины набор агентов не задается, т.е. могут существовать разные sc-машины с разным набором sc-агентов. Несколько разных sc-машин можно **интегрировать**. С формальной точки зрения это сделать не очень сложно:

- (1) интегрировать sc-текст, описывающий текущее состояние взаимодействия sc-агентов одной sc-машины, с аналогичным sc-текстом другой sc-машины;
- (2) полученный интегрированный sc-текст поместить в sc-память интегрированной sc-машины;
- (3) в интегрированную sc-машину включить все sc-агенты первой интегрируемой sc-машины и все sc-агенты второй интегрируемой sc-машины.

Более того, одна sc-машина может **интерпретировать** другую. Т.е. при интерпретации sc-машин можно не выходить за пределы класса sc-машин. Для этого необходима разработка целого семейства sc-языков программирования различного уровня. Тексты (программы) всех этих языков должны храниться в sc-памяти, т.е. должны быть семантическими сетями, представленными в SC-коде. Операционная семантика (интерпретация) каждого из этих языков задается определенным набором sc-агентов, процедура выполнения (поведения) каждой из которых описывается программой, написанной на языке более низкого уровня.

В абстрактных sc-машинах можно выделить следующие языки программирования:

- Семейство sc-языков программирования высокого и сверхвысокого уровня (как процедурных, так и не процедурных). Тексты программ этих языков хранятся в базе знаний интеллектуальной системы и описывают способы решения различных классов задач в соответствующих предметных областях.
- Базовый sc-язык программирования (язык SCP), на котором описываются sc-агенты и интерпретации sc-языков программирования высокого и сверхвысокого уровня, а также sc-операции, обеспечивающие интерпретацию различных логических исчислений, различных моделей интеллектуального решения задач.
- SC-язык программирования, на котором описываются sc-агенты интерпретации базового sc-языка программирования. Фактически, это 1-й язык микропрограмм для **sc-компьютера**, обеспечивающего аппаратную интерпретацию базового sc-языка программирования (языка SCP).
- При необходимости можно ввести 2-й язык микропрограммирования, описывающий интерпретацию 1-го и т.д.

На основе понятия абстрактной sc-машины можно уточнить понятие унифицированной логико-семантической модели интеллектуальной системы.

Такую унифицированную модель интеллектуальной системы будем называть абстрактной **sc-моделью интеллектуальной системы** или, сокращенно, **sc-системой**. Абстрактная sc-система включает в себя:

- интегрированную базу всех (!) знаний, которые необходимы для функционирования интеллектуальной системы и которые представлены в виде интегрированного sc-текста (такую семантическую модель базы знаний будем называть **sc-моделью базы знаний** или sc-текстом базы знаний);
- абстрактную **sc-машину**, в памяти которой хранится указанный sc-текст базы знаний.

Нетрудно заметить, что sc-текст базы знаний интеллектуальной системы является формальным и унифицированным уточнением того, что должна знать проектируемая интеллектуальная система, а sc-машина интеллектуальной системы и, в первую очередь, набор входящих в ее состав sc-агентов является формальным и унифицированным уточнением того, что должна проектируемая интеллектуальная система уметь делать со своими знаниями.

Подчеркнем также, что четкое выделение абстрактного семантического уровня интеллектуальной системы позволяет не только обеспечивать их семантическую совместимость интеллектуальных систем, но и сформировать критерии сравнения интеллектуальных систем по уровню их возможностей. Очевидно, что уровень возможностей интеллектуальной системы определяется качеством (корректностью, полнотой, многообразием) ее знаний и эффективностью ее умений (т.е. эффективностью используемых ее моделей решения задач).

Заметим также, что абстрактную логико-семантическую модель в принципе можно построить для любой компьютерной системы (как для интеллектуальной системы, так и для компьютерной системы традиционного вида), обеспечивая их семантическую совместимость на абстрактном логико-семантическом уровне.

Принцип 10. Унификация семантических моделей информационного поиска:

На основе унифицированных семантических сетей (т.е. на основе SC-кода) обеспечить построение **унифицированных семантических моделей информационного поиска** (унифицированных семантических моделей ассоциативного доступа к информации, хранимой в семантической памяти).

Ассоциативный доступ – это доступ, основанный не (!) на знании того, где находится искомая (требуемая) информация (в частности, на знании адреса или имени соответствующей области памяти), а на знании того, как искомая информация связана с известной информацией, хранимой в памяти, т.е. на знании некоторой спецификации

искомой информации.

Эффективность организации информационного поиска в базе знаний интеллектуальной системы во многом определяет эффективность самой интеллектуальной системы. Это обусловлено тем, что время, затрачиваемое интеллектуальной системой на поиск нужных в текущий момент знаний и навыков, занимает, мягко говоря, не меньше половины времени затрачиваемого на решение задачи в целом.

Унифицированная семантическая модель информационного поиска, которую будем называть **sc-моделью информационного поиска**, включает в себя:

- (1) **SC-язык вопросов**, с помощью которого в виде sc-текстов осуществляется описание (спецификация) запрашиваемых (искомых) фрагментов всего интегрированного sc-текста, хранимого в текущий момент в sc-памяти (т.е. sc-текста, который является sc-моделью базы знаний). Тексты, принадлежащие SC-языку вопросов, будем называть **sc-вопросами**.
- (2) **SC-язык оформления ответов**, с помощью которого осуществляется явное выделение sc-текстов, являющихся ответами, и явное описание их связи с явно выделенными sc-текстами, которые представляют вопросы, соответствующий указанным ответам.
- (3) Семейство информационно-поисковых sc-агентов, каждый из которых реагирует на соответствующий ему тип sc-вопроса (который при этом должен быть инициирован) и выполняет соответствующую поисковую процедуру в sc-памяти.

Семантическая типология вопросов является предметом отдельного рассмотрения. Приведем фрагмент такой типологии, чтобы проиллюстрировать семантическую мощность sc-языка вопросов.

Прежде всего, по аналогии с логическими формулами множество вопросов разбивается на:

- **атомарные вопросы**;
- **неатомарные вопросы**, каждый из которых представляет собой конечное множество вопросов.

Компонентами неатомарного вопроса могут быть как атомарные, так и неатомарные вопросы. При этом, если построить оргграф, вершинами которого будут знаки всех вопросов, входящих в состав заданного неатомарного вопроса, а дуги которого будут связывать знаки неатомарных вопросов, входящих в состав заданного неатомарного вопроса, с их компонентами, то этот оргграф будет деревом, все конечные вершины которого являются знаками атомарных вопросов. Частным видом неатомарного вопроса является конъюнктивный вопрос, ответом на который является конъюнкция (интеграция) ответов на все

вопросы, являющиеся компонентами этого конъюнктивного вопроса.

Поскольку в общем случае вопросу может соответствовать несколько правильных ответов (т.е. ответов, удовлетворяющих, релевантных, соответствующих заданному вопросу), множество вопросов разбивается на:

- вопросы, запрашивающие все правильные ответы;
- вопросы, запрашивающие один (или по крайней мере один) правильный ответ;
- вопросы, запрашивающие несколько разнообразных правильных ответов;
- вопросы, запрашивающие точно указанное число (большее единицы) правильных ответов.

Специальным видом неатомарных вопросов являются **сколько-вопросы**, запрашивающие не сами правильные ответы некоторого вопроса (который может быть как атомарным, так и неатомарным, и который является единственным компонентом сколько-вопроса), а количество таких правильных ответов.

Приведем некоторые типы атомарных вопросов:

- **какой-вопрос атомарного вида**. Каждый такой вопрос запрашивает фрагменты базы знаний, изоморфные заданному **образцу**, который может иметь произвольный размер, произвольную конфигурацию и который может быть представлен не только логической формулой существования в которой квантор существования действует на конъюнкцию атомарных логических формул, но также и логической формулой существования, в которой квантор существования действует на логическую формулу произвольного вида. Суть атомарного **какой-вопроса** заключается в поиске знаков таких объектов, которые заданным образом связаны с другими известными и неизвестными (искомыми) объектами, т.е. в поиске знаков таких объектов, которые удовлетворяют заданным требованиям. На основе **какой-вопросов** атомарного вида строится важный класс неатомарных вопросов ключевыми компонентами которых являются атомарные **какой-вопросы**, а остальными компонентами – вопросы любого вида, в формулировках которых используются переменные, входящие в состав соответствующих ключевых **какой-вопросов**;
- **запрос всех элементов** заданного конечного множества (чаще всего – это множество из элементов некоторой структуры);
- **запрос внешней информационной конструкции**, представленной некоторым файлом в том или ином формате;
- запрос полного текста заданного высказывания;
- **ли-вопрос**, запрашивающий факт истинности или ложности заданного высказывания в рамках заданной формальной теории;

- **вопрос выбора альтернатив**, запрашивающий одно или несколько истинных высказываний из заданного множества высказываний;
- **почему-вопрос**, запрашивающий обоснование (доказательство) истинности заданного высказывания;
- **что-это-вопрос**, запрашивающий основные сведения об указываемом объекте. Фактически, речь идет о выделении из базы знаний семантической окрестности, "центром" которой является знак указываемого объекта. Таким объектом может быть все, что угодно – понятие, предметная область, формальная теория, высказывание, любая структура, материальный объект.
- **запрос общих свойств** объектов, принадлежащих заданному классу;
- **запрос идентифицирующих признаков** заданного объекта. Здесь запрашиваются фрагменты базы знаний, каждый из которых однозначно (!) определяет (устанавливает, идентифицирует) заданный объект. Если заданным объектом является понятие, то таким идентифицирующим признаком является либо определение этого понятия, либо соответствующая теорема о необходимости и достаточности;
- **запрос связей между заданными объектами**;
- **запрос сравнительного анализа заданных объектов**;
- **запрос сходств** заданных объектов (сходства, аналогии – это частный вид связей между объектами);
- **запрос отличий** заданных объектов (отличия объектов – это тоже частный вид связей между ними);
- **запрос плана решения** заданной конкретной задачи, т.е. плана достижения заданной цели в заданных конкретных условиях;
- **запрос обобщенного способа решения** любой задачи из заданного класса задач. Таким обобщенным способом может быть алгоритм, декларативная (непроцедурная) программа, нестрогое предписание (рекомендация);
- **зачем-вопрос**, запрашивающий то, какой надцели соответствует заданная цель, которая может быть сформулирована как в декларативной, так и в процедурной форме.

Список типов атомарных вопросов можно продолжить, но почти все они будут подтипами (подмножествами) перечисленных типов вопросов. В основе sc-языка вопросов лежит построение онтологии вопросов, в рамках которой четко прописываются все теоретико-множественные (и, в первую очередь, родо-видовые) связи между всеми выделенными типами и подтипами вопросов. При

этом в формулировке каждого конкретного sc-вопроса явным образом отражаются иерархия всех типов вопросов, которым принадлежит данный конкретный sc-вопрос. Для этого каждому типу вопросов ставится в соответствие ключевой sc-узел, обозначающий этот тип вопросов.

В заключение заметим, что в **SC-языке оформления ответов** кроме отношения релевантности, связывающего вопросы с правильными на него ответами, используются языковые средства, описывающие качество, полноту ответов. Это вызвано тем, что некоторые типы вопросов предполагают наличие целого множества правильных ответов, но разного качества, с разной степенью полноты.

Принцип 11. Унификация семантических моделей интеграции знаний и семантических моделей интеграции целых интеллектуальных систем:

На основе унифицированных семантических сетей обеспечить построение **унифицированных семантических моделей интеграции знаний** (понимания знаний) и использовать эти модели (1) как основу процесса приобретения интеллектуальной системой новых знаний как со стороны конечных пользователей, так и со стороны разработчиков; (2) как основу интеграции программ и различных семантических моделей расширения задач; (3) как основу интеграции абстрактных логико-семантических моделей интеллектуальных систем.

Главное свойство интеллектуальной системы – не те интеллектуальные знания и навыки, не те интеллектуальные способности, которые она имеет в текущий момент, а метаспособность приобретать любые необходимые ей новые знания и навыки. А для этого интеллектуальная система, как минимум, должна уметь интегрировать эти приобретаемые знания и навыки. Следовательно, проблема формализации интеграции знаний и навыков является центральной для деятельности интеллектуальных систем.

Принципиальное свойство интеграции двух интеллектуальных систем заключается в следующем. Пусть имеется две интеллектуальные системы $s1$, $s2$, первая из которых способна решать задачи из множества $q1$, в вторая – $q2$. В результате простого соединения этих систем мы получаем систему, которая способна решать задачи из множества $(q1 \cup q2)$. Тогда как в результате интеграции мы получаем систему, которая способна решать значительно большее число задач, чем $(q1 \cup q2)$. Такое расширение числа решаемых задач происходит за счет тех задач, для решения которых некоторые (но не все) знания и навыки находятся в системе $s1$, а другие – в системе $s2$.

Таким образом, при интеграции

интеллектуальных систем происходит приобретение нового качества "на стыке" интегрируемых систем, когда для решения некоторых задач одна часть необходимых знаний и/или умений находится в одной интегрируемой системе, а другая часть – в другой системе.

Процесс интеграции двух семантических сетей рассмотрим как систему следующих взаимодействующих подпроцессов, некоторые из которых могут выполняться параллельно:

- приведение интегрируемых семантических сетей к унифицированному виду, т.е. представление (запись) их в SC-коде;
- согласование ключевых узлов и онтологий, используемых в интегрируемых sc-текстах. Очевидно, что полностью автоматизировать такое согласование невозможно, поэтому разработчикам интегрируемых фрагментов баз знаний и целых баз знаний необходимо уметь договариваться друг с другом;
- выделение в интегрируемых sc-текстах таких sc-элементов, которые имеют глобальные (уникальные) идентификаторы (внешние имена);
- выделение в интегрируемых sc-текстах sc-элементов, имеющих локальные идентификаторы вместе с областью действия каждого такого идентификатора. Область действия локального идентификатора – это такой фрагмент базы знаний, в рамках которого разные sc-элементы, имеющие этот локальный идентификатор, считаются синонимичными;
- склеивание sc-элементов, имеющих одинаковые глобальные идентификаторы;
- склеивание sc-элементов, имеющих одинаковые локальные идентификаторы, если каждый из этих sc-элементов принадлежит области действия своего локального идентификатора и области действия локального идентификатора другого sc-элемента;
- склеивание sc-элементов на основании однозначности используемых алгебраических операций;
- склеивание sc-элементов на основании логических высказываний о существовании единственности;
- склеивание кратных связей, принадлежащих отношениям:
 - не имеющим кратных связей;
 - имеющих кратные связи, но не для заданных типов компонентов (например, кратные связи принадлежности не могут выходить из знаков канторовских множеств).

Таким образом, интеграция семантических сетей, т.е. процесс погружения (понимания) одной семантической сети в другую – это нетривиальный процесс рассуждений, направленный на выявление пар синонимичных элементов семантической сети

на основе определенных знаний, имеющихся в базе знаний интеллектуальной системы.

От унифицированной семантической модели интеграции знаний (точнее, sc-текстов) можно достаточно легко перейти к интеграции sc-моделей интеллектуальных систем, поскольку после интеграции sc-моделей баз знаний интегрируемых интеллектуальных систем интеграция соответствующих им наборов sc-агентов сводится к простому теоретико-множественному объединению указанных множеств sc-агентов.

Принцип 12. Унификация и интеграция различных семантических моделей решения задач:

Обеспечить в рамках проектируемой интеллектуальной системы использование не только самых различных видов знаний, но и самых различных моделей и стратегий решения задач.

Для этого необходимо акцентировать внимание не столько на разработку новых моделей решения задач, сколько на унификацию и интеграцию в рамках проектируемых интеллектуальных систем уже разработанных и хорошо зарекомендовавших себя моделей (дедуктивных, индуктивных, абдуктивных, четких, нечетких, универсальных, специализированных...). Подчеркнем то, что в разных проектируемых интеллектуальных системах могут быть востребованы самые разные сочетания известных моделей и стратегий решения задач. Подавляющее число моделей представления знаний и решения задач не являются альтернативными и дополняют друг друга. Не составляют исключение и такие классы моделей, как фреймовые, логические, продукционные.

Рассмотренное выше понятие **вопроса** и его формализация является основой не только для информационно-поисковых моделей, но и для самых различных моделей решения задач. С точки зрения решателя задач вопрос – это **непроцедурная формулировка информационной цели**, т.е. декларативная формулировка некоторой информационной цели, которая описывает спецификацию (свойства) той информации, которую требуется либо найти, если она уже присутствует в текущем состоянии базы знаний, либо построить (сгенерировать, вывести), если она отсутствует в текущем состоянии памяти. Таким образом, вопрос можно считать описанием целевого (требуемого) состояния обрабатываемой базы знаний (а, точнее, определенного фрагмента этой базы знаний). Вопрос также можно считать одним из видов метазнаний, описывающих (специфицирующих) наше незнание, т.е. наше знание о том, что мы не знаем, но хотели бы знать.

Вопросы могут инициироваться (задаваться) как пользователями, так и самой системой. Это означает, что в процессе обработки информации интеллектуальная система сама себе может задавать (генерировать, порождать) вопросы. Если

инициирован некоторый sc-вопрос, то сначала активизируются соответствующие агенты информационного поиска в "надежде" на то, что запрашиваемый ответ (или ответы) на указанный sc-вопрос уже присутствует в текущем состоянии базы знаний. И только после того, как информационно-поисковые sc-агенты обнаружат отсутствие ответа в текущем состоянии базы знаний, начинается работа решателя задач, направленная на генерацию (построение, порождение, вывод) требуемого ответа.

Кроме вопроса используется также и **процедурная формулировка информационной цели** – это описание (спецификация) некоторого действия, которое требуется выполнить и которое направлено на преобразование (изменение состояния) базы знаний, хранимой в некоторой памяти. Указанное действие, выполняется либо одним sc-агентом (в случае, если это элементарное действие над sc-памятью), либо несколькими sc-агентами и порождает определенное событие (изменение состояния sc-памяти).

Для унификации различных моделей решения задач необходимо уточнить не только понятие **информационной цели**, но и понятие **информационной задачи**. Информационная задача задается (1) формулировкой информационной цели, т.е. описанием того, что требуется, и (2) той хранимой в памяти информацией, которая семантически связана с заданной информационной целью, является контекстом этой информационной цели, т.е. тем, что дано. В пределе, контекстом информационной цели можно считать текущее состояние всей хранимой базы знаний.

Формальное рассмотрение контекстов различных информационных задач требует разработки специальных языковых средств, предназначенных для описания текущего состояния хранимой базы знаний, а, точнее, для описания "границ" между тем, что в текущем состоянии базы знаний известно и тем, что неизвестно. К числу таких языковых средств, в частности, относятся следующие ключевые узлы, являющиеся знаками нестационарных множеств (т.е. множеств, которые в разные моменты времени могут иметь разные элементы):

- быть sc-дугой нечеткой принадлежности (такая sc-дуга связывает sc-узел, обозначающий некоторое множество, с sc-элементом о котором нам в текущий момент времени не известно, принадлежит он указанному множеству или нет);
- быть построенным конечным множеством (у каждого такого множества в текущем состоянии базы знаний известны и явно указаны все его элементы);
- быть построенным высказыванием (для каждого такого высказывания в текущем состоянии базы знаний представлен не только его знак, но и полный текст);

- быть построенной внешней информационной конструкцией (файлом);
- быть аксиоматизированной формальной теорией;
- быть построенным рассуждением (обоснованием, доказательством, решением);
- быть построенной программой;

Более подробно унифицированные семантические модели решения задач и технология их проектирования рассмотрены в работе [Заливако и др., 2012].

Принцип 13. Унификация визуализации семантических сетей:

В качестве основы организации графического пользовательского интерфейса использовать язык унифицированного визуального представления абстрактных унифицированных семантических сетей в виде, близком к изоморфному.

Указанный язык графического изображения sc-текстов назван SCg-кодом (Semantic Code graphical). Подчеркнем, что следует четко отличать язык абстрактных унифицированных семантических сетей (SC-код), который абстрагируется от того, как должны быть физически представлены узлы и коннекторы текстов этого языка (sc-текстов), и язык графического изображения таких семантических сетей. Т.е. абстрактная семантическая сеть и ее рисунок – принципиально разные вещи.

С помощью SCg-кода осуществляется отображение на экране не только пользовательских сообщений, адресуемых системе, и не только сообщений, адресуемых пользователю, но и всей остальной информации, необходимой для организации работы пользователя (прежде всего – это элементы управления интерфейсом). Такая унификация отображаемой пользователю информации дает возможность организовать взаимодействие пользователя с help-системой точно так же, как и его взаимодействие с основной (предметной) системой.

Трактовка элементов управления пользовательским интерфейсом как элементов отображаемого на экране SCg-текста позволяет:

- (1) унифицировать представлений любой информации, отображаемой на экране;
- (2) унифицировать способы инициирования различных вопросов, касающихся любой отображаемой на экране информации (в том числе, и элементов управления);

Для того, чтобы четко отделить те средства SCg-кода, которые обусловлены самим SC-кодом, от тех средств, которые обусловлены стремлением повысить наглядность SCg-текстов, введем ядро SCg-кода (или, просто, SCg-ядро), алфавит которого взаимно однозначно соответствует алфавиту SC-кода и, соответственно этому, тексты которого изоморфны (!) семантически эквивалентным текстам SC-кода.

Переход от SCg-ядра к SCg-коду заключается в ослаблении требований, предъявляемых к изображениям семантических сетей, в целях обеспечения удобства для человеческого восприятия. Такое ослабление осуществляется в следующих направлениях: вводится приписывание идентификаторов изображаемых sc-элементов, расширяется алфавит графических примитивов, допускается уникальное изображение некоторых sc-узлов, допускается синонимия sc.g-элементов, но при этом синонимичным элементам должны быть приписаны одинаковые идентификаторы, вводятся специальные графические средства, направленные на повышение наглядности (шинные линии, контуры).

Заметим также, что кроме SCg-кода для внешнего представления абстрактных унифицированных семантических сетей используются также и другие языки:

- SCs-код, обеспечивающий представление унифицированных абстрактных семантических сетей (sc-текстов) в виде, близком к традиционным текстам;
- SCn-код, обеспечивающий гипертекстовое представление абстрактных sc-текстов, предназначенное для оформления исходных текстов баз знаний.

Более подробно различные языки внешнего представления абстрактных sc-текстов вместе с большим количеством примеров рассмотрены в работах [Голенков и др., 2001].

Принцип 14. Унификация семантических моделей различных пользовательских интерфейсов:

Пользовательский интерфейс интеллектуальной системы, построенной на основе предлагаемой технологии, рассматривать как **специализированную интеллектуальную систему**, построенную по той же технологии и предназначенную для **трансляции адресуемых пользователю сообщений** с внутреннего абстрактного семантического языка представления знаний (SC-кода) на тот или иной внешний язык, тексты которого отображаются пользователю в удобном для него виде, а также для **трансляции пользовательских сообщений** с внешнего языка внутренний семантический язык интеллектуальной системы (т.е. в SC-код).

Трактовка пользовательских интерфейсов как интеллектуальных систем и унификация семантических моделей таких систем дает возможность:

- (1) унифицировать проектирование пользовательских интерфейсов;
- (2) легко наращивать возможности пользовательских интерфейсов;

- (3) легко интегрировать пользовательские интерфейсы с предметными (основными) интеллектуальными системами;
- (4) неограниченно использовать базу знаний предметных интеллектуальных систем для семантического анализа и понимания вводимой пользователем информации (в частности, естественно-языковых текстов).

Более подробно унифицированные семантические модели пользовательских интерфейсов и технология их проектирования рассмотрены в работе [Корончик, 2012].

Принцип 15. Библиотека типовых семантически совместимых компонентов интеллектуальных систем и методика модульного проектирования интеллектуальных систем:

В целях ускорения процесса проектирования интеллектуальных систем создать общую библиотеку многократно используемых семантически совместимых компонентов интеллектуальных систем, на основе которой разработать методику модульного (компонентного, сборочного) проектирования интеллектуальных систем.

В указанной библиотеке можно выделить следующие разделы (частные библиотеки):

- библиотека многократно используемых компонентов баз знаний. Прежде всего, в эту библиотеку входят самые различные по содержанию, но семантически совместимые онтологии. Кроме того, сюда относятся различные "джентльменские наборы" знаний, которыми должны владеть "образованные" интеллектуальные системы. К таким знаниям, в частности, относятся базовые знания по арифметике, базовые знания по теории множеств (каждая интеллектуальная система должна, по крайней мере, отличать элемент заданного множества от его подмножества), базовые знания по теории отношений (каждая интеллектуальная система должна уметь отличать бинарное отношение от многоместного отношения, должна понимать, что такое соответствие), базовые знания по логике (каждая интеллектуальная система должна понимать, что такое теория, высказывание, определение, переменная должна отличать фактографическое высказывание от высказывания, не являющегося фактографическим, должна уметь отличать высказывание от логической формулы, не являющейся высказыванием) и многие другие знания, востребованность которых может быть самой разной;
- библиотека компонентов семантических моделей информационного поиска. Сюда, прежде всего, входят различные информационно-поисковые агенты;

- библиотека компонентов семантических моделей интеграции знаний и машин обработки знаний;
- библиотека интерпретаторов программ, соответствующих различным языкам программирования;
- библиотека различных стратегий решения задач, различных моделей решения задач и агентов, входящих в состав таких моделей;
- библиотека компонентов пользовательских интерфейсов.

Все компоненты, включаемые в состав общей библиотеки компонентов интеллектуальных систем, оформляются как компоненты интеллектуальной собственности (intellectual property), поэтому будем их также называть ip-компонентами.

Особо подчеркнем то, что модульное проектирование интеллектуальных систем возможно только в том случае, если отбор компонентов, включаемых в состав рассмотренной библиотеки, будет осуществляться на основе тщательного анализа качества этих компонентов. Одним из важнейших критериев такого анализа является семантическая совместимость анализируемых компонентов со всеми компонентами, имеющимися в текущей версии библиотеки.

Для обеспечения семантической совместимости таких компонентов интеллектуальных систем, которые являются унифицированными семантическими моделями (sc-моделями знаний, sc-моделями машин обработки знаний, sc-агентов, sc-моделями интеллектуальных подсистем), необходимо (1) согласовать семантику (смысл) всех используемых ключевых узлов и (2) 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.

Принцип 16. Платформенно-независимый характер проектирования интеллектуальных систем:

Для максимальной платформенной независимости технологии обеспечить четкое разделение процесса проектирования формального описания логико-семантической модели разрабатываемой интеллектуальной системы от процесса реализации (интерпретации) этой модели на той или иной платформе.

Подчеркнем при этом следующее. Если каждой интеллектуальной системе соответствует своя уникальная логико-семантическая модель, то каждый интерпретатор абстрактных логико-семантических моделей интеллектуальных систем должен обеспечивать интерпретацию целого класса таких моделей, а в идеале – интерпретацию любой такой модели. Следовательно, разработка указанных

интерпретаторов может осуществляться абсолютно независимо от разработки логико-семантических моделей конкретных интеллектуальных систем.

Таким образом, SC-код, обеспечивающий унификацию семантического представления любых знаний, вместе с языком SCP, обеспечивающим унификацию формального описания агентов, работающих над семантической памятью, являясь средством унификации логико-семантических моделей интеллектуальных систем, выполняют в рамках предлагаемой технологии роль, аналогичную той, которую выполняет язык VHDL в современных микроэлектронных технологиях. В лице SC-кода и языка SCP мы имеем стандарт полного (!) формального описания логико-семантических моделей интеллектуальных систем, обеспечивающий независимость проектирования абстрактных логико-семантических моделей конкретных интеллектуальных систем от разработки различных вариантов реализации (различных вариантов их интерпретации на различных платформах). Такой стандарт является своего рода "водоразделом" между полным платформенно-независимым описанием интеллектуальной системы (абстрактной логико-семантической моделью) и платформенно зависимой реализацией (интерпретацией) этой абстрактной модели.

Полностью построенная абстрактная логико-семантическая модель проектируемой интеллектуальной системы:

- (1) является открытой, поскольку ее можно легко пополнять новыми знаниями и навыками, интегрируя их в текущую версию модели;
- (2) концентрирует внимание на семантические аспекты функционирования интеллектуальной системы и не содержит никаких лишних деталей, обусловленных тем или иным способом ее технической реализации (интерпретации);
- (3) является абстрактным инвариантом целого множества самых различных способов ее технологической реализации (в том числе и с помощью принципиально новых компьютеров).

Разработка прототипа интеллектуальной системы завершается разработкой полной sc-модели этой системы, которая записывается в виде исходного текста с использованием таких языковых средств, как SCg-код, SCs-код, SCn-код. После этого разработчик выбирает один из универсальных (!) вариантов интерпретации (реализации) sc-моделей, загружает разработанные им исходные тексты в выбранный интерпретатор и получает прототип, пригодный для опытной эксплуатации и последующего совершенствования.

Если же после этого разработчиков интеллектуальной системы что-то не устраивает в выбранном варианте интерпретации sc-моделей (в частности, производительность), должна существовать достаточно продуманная методика совершенствования выбранного варианта интерпретатора sc-моделей интеллектуальных

систем. Очевидно, что для каждого варианта интерпретации sc-моделей интеллектуальных систем указанная методика будет иметь свои особенности.

Следовательно, нижние уровни детализации проектируемых интеллектуальных систем, в отличие от верхнего (логико-семантического) являются платформенно-зависимыми. Можно говорить о различных модификациях технологии проектирования интеллектуальных систем, соответствующих разным платформам. Напомним при этом, что основная трудоемкость проектирования интеллектуальных систем, полностью определяющая уровень ее возможностей (уровень знаний и навыков) концентрируется именно на 1-ом этапе проектирования – на разработке ее абстрактной логико-семантической модели.

Таким образом, проектирование интеллектуальной системы можно организовать как два следующих самостоятельных процесса, выполняемых одновременно и независимо друг от друга:

- (1) Процесс разработки абстрактной унифицированной логико-семантической модели проектируемой интеллектуальной системы;
- (2) Процесс совершенствования выбранного интерпретатора абстрактных унифицированных логико-семантических моделей интеллектуальных систем.

Заметим, что сама идея обеспечения кросс-платформенной разработки компьютерных систем путем внедрения формального языка, обеспечивающего описание абстрактных (логических) моделей этих систем не нова. Существует целый ряд кросс-платформенных технологий. Вопрос в том (1) о каком классе разрабатываемых компьютерных систем идет речь, (2) какими свойствами обладают используемые абстрактные модели компьютерных систем, (3) какими достоинствами обладает технология разработки самих этих абстрактных моделей.

Вопросы программной реализации и, в частности, web-ориентированной реализации унифицированных логико-семантических моделей интеллектуальных систем рассмотрены в работе [Колб, 2012].

Принцип 17. Семантический ассоциативный параллельный компьютер:

Обеспечить возможность реализации унифицированных логико-семантических моделей интеллектуальных систем на **семантических ассоциативных параллельных компьютерах**, специально ориентированных на аппаратную реализацию таких моделей.

Очевидно, что для указанных компьютеров базовый графовый язык программирования (язык

SCP) является их ассемблером, т. е. аппаратно интерпретируемым языком программирования.

В связи с проблемой создания компьютеров, ориентированных на обработку знаний, необходимо отметить следующее:

- (1) В таких компьютерах принципиально важна поддержка именно параллельной обработки знаний;
- (2) Опыт использования параллельных компьютеров показывает, что эффективное их использование предполагает разработку качественных параллельных программ требует особой профессиональной подготовки и высокой квалификации. Мир параллельного программирования требует особой культуры, особого стиля мышления. Еще более серьезная профессиональная подготовка требуется для разработки параллельных программ, ориентированных на обработку знаний и использующих ассоциативный доступ к обрабатываемой информации;
- (3) Уровень развития микроэлектронных технологий сейчас позволяет достаточно быстро реализовывать самые смелые компьютерные архитектуры и модели обработки информации. Необходима только четкая постановка задачи;
- (4) Созданию параллельных компьютеров для обработки знаний должно предшествовать создание **технологии** проектирования интеллектуальных систем, в основе которой лежат те модели параллельной обработки знаний, которые будут аппаратно поддерживаться в указанных компьютерах. Иначе мы получим "грудю" талантливо сделанного "железа", эффективность использования которого будет, мягко говоря, весьма низкой. Это главная причина неудач такого рода проектов;
- (5) Предлагаемая технология проектирования интеллектуальных систем как раз и предполагает последовательное выполнение следующих этапов:
 - разработка технологии проектирования абстрактных унифицированных логико-семантических моделей интеллектуальных систем;
 - разработка нескольких вариантов программной реализации абстрактных унифицированных логико-семантических моделей интеллектуальных систем, выполненных на современных компьютерах;
 - разработка, эксплуатация достаточно большого количества прикладных интеллектуальных систем и совершенствование технологии проектирования интеллектуальных систем на основе приобретенного опыта;

- и только после этого разработка семантического ассоциативного компьютера, появление которого не отменит абсолютно ничего, сделанного ранее. Просто появится еще один, но уже аппаратный вариант реализации абстрактных унифицированных логико-семантических моделей интеллектуальных систем, применение которого для уже разработанных абстрактных унифицированных логико-семантических моделей самых различных прикладных систем для конечных пользователей этих интеллектуальных систем абсолютно ничего не изменит, кроме существенного повышения быстродействия.

Рассматривая абстрактную sc-машину обработки знаний на самом верхнем уровне, мы не уточняем (не детализируем) "внутреннее устройство" sc-агентов обработки знаний. Разработав язык SCP, мы получили возможность формально описывать (детализировать) поведение sc-агентов обработки знаний. Если трактовать язык SCP как ассемблер семантического ассоциативного компьютера, то проектирование этого компьютера можно рассматривать как формальный переход к sc-машинам более низкого уровня, обеспечивающим интерпретацию sc-машин более высокого уровня. Существенным здесь является то, что при этом мы не выходим за пределы класса абстрактных sc-машин. Просто вводится последовательность sc-языков программирования все более и более низкого уровня, каждый из которого обеспечивает формальное описание sc-агентов, входящих в состав sc-машины, интерпретирующей программы непосредственно предшествующего ему sc-языка программирования более высокого уровня (см. принцип 9). При этом число таких уровней, т. е. число таких специальных sc-языков программирования (которые можно назвать sc-языками микропрограммирования) должно быть столько, сколько необходимо для доведения формального описания sc-машин до такого уровня детализации, который позволяет перейти от соответствующего абстрактного языка микропрограммирования к формальному описанию цифровой аппаратуры на языке VHDL.

Архитектуру аппаратной реализации семантических моделей обработки знаний можно рассматривать как иерархию абстрактных машин, описывающих переход от агентов, имеющих доступ ко всей семантической памяти, к агентам, имеющим доступ только к своей семантической окрестности и, в конечном счете, взаимодействующим только со своими семантическими соседями.

Аппаратная интерпретация абстрактных sc-машин предполагает создание реконфигурируемой памяти с распределенными в ней процессорными элементами. Такую интеграцию памяти и процессора будем называть **процессор-памятью**. Реконфигурируемость (структурная

перестраиваемость) памяти может быть обеспечена коммутационной средой для процессорных элементов. Можно рассматривать целый ряд подходов к реализации реконфигурируемой семантической ассоциативной процессор-памяти. В частности, процессорным элементам можно ставить в соответствие узлы обрабатываемых унифицированных семантических сетей, а коммутируемым каналам связи между процессорными элементами – коннекторы этой семантической сети. В этом случае текущее состояние конфигурации коммутируемых каналов связи будет полностью соответствовать текущему состоянию конфигурации обрабатываемой семантической сети. Следовательно, память "превращается" из пассивного хранилища байтов в коммутационную среду между процессорными элементами.

Принцип 18. Встроенные подсистемы интеллектуальных систем, обеспечивающие их эффективную эксплуатацию и эволюцию:

Каждую проектируемую интеллектуальную систему трактовать как **результат интеграции следующих интеллектуальных подсистем:**

- предметной (основной) интеллектуальной системы;
- интеллектуального пользовательского интерфейса;
- интеллектуальной подсистемы адаптивного управления диалогом с конечным пользователем;
- интеллектуальной help-системы для информационного обслуживания и обучения конечных пользователей предметной интеллектуальной системы, которые, начиная работать с системой не обязаны иметь сразу высокую квалификацию;
- интеллектуальные системы управления проектированием интеллектуальной системы, которая координирует деятельность разработчиков предметной интеллектуальной системы [Грибова, 2010];
- интеллектуальные системы управления информационной безопасностью предметной интеллектуальной системы.

Подчеркнем, что для обеспечения интегрируемости (семантической совместимости) перечисленных интеллектуальных систем они должны проектироваться на основе одной и той же технологии.

Таким образом, проектируя каждую интеллектуальную систему, необходимо одновременно проектировать:

- также и подсистему, которая осуществляет информационное обслуживание и обучение конечных пользователей данной интеллектуальной системы, т. е. фактически, является оформлением документации по эксплуатации системы в виде интеллектуальной справочной и обучающей системы. Это

существенно расширит контингент конечных пользователей, повысит эффективность эксплуатации системы и существенно упростит эту эксплуатацию;

- также и подсистему, которая обеспечивает координацию разработчиков проектируемой интеллектуальной системы, поскольку разработка (совершенствование) системы продолжается в ходе её эксплуатации и требует создания специальных методов и компьютерных средств постоянного совершенствования предметной интеллектуальной системы непосредственно в ходе её эксплуатации. Это существенно отодвинет срок её морального старения;
- также и подсистему, обеспечивающую управление информационной безопасностью проектируемой интеллектуальной системы.

Если подсистема управления проектированием интеллектуальной системы будет создаваться действительно как интеллектуальная система, интегрируемая с основной (предметной) интеллектуальной системой, то в перспективе она может стать не только координатором деятельности разработчиков, но и самостоятельным субъектом проектирования, способным тестировать, диагностировать, анализировать как основную проектируемую интеллектуальную систему, так и самоё себя.

Принцип 19. Доступность и открытость технологии:

Обеспечить максимально возможное **расширение контингента разработчиков** интеллектуальных систем, использующих предлагаемую технологию, за счет максимальной доступности этой технологии и открытого характера её развития.

Если технология проектирования интеллектуальных систем ориентируется на широкое, массовое распространение и на интенсивное собственное развитие, опирающееся на накапливаемый опыт её использования, она должна быть доступной и открытой. Это означает:

- свободный доступ к всей документации и основанным средствам автоматизации (компьютерной поддержки) проектирования интеллектуальных систем;
- открытость исходных текстов всех основных средств компьютерной поддержки проектирования интеллектуальных систем, всех основных многократно используемых (типовых) компонентов интеллектуальных систем;
- открытость исходных текстов всех "пилотных" проектов прикладных интеллектуальных систем, выполняющих роль "образцово-показательных" проектов;
- открытый характер организации (project-менеджмента) процесса развития технологии, имеющий форму открытого (open source) проекта, участником которого может быть любой

желающий, в том числе, и любой пользователь этой технологии, указывающий на различные ошибки и высказывающий различные пожелания.

Завершая рассмотрение открытого характера предлагаемой технологии, сделаем следующие замечания:

- открытый характер технологии не является препятствием для реализации коммерческих интересов, связанных с этой технологией. Так, например, на коммерческой основе могут создаваться и предоставляться
 - (1) самые различные прикладные интеллектуальные системы,
 - (2) некоторые варианты реализации различных многократно используемых (типовых) компонентов интеллектуальных систем,
 - (3) некоторые варианты реализации интерпретатора абстрактных семантических логико-семантических моделей интеллектуальных систем, в частности, различные варианты построения семантических ассоциативных параллельных компьютеров;
- открытый характер технологии, при грамотном использовании фактора её открытости, способствует обеспечению информационной безопасности, как самой технологии, так и прикладных интеллектуальных систем, созданных на её основе;
- открытый характер предлагаемой технологии проектирования интеллектуальных систем может быть эффективно реализован только на базе **технологии облачных вычислений**, в рамках которой вся предлагаемая технология проектирования интеллектуальных систем рассматривается как некий Internet-сервис [Грибова и др., 2011]

Принцип 20. Эволюционная методика проектирования:

Использовать **методику поэтапного эволюционного проектирования** интеллектуальных систем.

Указанная методика предполагает:

- быстрое проектирование;
- скорейшее введение в эксплуатацию первых версий проектируемой системы с минимальными, но практически полезными возможностями;
- эволюционное поэтапное совершенствование проектируемой интеллектуальной системы путем её расширения новыми знаниями и навыками непосредственно в ходе эксплуатации интеллектуальной системы и активным привлечением её конечных пользователей.

С формальной точки зрения проектирование унифицированной логико-семантической модели (sc-модели) интеллектуальной системы в конечном счете сводится к проектированию sc-модели **базы знаний** этой интеллектуальной системы, поскольку scr-программы, описывающие поведение

sc-агентов, можно рассматривать как часть базы знаний. Таким образом, проектируемая база знаний включает в себя:

- базу знаний предметной (основной) интеллектуальной системы;
- тексты всех scr-программ, описывающих поведение sc-агентов;
- текст документации, представленный в виде базы знаний интеллектуальной help-системы, обеспечивающей всестороннее информационное обслуживание пользователей проектируемой интеллектуальной системы.

Начальный этап проектирования базы знаний интеллектуальной системы – это уточнение иерархической системы предметных областей, которые должны быть описаны в проектируемой базе знаний. Каждой такой предметной области ставится в соответствие определенный раздел проектируемой базы знаний. Среди выделенных разделов проектируемой базы знаний имеются разделы, которые делятся (декомпозируются) на подразделы, а также атомарные (недекомпозируемые) разделы. Далее процесс проектирования всей базы знаний сводится к проектированию каждого её атомарного раздела с последующей их интеграцией в единую базу знаний.

В целом начальную стадию проектирования всей интеллектуальной системы на основе предлагаемой технологии условно разбить на следующие четыре этапа:

- (1) Разработка 1-й версии интеллектуальной системы, которая включает в себя:
 - 1-ю версию её базы знаний;
 - типовое ядро интеллектуальной информационно-поисковой машины, которое входит в состав библиотеки многократно используемых компонентов интеллектуальных систем;
 - типовое ядро интеллектуального решателя, которое входит в состав библиотеки многократно используемых компонентов интеллектуальных систем;
 - типовое ядро пользовательского интерфейса, которое входит в состав библиотеки многократно используемых компонентов интеллектуальных систем.Разработанная 1-я версия интеллектуальной системы уже обладает определенной целостностью, её можно тестировать и запускать в предварительную опытную эксплуатацию.
- (2) Разработка 2-й версии интеллектуальной системы, которая включает в себя:
 - 2-ю версию её базы знаний;
 - 1-ю версию её информационно-поисковой машины;
 - типовое ядро её интеллектуального решателя;
 - типовое ядро её пользовательского интерфейса;
- (3) Разработка 3-й версии интеллектуальной системы, включающей в себя:

- 3-ю версию её базы знаний;
- 2-ю версию её информационно-поисковой машины;
- 1-ю версию её интеллектуального решателя;
- типовое ядро её пользовательского интерфейса;

- (4) Разработка 4-й версии интеллектуальной системы, включающей в себя:
 - 4-ю версию её базы знаний;
 - 3-ю версию её информационно-поисковой машины;
 - 2-ю версию её интеллектуального решателя;
 - 1-ю версию её пользовательского интерфейса.

Дальнейшее развитие проектируемой интеллектуальной системы может акцентировать внимание на самых разных направлениях, приоритетность которых определяется самим приложением.

Более подробно методика эволюционного коллективного проектирования унифицированных семантических моделей интеллектуальных систем, на основе содержательной структуризации знаний (см. принцип 6), описана в работе [Давыденко, 2012]

Принцип 21. Реализация предлагаемой технологии в виде интеллектуальной метасистемы:

Реализовать предлагаемую технологию как **интеллектуальную метасистему ориентированную на поддержку проектирования интеллектуальных систем**, построенную по тем же самым принципам (т. е. по той же технологии), что и интеллектуальные системы, разрабатываемые на её основе.

Указанная интеллектуальная система должна включать в себя:

- теорию (принципы построения) проектируемых интеллектуальных систем, которая входит в состав базы знаний метасистемы;
- библиотеку типовых многократно используемых компонентов (ip-компонентов) интеллектуальных систем, которая входит в состав базы знаний рассматриваемой метасистемы;
- средства автоматизации синтеза, анализа и имитационного моделирования проектируемых интеллектуальных систем и их компонентов (это подсистема интеллектуальной метасистемы, ориентированная на решение задач проектирования интеллектуальных систем);
- интеллектуальную help-систему, являющуюся подсистемой рассматриваемой интеллектуальной метасистемы ориентированной на информационное обслуживание и обучение разработчиков интеллектуальных систем;
- методику проектирования интеллектуальных систем, которая оформляется как часть базы знаний метасистемы;

- методику обучения проектированию интеллектуальной системы, которая также является частью базы знаний метасистемы;
- интеллектуальную подсистему управления проектированием самой метасистемы;
- интеллектуальную подсистему управления информационной безопасностью метасистемы;
- семейство различных вариантов реализации интерпретаторов унифицированных абстрактных логико-семантических моделей интеллектуальных систем.

Учитывая рассматриваемые выше принципы построения предлагаемой нами технологии, она названа Открытой Семантической Технологией проектирования Интеллектуальных Систем (Open Semantic Technology for Intelligent Systems – OSTIS). Можно также её было бы назвать **SC-технологией**, поскольку основой этой технологии является SC-код. Соответственно этому, интеллектуальную метасистему, ориентированную на поддержку проектирования интеллектуальных систем, будем называть **метасистемой OSTIS**.

В интеллектуальной метасистеме OSTIS можно выделить целый ряд подсистем, ориентированных на поддержку проектирования различных компонентов интеллектуальных систем, таких, как:

- базы знаний и различные фрагменты баз знаний (онтологии, формальные теории, программы);
- информационно-поисковые машины, машины интеграции знаний, решатели задач;
- пользовательские интерфейсы (графические, естественно-языковые, мультимодальные).

В интеллектуальной метасистеме OSTIS можно также выделить семейство интеллектуальных подсистем, ориентированных на поддержку проектирования различных классов интеллектуальных систем, таких, как:

- интеллектуальные справочные системы (системы информационного обслуживания);
- интеллектуальные обучающие системы (имеющие подсистемы интеллектуального управления обучением);
- интеллектуальные help-системы для пользователей различных компьютерных систем;
- интеллектуальные системы автоматизированного проектирования;
- интеллектуальные системы управления проектами.

ЗАКЛЮЧЕНИЕ

В предлагаемой технологии OSTIS существенным являются не сами рассмотренные выше принципы, некоторые из которых выглядят очевидными и бесспорными, а весь целостный комплекс этих принципов и их максимально возможная согласованность.

Ключевыми проблемами, решение которых лежит в основе предлагаемой технологии являются:

- обеспечение семантической совместности

(интегрируемости) различных моделей представления и обработки знаний;

- создание общей теории абстрактных семантических моделей интеллектуальных систем, не противопоставляя, а интегрируя самые различные подходы;
- обеспечение максимальной возможной независимости интеллектуальных систем от многообразия вариантов и платформ их технической реализации (в т. ч. и от будущих компьютеров, специально ориентированных на аппаратную поддержку обработки знаний).

Таблица 1 – Стандарты технологии OSTIS

sc-модели интеллектуальных систем:			
<ul style="list-style-type: none"> • sc-модели пользовательских интерфейсов; • sc-модели help-систем; • sc-модели подсистем управления проектами; • sc-модели систем поддержки проектирования; • sc-модели подсистем управления информационной безопасностью 			
sc-модели решателей задач			
sc-модели информационного поиска		sc-модели интеграции знаний	
sc-модели баз знаний			
язык SCK		унифицированные идентификаторы sc-элементов	
SC-язык онтологий	sc-языки целей, вопросов и задач	Логический sc-язык	sc-языки программирования
Язык SCP			
scp-машина			
SC-код	SCg-код	SCs-код	SCn-код
программные интерпретаторы scp-машины		scp-компьютеры	
Internet	Локальные платформы		

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.

[Бениаминов, 1988] Бениаминов, Е.М. Основания категорного подхода к представлению знаний. Категорные средства / Е. М. Бениаминов // Изв. АН СССР. Техн. кибернет. - 1988. - N 2. - С. 21-33.

[Борщев, 1983] Борщев, В.Б. Схемы на клубных системах и вегетативная машина / В. Б. Борщев // Семиотика и информатика. - 1983. - Вып. 22. - с. 3-44.

[Вагин и др., 2008] Вагин, В.Н. Достоверный и правдоподобный вывод в интеллектуальных системах / В. Н. Вагин [и др.]. - М. : ФИЗМАТЛИТ, 2008. - 712 с.

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

[Гастев, 1975] Гастев, Ю.А. Гомоморфизмы и модели. Логико-алгебраические аспекты моделирования / Ю.А. Гастев. - М.: Наука, 1975.

[Гладун, 1994] Гладун, В.П. Процессы формирования новых знаний / В. П. Гладун. – София : Педагог, 1994.

[Голенков и др., 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков [и др.] – Мн. : БГУИР, 2001.

[Грибова и др., 2011] Грибова, В. В. Автоматизация разработки пользовательских интерфейсов с динамическими данными / В. В. Грибова, Н. Н. Черкезишвили // Материалы

- международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2011. – С. 287-292
- [**Гуляева, 1989**] Гуляева, Д.М. Решение прикладных задач на расширенных семантических сетях. / Д.М. Гуляева // Математическое обеспечение ЭВМ и систем программирования. - М., 1989.
- [**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
- [**Гусаков и др., 1981**] Гусаков, В.Я. Динамические алгебраические системы как математическая модель банка данных / В.Я. Гусаков, С.М. Гусакова // Семиотика и информатика. - 1981. - Вып. 17. - С. 43-52.
- [**Давыденко, 2012**] Давыденко, И. Т. Комплексная методика проектирования семантических моделей интеллектуальных справочных систем / И. Т. Давыденко // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012. –С. 457-466
- [**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.
- [**Епифанов, 1984**] Епифанов, М.Е. Индуктивное обобщение в ассоциативных сетях / М. Е. Епифанов // Известия АН СССР. Техническая кибернетика. №5, 1984.- С. 132-146.
- [**Ефимова и др., 1988**] Ефимова, С.М. Поиск в базах знаний, опирающихся на модель П-графов, и его аппаратная реализация на основе метода M^3 / С. М. Ефимова, Е.В. Суворова. – М : Вычислительный центр АН СССР, 1988.
- [**Загорюлько, 1988**] Загорюлько, Ю.А. Технология конструирования средств обработки знаний на основе семантических сетей. Средства спецификации и настройки / Ю. А. Загорюлько. - Новосибирск, 1988.
- [**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, 2012. –С. 297-314.
- [**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
- [**Калиниченко, 1983**] Калиниченко, Л.А. Методы и средства интеграции неоднородных баз данных / Л. А. Калиниченко. - М.: Наука, 1983.
- [**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, 1986**] Клещев, А. С. Семантические порождающие модели. Общая точка зрения на фреймы и продукции в экспертных системах / А. С. Клещев. - Владивосток, 1986.
- [**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. –С. 111-122
- [**Колмогоров, 1958**] Колмогоров, А.Н. К определению алгоритма / А. Н. Колмогоров // Успехи математических наук. - 1958. - Т.13. - № 4(82). - С. 3-28.
- [**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, 2012. –С. 339-346
- [**Котов и др., 1966**] Котов, В.Е. Асинхронные вычислительные процессы над общей памятью / В. Е. Котов, А. С. Нариньяни // Кибернетика. - 1966. - № 3. - С. 64-71.
- [**Кузнецов В.Е., 1989**] Кузнецов, В.Е. Представление в ЭВМ неформальных процедур / В. Е. Кузнецов. - М.: Наука, 1989.
- [**Кузнецов И.П., 1986**] Кузнецов, И.П. Семантические представления / И.П. Кузнецов. – М : Наука, 1986.
- [**Лозовский, 1984**] Лозовский, В.С. Семантические сети / В. С. Лозовский // Представление знаний в человеко-машинных и робототехнических системах. – М. : ВИНТИ, 1984. – С. 84-121.
- [**Любарский, 1980**] Любарский, Ю.Я. Интеллектуальные информационные системы / Ю.Я. Любарский. – М : Наука, 1980.
- [**Люгер, 2003**] Люгер, Дж.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Дж.Ф. Люгер. – М. : Вильямс, 2003.
- [**Мальцев, 1970**] Мальцев, А. И. Алгебраические системы / А. И. Мальцев. – М.: Наука, 1970.
- [**Марковский, 1997**] Марковский, А. В. Анализ структуры знаковых ориентированных графов / А. В. Марковский. // Известия РАН : Теория и системы управления. - 1997. - №5.
- [**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.
- [**Месарович и др., 1978**] Месарович, М. Общая теория систем: математические основы / М. Месарович, Я. Тахакара. – М. : Мир, 1978.
- [**Молокова, 1992**] Молокова, О.С. Методология анализа предметных знаний / О. С. Молокова. // Новости искусственного интеллекта. - 1992. – № 3. - С.11-60.
- [**Нариньяни, 1994**] Нариньяни, А.С. НЕ-факторы и инженерия знаний: от наивной формализации к естественной программатике / 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.
- [**Осипов, 1990**] Осипов, Г.С. Построение моделей предметных областей. Неординарные семантические сети / Г. С. Осипов. // Известия АН СССР. Техническая кибернетика. – 1990. - № 5.
- [**Петров, 1978**] Петров, С.В. Графовые грамматики и автоматы (обзор) / С. В. Петров. // Автоматика и телемеханика. - 1978. - № 7. - С. 116-136.
- [**Петрушкин, 1992**] Петрушкин, В.А. Экспертно-обучающие системы / В. А. Петрушкин. – Киев : Наукова думка. – 1992.
- [**Плесневич, 1982**] Плесневич, Г.С. Представление знаний в ассоциативных сетях / Г. С. Плесневич // Изв. АН СССР. Техн. кибернет. - 1982. – № 5. - с.6-22.
- [**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.
- [**Попков, 1986**] Попков, В.К. Гиперсети и их характеристики связности / В. К. Попков. // Исследования по прикладной теории графов. - Новосибирск: Наука, 1986. - С. 25-58.
- [**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.

[Рабинович, 1995] Рабинович, З.Л. О концепции машинного интеллекта и ее развитии / З. Л. Рабинович // Кибернетика и системный анализ. - 1995. - N2. - С.163-173.

[Рассел, 2006] Рассел, С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. - М.: Вильямс, 2006.

[Резанов, 1989] Резанов, С.Н. Об одном методе обобщения на семантических сетях в системе управления энергообъединением / С. Н. Резанов // Изв. АН СССР. Техн. кибернет. - 1989. - № 5. - С. 55-62.

[Рубашкин, 1989] Рубашкин, В.Ш. Представление и анализ смысла в интеллектуальных информационных системах / В. Ш. Рубашкин. - М.: Наука, 1989.

[Рыбина, 2010] Рыбина, Г.В. Основы построения интеллектуальных систем: учеб. пособ./ Г.В. Рыбина. - М.: Финансы и статистика, 2010.

[Сапатый, 1983] Сапатый, П.С. Об эффективности структурной реализации операций над семантическими сетями / П. С. Сапатый // Техн. кибернет. - 1983. - № 5. - С. 128-134.

[Семенов, 1980] Семенов, В.В. Семантические фреймвые сети как модели предметной области для САПР САУ / В. В. Сапатый // Представление знаний в системах искусственного интеллекта. - М.: МДНТП, 1980. - С. 117-122.

[Скороходько, 1989] Скороходько, Э.Ф. Семантические сети и автоматическая обработка текста. / Э. Ф. Скороходько. - Киев: Наук. думка, 1983.

[Скрэгг, 1983] Скрэгг, Г. Семантические сети как модели памяти / Г. Скрэгг // Новое в зарубежной лингвистике. - Вып. 12. - М.: Радуга, 1983. - С. 228-271.

[Соловьев, 1990] Соловьев, В.А. Формирование на семантической сети понятий и суждений с помощью рассуждений по аналогии / В. А. Соловьев // II Всесоюзная конференция "Искусственный интеллект-90". Секционные и стендовые доклады. - Минск, 1990. - Т. 1. - С. 166-169.

[Тузов, 1984] Тузов, В.А. Математическая модель языка / В. А. Тузов. - Л.: Изд-во ленингр. ун-та, 1984.

[Тузов, 1986] Тузов, В.А. О формализации понятия задачи / В. А. Тузов. - М.: Наука, 1986. - С. 73-83.

[Тыгу, 1989] Тыгу, Э.Х. Интеграция знаний / Э. Х. Тыгу // Изв. АН СССР. Техн. кибернет. - 1989. - № 5. - с. 3-13.

[Финн, 2008] Финн, В.К. Многочленные логики и их применения / ред. В. К. Финн - М.: ЛКИ, 2008. - Т.1, Т.2

[Уварова, 1987] Уварова, Т. Г. Формальное описание операционного языка для семантических сетей. / Т. Г. Уварова, Л. Л. Лифшиц - М.: ВЦ АН СССР, 1987.

[Хельбиг, 1980] Хельбиг, Г. Семантическое представление знаний в вопросно-ответной системе FAS-80 / Г. Хельбиг. // Представление знаний и моделирование процессов понимания. - Новосибирск, 1980. - С. 97-123.

[Хендрикс, 1975] Хендрикс, Г. О расширении применимости семантических сетей введением разбиений / Г. О. Хендрикс // Труды IV Международной объединенной конференции по искусственному интеллекту. - М., 1975. - Т. 1. - С. 190-206.

[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.

[Цаленко, 1989] Цаленко, М.Ш. Моделирование семантики в базах данных. / М. Ш. Цаленко. - М.: Наука, 1989.

[Шенк, 1980] Шенк, Р. Обработка концептуальной информации / Р. Шенк. - Москва: Энергия, 1980.

[Шрейдер, 1971] Шрейдер, Ю.А. Системы и модели / Ю. А. Шрейдер, А. А. Шаров. - М.: Радио и связь, 1982.

[Шуберт, 1979] Шуберт, Л. Усиление выразительной мощности семантических сетей / Л. Шуберт // Кибернетический сборник. Новая серия. - 1979. - Вып. 16. - С. 171-212.

[OSTIS, 2011] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. - 2011. - Режим доступа: <http://ostis.net>. - Дата доступа: 20.11.2011

[Russell et al., 1995] Russell, S. Artificial Intelligence. A Modern Approach / S Russell, P Norvig. - New Jersey : Prentice Hall, - 1995.

[Sowa, 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

[Wooldridge et al., 1994] Wooldridge, M. Agent Theories, Architectures and Languages: A Survey / M. Wooldridge, N. Jennings // Intelligent Agents. Languages. Amsterdam : Springer Verlag, August, - 1994. - P. 3-39.

GRAPHODYNAMICAL MODELS OF PARALLEL KNOWLEDGE PROCESSING

Golenkov V. V., Guliakina N. A.

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

golen@bsuir.by

guliakina@bsuir.by

The principles of construction technology of designing intelligent systems which are oriented on semantic representation of knowledge, expansion of the number of developers and shortening time of design are considered.

INTRODUCTION

Current state in the designing of computer systems reminds Babel of various approaches, models, methods, tools and platforms. Artificial intelligence technologies are not targeted to a wide range of developers of intelligent systems and therefore did not get mass distribution.

MAIN PART

We suggest the following approach, which are aimed to eliminate the above drawbacks:

- orientation on semantic knowledge representation, which completely abstracts from the peculiarities of the technical implementation of intelligent systems;
- development of the semantic-unified and easily integrable computer systems models in the first stage of their design;
- modular designing based on the libraries of the typical reusable components of intelligent systems;
- gradual evolutionary design based on a basis rapid prototyping;
- fully compatible designing tools with the developed system - the tools are built as intelligent systems and are based on the same principles

CONCLUSION

The paper presents the key provisions of the one of the possible approaches to the mass systems development with different levels of intelligence. The results presented in the paper are being tested within the open source project OSTIS (<http://www.ostis.net>) bounds. This work is supported by Belorussian and Russian foundation for basic research.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 007.5:510.66:159.955.5

ОБРАЗНЫЕ ПРЕДСТАВЛЕНИЯ СПЕЦИАЛИСТА И ПРОБЛЕМА ИХ ОТРАЖЕНИЯ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

Кобринский Б.А.

*ФГБУ «Московский НИИ педиатрии и детской хирургии» Минздравсоцразвития России,
ГБОУ ВПО Российский национальный исследовательский медицинский университет
им. Н.И.Пирогова, г. Москва, Российская Федерация*
bakob@pedklin.ru

В работе рассматривается проблема образных представлений специалистов и включения визуальных образов в базы знаний интеллектуальных систем. Предложенное автором понятие образных рядов проанализировано с разных позиций, включая нечеткий образный ряд. Рассмотрены принципы формирования базы знаний на основе фреймоподобных структур, включающих кортежи образов. Представлен фрагмент лингво-образного фрейма в области медицинской диагностики. Результатом должно быть построение понятийно-образной или лингво-образно-логической интеллектуальной системы.

Ключевые слова: визуальные образы, лингво-образные базы знаний, лингво-образно-логическая интеллектуальная система.

Введение

В настоящее время системы искусственного интеллекта опираются на имена как символы. Хотя для человека символами могут являться также и образные представления. Характерным представителем образа-символа служат петроглифы, представляющие собой одномерные знаки, что и обеспечивает мысленную символическую трансформацию таких изображений без существенной потери содержательного смысла. В то же время многоплановые образы – это многосмысловые или многомерные знаки, отличающиеся целостностью, что затрудняет или делает невозможной их прямую трактовку в качестве символов.

Образное мышление основано не столько на анализе отдельных признаков (за редкими исключениями), сколько на неявном учете их связей, ассоциаций с другими признаками. Попытка «сборки» ассоциирующих признаков, неявно учитываемых экспертами и воспринимаемых со стороны как интуитивное принятие решения была сделана при построении медицинской интеллектуальной системы [Кобринский и др., 1995]. Впоследствии это явилось поводом для разработки проблемы отражения образного мышления специалиста в базе знаний интеллектуальной системы.

По-видимому ни у кого не вызывает особых сомнений тот факт, что интуиция, довольно тесно связанная с образным мышлением, играет значительную роль в формировании первичных гипотез в слабо структурированных областях

знания, таких как медицина и гуманитарные науки. Определенному образу или образам, возникающим у человека в различных ситуациях при воспоминании, может соответствовать множество близких или относительно близких (не тождественных) изображений или явлений. Сходные проявления образа, отвечающего одному понятию, можно рассматривать как множество или образный ряд [Кобринский, 1998, 2000], формирующийся на основе реальных объектов или мысленных (псевдовизуальных) образов. Включение в образный ряд может осуществляться на основе различных принципов: (а) по сходству (близкие лица, объекты, ситуации, в том числе отдельные фрагменты), (б) по ассоциации (сходство во времени, в пространстве, на базе метафорического представления). Избыточное число степеней свободы образа по отношению к оригиналу как необходимое условие однозначного восприятия действительности, отражения ее пространственных и предметно-временных форм В.П. Зинченко [1991] относит к парадоксам психологии и характеризует, с одной стороны, как перцептивное действие, а с другой – как образный или визуальный интеллект. Новые экспериментальные данные, полученные группой ученых Бостонского университета, позволили профессору Такео Ватанабэ [Watanabe, 2011] сделать заключение, что первичная зрительная кора взрослых достаточно пластична и позволяет осуществлять визуальное перцептивное обучение.

Зрительные образы отличаются большой ассоциативной мощностью. Фрагмент картины, эпизод кинофильма могут на основе принципа

визуальной аллюзии восстановить в памяти полностью всю картину, а зачастую и вызвать из памяти сходные сюжеты или название, т.е. активизировать как право-, так и левополушарные процессы, что указывает на связь образных воздействий с обеими формами восприятия мира.

Образы и интуиция представляют собой в основном невербализуемые знания. Попытка замены их вербальными понятиями может вести к искажению или разрушению холистического визуального образа. И если в действующих медицинских экспертных системах отдельные признаки – форма черепа, носа – вынужденно подвергаются вербализации, то очень трудно уточнить форму носа: орлиный, клювовидный или бульбообразный, грушевидный, сливовидный. Зачастую между ними крайне сложно провести границу и отнести к одному из названных вариантов. Значительно более надежным было бы представление ряда визуализированных признаков.

Вопрос о включении в интеллектуальные системы образных представлений подкрепляется мнением о том, что как дискретная символическая система языковых представлений, так и аналоговая или функциональная система образных и действенных представлений имеют свою долговременную память и кодируют поступающую информацию, соответственно, в форме символических и образных репрезентаций [Цапкин, 1997]. Это находит свое отражение и в информационной избыточности двойного кодирования, характерной для сверхсложных систем, которые в поисках эффективного поведения, при неполноте информации, стремятся восполнить этот дефицит разнообразием. Такая постановка вопроса позволяет предположить возможность извлечения из памяти эксперта, в особенности с развитым правополушарным мышлением, дополнительной информации. В медицине, к примеру, это касается включения знаний не только в виде сведений о релевантности симптомов болезней, но и в форме диагностически значимых “образов”, которые должны найти свое место в составе гибридных понятийно-образных баз знаний.

1. Интуиция и образное мышление специалиста

Эффект озарения – это восприятие образа “по наитию” (например, предварительного диагноза болезни до подкрепления первичной гипотезы дополнительными фактами) на основе не всегда понятных субъекту характеристик (проявлений). При этом наблюдаемый образ может быть как полноценным (присутствуют все формирующие его признаки), так и неполным, вследствие разной степени выраженности признаков и/или отсутствия части из них, или размытым (нечетким), т.е. совокупность явлений не обязательно в полной мере соответствует “классическому” образу в памяти человека. В построении внутреннего образа важную роль играют механизмы распознавания,

хотя при этом часть входной информации теряется [Кузнецов, 1998]. Можно предполагать, что псевдовизуальный образ при опознании лица или проявлений болезни постепенно делается более четким (по типу фокусировки бинокля). Изменение четкости образа – «центрирование образа» – это своего рода путь от диафоры (x или y), т.е. любого высказывания о различии, к эпифоре, т.е. любому высказыванию о сходстве.

Мышление образами, как первый этап оценки ситуации, позволяет иногда составить относительно полное представление о предмете путем мысленного сравнения с “изначальным” образом, который есть энграмма или “осадок в памяти” [Юнг, 1995], но вне прямой связи с последовательным сканированием признаков в процессе наблюдения. Этому соответствует представление, что слова обрабатываются последовательно, тогда как визуальные образы параллельно, “сразу целиком” [Paivio, 1969], т.е. типичный, например, для определенного заболевания внешний вид больного может восприниматься в виде единого целого, тогда как роль выявляемых субъективных и объективных признаков всегда подвергается последовательному анализу в процессе рассуждения и аргументации. Возникновение визуального образа, возможное на любом шаге рассуждения, позволяет сформулировать или уточнить диагностическую гипотезу в эмерджентном процессе.

2. Праобраз и механизм вспоминания

При встрече со знакомым явлением у человека, как правило, имеет место ассоциация образа с праобразом (архетипом). Мысленный образ может быть: 1) основан на воспоминании об аналогичной ситуации (факте) – обычно яркий, со специфическими особенностями (известен из личного опыта или литературы, искусства), возникает (“всплывает”) перед внутренним взором человека. 2) синтезирован на основе вербальных характеристик в виде обобщенного (целостного) псевдовизуального представления – семантический или, скорее, псевдосемантический метафорический образ (к примеру, птицеголовой карлик).

Праобразы-понятия должны рассматриваться как индикаторы конкретных образных рядов, в составе которых могут быть представлены и гротескные формы из мира образов-метафор [Кобринский, 2009б]. По мнению ряда психологов при анализе изображений человек использует операцию концентрации. При обнаружении композиционного центра, например, картины, формируется фокус внимания. Образ в фокусе становится четче, когда человек может подобрать к нему словесный эквивалент. Но именно в этом и заключается сложность анализа образов. Словесный эквивалент может оказаться слишком грубым или ошибочным. В обоих случаях будем иметь неверное решение. Выходом из положения могло бы быть визуальное сравнение наблюдаемого явления или признака с вариантами, хранящимися в базе знаний.

При этом возможно учесть многообразие вариантов, их цветовые и другие характеристики, зависящие от исходной точки наблюдения и различий в восприятии цвета.

3. Многообразие образных представлений

У людей существуют значительные различия в мысленном представлении литературных персонажей и таких традиционных мифологических понятий как Баба Яга, которые характеризуются разнообразными художественными и театральными отображениями. Словесное (вербализованное) описание героя литературного произведения (Мюнхгаузен, Базаров, Пьер Безухов, Иван Карамазов и т.д.) порождает индивидуальный мысленный псевдовизуальный образ. И этот образ может быть достаточно «жестким». Отсюда иногда возникающее неприятие театрального или кинематографического персонажа литературного произведения ввиду его несоответствия со сложившимся при чтении образом, своего рода первоэштаблом, с которым в последующем происходит сравнение встречающихся в жизни вариантов (представителей) того же образа. В то же время, в изобразительном искусстве мы встречаемся с множественным повторением одних и тех же сюжетов, внешний вид персонажей которых может сильно различаться. И чаще всего мы не отрицаем большинства вариантов. Таким образом, в памяти может зафиксироваться некоторое множество сходных, но различающихся вариантов, формирующих образный ряд, базисом которого служит символ определенного образа. Предъявление или обнаружение нового представителя такого ряда, неизвестного ранее, позволяет более или менее быстро его идентифицировать, объединив с известным множеством, и, одновременно, зафиксировать определенные, иногда существенные, различия. Понятно, что в памяти сохраняются или типичные представители каждой подгруппы (возможно подвергшиеся упрощению), или набор образов, отражающих специфические особенности исходного символа. Также как в искусстве, образный ряд формируется у врача на основании зафиксировавшихся в памяти изображений больных.

Образные представления можно подразделить на семантические (знаковые, восстанавливающие понятие или смысл имени – концепт) и визуальные или псевдовизуальные (мысленные), которые носят также название иконических знаков. Некоторая условность такого деления проявляется при обращении к уже упоминавшимся петроглифам, которые содержат элементы языковой и иконической составляющих. Рассматривая в этом контексте “семантический треугольник” [Pospelovetal., 1997], можно подставить на место ментального образа как собственно визуальный, так и псевдовизуальный образ, соответствующий определенной ситуации, специфическое проявление

определенного явления, соответствующего денотату реального мира.

4. Образный ряд как отражение этнических и индивидуальных представлений

Образный ряд условно бесконечен, так как индивидуум, как уникальный мир, представляет любое явление по-своему, даже если эти варианты близки в определенном смысле. В особенности это касается «зрения» художника. Каждый видит свое в известном явлении и отсюда проистекают различия вариантов в художественном отображении всем известного образа, например, Венеры у художников разных времен и народов. Попробуем проиллюстрировать и обосновать это на примерах сюжетных рядов в искусстве.

Обратимся к маскам разных народностей Африки, которые позволяют наряду с общим для их культуры, увидеть и выделить в этом ряду характерное для конкретных этносов (рис.1). Наряду с общим, у них имеются более или менее существенные различия. Последнее может быть крайне важно для идентификации изображений, относящихся к определенным народностям, ареалам их распространения, изменениям в их мировоззрении. Аналогично и в медицине необходимо учитывать этнические особенности при анализе внешнего вида больных с определенными заболеваниями, в особенности наследственными.

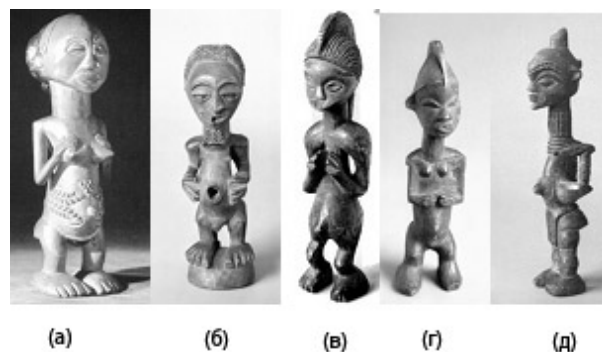


Рисунок1 – Скульптуры народностей Заира:
(а) Балуба, (б)Басонге,(в) Бапенде,
(г) Бахуана,(д) Лулуа

По такому же принципу можно отождествлять работы художников по определенным повторяющимся признакам. Например, специфические природные «фигуры» на коре деревьев в циклах работ «Кора» и «Стволи» художника Петра Фатеева [Пospelov, 2007] указывают не только на особенности художественного восприятия окружающего мира, но могут служить определяющим знаком (символом) для его узнавания. Мистическо-интуитивный мир Петра Фатеева, получивший отражение в книге Д.А. Пospelова [2007], можно проследить в метафорическом образном ряду, где мы встречаем «преображение» пастели «Рок» в

образ дракона войны в одноименной картине, что указывает на возможность формирования ряда на основе преломляющихся один в другой образов, объединенных в данном случае понятием страшной силы, стоящей над человеком. У этого же художника мы видим крайне интересное отображение различных граней видения мира Заратустрой – своего рода динамический философский образный ряд носителя определенной идеи (рис.2).



Рисунок2 – П.П. Фатеев (из цикла «Заратустра»)

При этом образы одного визуального ряда, имеющие те или иные отличия друг от друга, представляют собой как бы условно нечеткий образ типичного представителя ряда.

Таким образом, использование образных рядов может позволить ставить и решать задачи поиска характерных объединяющих особенностей среди образов одного типа как в гуманитарных областях (например, образы гаргулий на соборе Парижской Богородицы – рис.3), так и в естественных науках (например, лицо человека в процессе развития заболевания).



Рисунок3 – Гаргульи (Собор Парижской Богородицы)

5. Типы образных рядов

Задачу более или менее адекватной передачи образа для формирования у человека нужного представления, приводящего к изменению в его когнитивной сфере, В.Б. Тарасов [1998] называет коммуникативно-когнитивной. Это относится как к взаимодействию людей с разнообразными информационными материалами (текстовыми, визуальными), так и к интерактивному взаимодействию с системами искусственного интеллекта. Последние могут включать множества образов, представляющих разнообразные варианты (по форме отображения, по деталям, по временным характеристикам) образных рядов.

5.1. Контурные ряды

Представляется, что в базе знаний в качестве элементов можно было бы использовать как контурные (графические), так и целостные изображения. Штриховые изображения могут

служить для первичного интеллектуального анализа с последующим уточнением в ряде последовательных сопоставлений с образами, совпадающими по определенным оценочным критериям. По принципу введения масок на различных признаковых пространствах, можно подумать о введении маски образа для отображения его метаморфоз в различных ситуациях или на временной шкале возрастных изменений, тем более, что графическое представление в своей основе близко к маскообразному, подчеркивавшему (выделявшему) основную идею или эмоцию. Исторически маска отвечала, по мнению поэта и философа Вячеслава Иванова [1994], культовому понятию вселенского закона превращений и метаморфоз.

Ассоциация с видимым ранее образом, его восстановление может происходить и при предъявлении схематического (контурного) изображения графического типа. Это обусловлено тем, что с помощью механизмов памяти в соответствующих акцепторах результатов действия запечатляются образы, формируются множественные динамические стереотипы действительности. Отпечатки действительности выполняют в этом случае роль информационных голографических экранов, с которыми взаимодействуют внешние объекты [Судаков, 2002]. Видимо в роли таких «экранов» могут быть и схематизированные явления действительности. Ведь именно штрихи использовали первобытные художники для создания петроглифов, контурными линиями Пикассо формирует образ быка, у которого и в наиболее реальном первом изображении художник видит ключевые линии, а на последнем, четвертом, рисунке вообще абстрагируется от «реальности» и представляет лишь «графическую формулу быка» (рис.4).

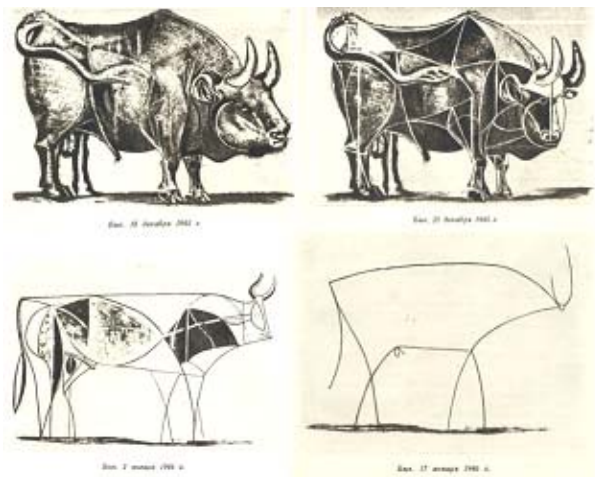


Рисунок4 – Трансформация изображения быка в графике Пикассо

Обращаясь к информационно-голографическому представлению, линии быков Пикассо можно условно сопоставить осколки голограммы, которые чем мельче, тем более

низкого качества изображение (контур быка с минимальным количеством деталей тоже явно хуже по сходству, чем его полноценное изображение на первом и последующих рисунках серии).

Фактически любое упрощенное изображение, как древние статуэтки, наскальная живопись, штриховые изображения, в определенной степени близкие иероглифическому письму, есть образ-символ. Можно сказать, что скелетизация изображения помогает уточнению (или даже выявлению) основных линий в целостном образе, аналогично тому, как тезисы доклада определяют линию доклада, его узловые точки и позволяют увидеть недостающие звенья или проверить последовательность шагов.

Использование принципа упрощения изображения – по типу графической серии быка Пикассо, чему соответствует превращение упрощенного образа или, по Д.А.Поспелову [1998], сведение к эталонам – это своего рода дефрагментация образа, но в пределах, позволяющих осуществить мысленное восстановление первичного перцептивного образа. Подобное, но менее выраженное упрощением можно видеть, например, в графических изображениях женского и мужского лиц при сохранении ключевых деталей, обеспечивающих выявление их гендерных различий (рис.5). К обоим случаям может быть применимо понятие приближения к исходному образу на основе постепенного повышения детализации визуального представления.



Рисунок5 – Графика женского и мужского лиц

Фактически при переходе к графическим изображениям происходит выделение существенных деталей и исключение малозначимых и случайных характеристик образа. Этот аспект, в сравнении логического и образного понятий, подробно рассмотрен в работе Ю.Р. Валькмана [2009], где было отмечено, что образ-понятие может быть неполным, так как различные признаки имеют в нем разную значимость. В образе признаки трактуются весьма широко – это могут быть различные характеристики, атрибуты, свойства, параметры образной целостности. Из этого свойства образа вытекает базовое различие между понятием (в аристотелевской трактовке чаще термином) и понятием-образом. Если признаки, входящие в определение логического понятия, формально равноправны – отсутствие любого из них означает,

что данный объект не удовлетворяет этому определению (за исключением случаев, когда одному явлению соответствуют несколько равнозначных логических понятий), – то в образе есть признаки более и менее существенные.

Наложение образов и их последующее сопоставление может осуществляться с использованием пропорций и меток или точек-маркеров на изображении, т.е. своего рода параметрическое сопоставление (например, на рис.6 с изображением силуэта головы человека у Леонардо да Винчи). Однако такой подход не всегда позволяет решить поставленную задачу, так как, к примеру, в медицинской диагностике большое значение может иметь форма ушей, приращение мочки уха, толщина губ, многочисленные варианты формы носа и многое другое. Это же относится и к другим предметным областям.

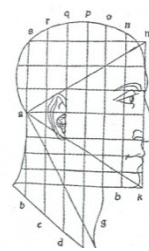


Рисунок6 – Схематизированное изображение головы человека

При когнитивном моделировании субъект представляет наблюдаемую ситуацию в понятийной системе поля знаний в виде иерархии «часть – целое» [Кулинич, 2002]. Используя содержание обобщенного понятия в качестве критерия поиска в собственной памяти (с привлечением воображения и интуиции) или в доступных хранилищах информации, делается попытка определить элементы объема обобщенного понятия. Структуризация семантического пространства осуществляется в форме понятийного кластера. Визуальные характеристики также могут быть сопоставлены такой модели по принципу «фрагменты и целое» с формированием структуры образного кластера.

5.2.Временной образный ряд

Как пишет Бертран Рассел [1999], «множества явлений различаются время от времени». Мысленный процесс узнавания (за исключением мгновенно происходящего неосознаваемого симультанного узнавания) представляет собой сравнение наблюдаемого с «эталонами» в памяти, но при учете возможного отклонения в деталях, без чего было бы невозможно узнавать людей через много лет или опознавать сходные предметы, картины, явления, обнаруживать параллели, выявляющие гротескные особенности личности.

На продолжительном историческом временном интервале у разных народов можно наблюдать

вариации в представлении, например, образа сфинкса (рис.7), изображения которого хотя и претерпевают изменения, но сохраняют ключевые характеристики. Выстраивание такого образного ряда позволяет, с одной стороны, проследить динамику изменений, с другой стороны, география мест создания скульптур позволяет получить представление об этнографических особенностях отдельных образов. Представленные в таком виде изображения позволяют выйти на искомый понятийный образ даже при весьма смутном (отдаленном) представлении об изображаемом предмете.



Рисунок7 – Сфинксы различных времен и народов

Множество вариантов («двойников») одного образа можно сравнить с серией последовательных фотографий одного лица или явления, отличающихся в той или иной степени в зависимости от временного интервала, или с изображениями одного образа на картинах разных художников. Это касается в первую очередь лиц, меняющихся в разных эмоциональных состояниях, являя собой метаморфозы конкретного человеческого образа. Естественно предполагать, что временной (динамический) образный ряд характеризуется трансформацией или метаморфозами образов при сохранении общего и характерного.

Динамика во времени, характеризующая изменения внешнего вида, позволяет говорить о динамическом образном ряде, демонстрирующем последовательность трансформации. Вербализация нечетких переходов разрушает единство образа. А визуальный ряд демонстрирует их единство. Образные динамические ряды имеют место в искусстве, археологии, этнографии, медицине. Если обратиться к медицинской диагностике, то можно представить последовательности образов, характеризующих болезнь в ее динамике, что формирует в памяти врача серии образных рядов, которые желательно отражать в базе знаний, если мы ставим себе целью осуществлять распознавание заболевания на разных этапах его развития.

5.3. Нечеткие образные ряды

Образ обычно более или менее четко фокусируется мысленным зрением, но именно его «ядерная» составляющая, тогда как периферия образа выглядит расплывчатой или в форме неясных теней, которые могут являться как составной частью образа, так и быть примыкающими к нему, т.е. сопутствующими (в том числе случайными) проявлениями [Кобринский, 1998]. Это относится как к мысленным образам, в особенности далеко отстоящим по времени событиям или встречам, так и

представленным схематически (по типу рассмотренных выше контурных изображений) и, конечно, к образному ряду, включающему близкие, но не идентичные изображения.

Рассматривая понятие образного ряда нужно понимать его как образный континуум или, точнее, квазиконтинуум близких или относительно близких изображений, что предполагает нечеткость переходов между отдельными представителями рассматриваемой последовательности [Кобринский, 2009a]. В медицине элементы – члены кортежа – образного ряда демонстрируют различные варианты сходства и различия в клинической картине заболевания, наблюдавшиеся специалистами различных клиник, в том числе в разных странах, т.е. у людей различающихся этносов и в различные возрастные периоды, соответствующие разным фазам болезни.

Врач, способный на интуитивное решение, как правило, в довольно высокой степени уверен в своем предположении, базирующемся на его знаниях о роли ограниченного числа симптомов, обнаруженных при начале осмотра. То же касается и ситуации с возникновением у него аналогичного образа при условии развитого образного мышления (хорошей памяти «на лица»). Исходя из этого, уверенность эксперта должна включать оценку по шкале нечетких представлений, начинающейся с однозначно уверенной оценки:

- “абсолютно достоверно” (полная определенность),
- “скорее всего” (крайне высокая степень определенности),
- “весьма вероятно” (высокая степень определенности),
- “противоречивые или спорные сведения”, что относится, как правило, к комиссионному решению при проведении консилиума (неопределенность),
- “мало вероятно” (высокая степень отрицательной определенности),
- “сомнительно, но не исключено” или “крайне мало вероятно”, т.е. почти полное отрицание (крайне высокая степень отрицательной определенности),
- “абсолютно невозможно”, т.е. полное отрицание (полная отрицательная определенность).

Предложенная лингвистическая шкала, в отличие от линейных порядковых шкал, задает не определенный порядок сущностей, а размытый, так как имеет место пересечение интервалов числовой шкалы [Поспелов, 1997]. Так достигается упорядочение нечетких образов в ряду на основе оценки близости по отношению к некоторому «идеальному» объекту которому можно сопоставить архетипический образ [Кобринский, 2008]. Другим вариантом упорядочения является последовательное размещение образов соответственно возрасту анализируемых субъектов (больных в медицинских исследованиях, тем более, что ряд заболеваний характеризуется прогрессирующим течением, приводящим к

изменению фенотипа, т.е. внешнего вида пациентов). Упорядочение нечетких образов в ряду должно осуществляться на основе некоторой лингвистической характеристики, сопровождающейся фактором уверенности специалиста.

Нечеткие множества можно рассматривать как способ установления упорядочения на множестве возможных миров или на возможных интерпретациях. Такое применение приводит к двум нечетким логикам с различной семантикой: логике сходства или близости [Ruspini, 1973] и логике предпочтения или возможностей [Duboisetal., 2005]. Первая занимается различными метриками для возможных миров, вторая – различием более или менее правдоподобных интерпретаций. Соответственно, логике сходства можно поставить в соответствие истинные образы в ряду, тогда как сравнение истинных и метафорических образов, архетипически послуживших причиной наименования конкретного образного ряда, может осуществляться с использованием обеих логик или логики предпочтения [Кобринский, 2008].

Интуитивные и образные представления эксперта, включающие его уверенность в своих предположениях, должны быть сформулированы в базе знаний. Различная степень неопределенности, в зависимости от характера возникающего образа болезни, может быть отражена с помощью категорий нечеткой логики.

Рассматривая различаемые сущности как денотаты, а множество объектов восприятия как денотативное пространство [Нариньяни, 2003], можно говорить о денотатах образного ряда в базе знаний и об отражении «подлинников» в системе восприятия пользователя. Но в нашем случае компоненте реальности в когнитивном процессе сопоставлен ряд денотатов – денотаты «подлинника» в образном нечетком ряду и денотат отражения в когнитивном процессе. Денотаты в ряду можно поставить в соответствие метаденотатам, являющимся производными метапонятий – элементов когнитивного пространства. С каждым метаденотатом соотносится класс (нозологическая группа) или метаобразный ряд, т.е. открытое множество наблюдавшихся, наблюдаемых и/или потенциально возможных (т.е. виртуальных) денотатов, включаемых в данный класс по принципу сходства визуальных проявлений группы заболеваний [Кобринский, 2009а].

На вход системы могут поступать нечеткие (в смысле сходства с архетипом) образы (например, лица разных больных), а в базе знаний храниться серии фрагментарных и целостных изображений аналогичных образов, представляющих собой прецеденты, имевшие место в прошлом, что должно позволять принимать решение о сходстве при определенном заболевании.

Для выявления сходства членов образного ряда может быть использована методика субъективного шкалирования, но видимо в квантовано-

континуальной форме, в условиях попарной оценки экспертом степени сходства визуальных образов по категориальной шкале (рис.8). При таком подходе объективное знание, будучи представлено в форме попарных мер близости между ситуациями, составляет так называемое «семантическое пространство» предметной области [Воинов, Кобринский, 2004].



Рисунок 8 – Методика субъективного шкалирования

6. Образно-лингвистическая база знаний

Удобной математической моделью для представления многопризнаковых объектов является мультимножество или множество с повторяющимися элементами. Метод упорядочения объектов основан на оценке их близости по отношению к некоторому «идеальному» объекту в многопризнаковом пространстве. Совокупность многомерных объектов может иметь в пространстве сложную структуру, достаточно трудную для анализа. Представления многопризнаковых объектов, основанные на формализме мультимножеств [Петровский, 2002], позволяют одновременно учесть все комбинации значений количественных и качественных (непрерывных и дискретных) признаков. Продолжая этот ряд определений, можно сказать, что признаки могут носить также и квазиконтинуальный визуальный характер. В этом случае мы будем иметь визуальное множество квазиконтинуального типа, в отдельных случаях включающее дискретизируемые вторичные или дополнительные признаки.

Включение в базы знаний, наряду с логико-лингвистическими понятиями, визуальных образов (рисунков, картин, фотографий, в том числе, представленных в элементарной форме), рассматриваемых как своего рода символы, может быть тем путем, который позволит отражать невербализуемые ментальные представления и использовать их в системе рассуждений, логико-аргументационных построений и принятии решений

путем обработки непосредственно «сенсорных» образов. По мнению Д.А. Поспелова [1998], система должна порождать образы из образов, т.е. должна быть реализована система операций над образами. В качестве элементов в такой базе знаний можно было бы использовать как контурные, так и целостные изображения. Контурные изображения могут послужить для «первичного» интеллектуального анализа (ориентировки) на минимальный уровень сходства с последующим уточнением в ряде последовательных сопоставлений с образами ряда, совпадающими по определенным критериям (точкам-маркерам или памятным знакам). В определенной степени на этом принципе построена работа фоторобота, где образ собирается из отдельных деталей. В нашем случае, при использовании графических изображений используется как бы принцип дефрагментации образов с выделением существенных для распознавания фрагментов (деталей).

Условие выдвижения гипотезы об «образе» в базе знаний интеллектуальной системы должно осуществляться в интерактивном режиме и сопровождаться указанием эксперта о степени его уверенности в отношении наблюдаемого образа [Кобринский, 1998]. В этом плане можно вспомнить работы по созданию аппарата алгебры образов для работы с нечеткими когнитивными изображениями [Ковалев, 1991].

Речь должна идти, так или иначе, о включении визуального ряда в качестве второй составляющей, наряду с лингвистической компонентой, в традиционные семантические базы знаний, которые в этом случае будут базироваться на двух символических представлениях – лингвистических и иконических, для формализации которых могут быть использованы фреймовые структуры [Кобринский, 2008]. Можно сказать, что вторая составляющая должна быть ориентирована на «технологии визуализации в рассуждениях, основанных на прецедентах» [Осипов, 2001], а также аналогах. При этом использование библиотек визуальных нечетких образных рядов в процессе рассуждения по аналогии (понимая под аналогами в данном случае и образы-прецеденты) может опираться на принципы, положенные в основу СВР-технологии.

Рассмотрим возможность использования фреймового подхода для формализации понятийно-образной базы знаний. Визуальные образные ряды можно было бы строить по типу фреймоподобной структуры, где фрейму соответствует традиционное представление определенного образа (или типичный представитель образного ряда, своего рода архетип или родительский фрейм), а слоты представлены образами данного типа («индивидуумами» или элементами образного кортежа), отличающимися по отдельным невербализуемым или трудно вербализуемым характеристикам (к примеру, особенностям внешних проявлений болезни у разных пациентов и в разные периоды заболевания). В качестве элементов образного кортежа могут быть включены как

целостные, так и фрагментарные изображения, позволяющие представить (подчеркнуть) специфические особенности, присущие конкретному образному ряду. При таком подходе в слотах можно отображать варианты образов, их цветовые и некоторые другие характеристики. Контекст во фреймах может пояснять (сопровождать) визуальные образы, но и изображения ряда могут быть контекстом по отношению к семантическим представлениям. В этом случае, при вводе определенных признаков, "демон" должен будет приводить соответствующий фрейм (фреймы) с изображениями (образами) в активное состояние. И соответствующие образные ряды, содержащие кортежи близких по проявлениям изображений, должны будут демонстрироваться пользователю на определенном этапе работы интеллектуальной системы с учетом предполагаемых диагнозов, входящих в дифференциально-диагностический ряд, для визуального определения сходства с исследуемым образцом. В естественном (человеческом) интеллекте срабатывает механизм ассоциации с встречавшимся ранее визуальным образом. В системе на основе искусственного интеллекта речь должна идти о включении образных рядов, с точки зрения оценки их полезности экспертом / пользователем, в процесс разрешения конфликтов при активизации более одного правила в сети фреймов. В конечном счете построение логико-лингво-образных систем может быть основано на сочетании системы аргументов с обработкой различных составляющих нечеткой образной компоненты или целостного визуального образа.

Таким образом, сравнение «нового» образа с известным, предполагающее совпадение неких ключевых понятий с одним или несколькими аналогами или прецедентом в образном ряду, должно обеспечивать классификацию и идентификацию неизвестного образа. Еще один аспект – воспроизведение из памяти зрительных образов по ассоциации при предъявлении видеорядов, хранящихся в базе знаний. Когда при распознавании образов человеческому мозгу или компьютеру на первом шаге могут быть предъявлены какие-то черты лица, например, нос или глаза, то этим «вызывается» соответствующий параметр порядка, который вступает в конкуренцию со всеми остальными параметрами порядка. «Победив» их, этот параметр порядка восполняет заданные черты другими и восстанавливает целиком все изображение [Хакен, 2001]. В качестве примера образа по ассоциации в живописи можно назвать амура, представляющего собой стимул для всплывания образного ряда Венеры. Такие образы, используемые в качестве медиаторов, могут выполнять, функцию эффективного кода, который облегчает запоминание ассоциативных пар [Солсо, 1996].

В рядах из области искусства в качестве образа-архетипа, представляющегося наиболее характерным, могут быть как маски (гнева,

отчаяния и др.), так и стилизация или графика как элементы выделения главного в образе, придании «выпуклости» характерным чертам. В области скульптуры и археологии родительским фреймом может служить, например, наиболее типичный представитель образного ряда сфинксов, – пожалуй, самый известный Большой сфинкс из Гизы (Египет). Образные ряды в изобразительном искусстве могут акцентироваться на одной главной фигуре. В естественных науках примером может служить фрейм из области медицины, где для конкретного заболевания архетипом может являться носитель наиболее характерных его проявлений – классический вариант (фрагмент фрейма с решателем на рис.9). В образном ряду, сопровождающем контекст (в данном примере умственная отсталость как один из признаков понятийного симптомокомплекса), можно видеть три различающихся фото больных с мукополисахаридозом первого типа (синдром Хурлер), являющихся представителями более длинного кортежа нечеткого визуального ряда. В случае обнаружения сходства диагностируемого пациента с одним из членов представленного образного ряда (в практике будет несколько близких образных рядов), вопрос о диагнозе будет решен, при отличии – продолжение поиска по другим фреймам.

Фрейм СИНДРОМ ХУРЛЕР

И {умственная отсталость}



ЕСЛИ



ТО > подтверждение



Рисунок 9 –Лингво-образный фрейм (фрагмент)

Реализация интеллектуальной системы на фреймах с решателем продукционного типа представляется в настоящее время наиболее перспективной.

Заключение

Интуитивно-образные представления специалистов предметной области нередко значительно ускоряют принятие решений в практике

естественного интеллекта, дополняя систему рассуждений, построенных на логико-аргументационной основе, мысленными псевдовизуальными представлениями. Исходя из этого, принципиально важной особенностью будущих интеллектуальных систем должна быть возможность отражения визуальных образов в качестве одной из составляющих описываемого «предмета» или явления.

Реализация предлагаемого подхода предполагает переход к созданию баз знаний на основе символов двух типов – лингвистических и иконических, последние из которых должны опираться на концепцию образных рядов [Кобринский, 2000]. В естественных науках особый интерес данный подход представляет для медицинской диагностики, где характерные внешние проявления могут служить маркерами конкретного заболевания.

Включение в состав медицинской базы знаний элементов визуализации в виде образных рядов может быть с большим интересом воспринято врачами. К примеру, крайне трудно словами описать грубые черты лица или череп в форме трилистника, которые неотделимы от денотата, однако совсем несложно продемонстрировать это в визуальной форме. В то же время, визуальное представление выходных параметров потребует “поднастройки” экспертной системы на конкретного пользователя, учитывая его индивидуально-опытные представления, характерологические особенности личности и способность к формированию образных представлений или аналитико-синтетической деятельности мозга в зависимости от преобладания деятельности правого или левого полушария мозга [Кобринский, 1996].

В развитии данного представления лингвистическая и образная компоненты (образы-прецеденты или образы-аналоги), как две взаимодополняющие разнотипные знаковые системы (понятие как символ и образ как символ), должны найти формальное выражение в базах знаний. Их представление во фреймовой или фреймоподобной форме может явиться одним из вариантов решения для логико-лингво-иконических интеллектуальных систем.

Методология фреймоподобных структур образных рядов в системах искусственного интеллекта может найти применения в медицине, этнографии, археологии, искусствоведении.

Библиографический список

[Валькман, 2008] Валькман Ю.Р. О моделировании образного мышления: отношения «образы – понятия» / Ю.Р. Валькман // Одиннадцатая национальная конференция по искусственному интеллекту с международным участием: Тр. конф. Т.1. – М.: Ленанд, 2008. – С.369-377.

[Воинов и др., 2004] Воинов А.В., Кобринский Б.А. Иерархия локально – непротиворечивых полей знаний как модель образного мышления и интуиции эксперта в мягких предметных областях / А.В. Воинов, Б.А. Кобринский // Девятая национальная конференция по искусственному интеллекту с международным участием: Тр. конф. – М.: Физматлит, 2004. – С.785– 793.

[Зинченко, 1991] Зинченко В.П. Искусственный интеллект и парадоксы психологии / В.П. Зинченко // Будущее искусственного интеллекта. – М.: Наука, 1991. – С.185-192.

[Иванов, 1994] Иванов В. Споры / В. Иванов // Родное и вселенское. – М.: Республика, 1994. – С.73-90.

[Кобринский и др., 1995] Кобринский Б.А., Фельдман А.Е. Анализ и учет ассоциативных знаний в медицинских экспертных системах / Б.А. Кобринский, А.Е. Фельдман // Новости искусственного интеллекта. – 1995. - №3. – С.90-96.

[Кобринский, 1996] Кобринский Б.А. К вопросу учета образного мышления и интуиции в экспертных медицинских системах / Б.А. Кобринский // V Национальная конференция с международным участием "Искусственный интеллект-96": Сб. науч. тр. Т.2. – 1996. – С.110-117.

[Кобринский, 1998] Кобринский Б.А. К вопросу о формальном отражении образного мышления и интуиции специалиста слабо структурированной предметной области / Б.А. Кобринский // Новости искусственного интеллекта. – 1998. - №3. – С.64-76.

[Кобринский, 2000] Символьно-образный подход в искусственном интеллекте / Б.А. Кобринский // КИИ'2000: Седьмая национальная конференция по искусственному интеллекту с международным участием: Тр. конф. Т.2. – М.: Изд-во физико-математ. лит., 2000. – С.601-608.

[Кобринский, 2008] Кобринский Б.А. Образные ряды и их отображение в базе знаний / Б.А. Кобринский // Одиннадцатая национальная конференция по искусственному интеллекту с международным участием: Тр. конф. Т.1. – М.: Ленанд, 2008. – С.393-400.

[Кобринский, 2009а] Кобринский Б.А. Нечеткий образный ряд в клинической медицине / Б.А. Кобринский // Интегрированные модели и мягкие вычисления в искусственном интеллекте: Сб. науч. тр. V-й Международной научно-практ. конф. Т.1. – М.: Физматлит, 2009. – С.121-127.

[Кобринский, 2009б] Кобринский Б.А. Образный ряд в интеллектуальной системе / Б.А. Кобринский // Искусственный интеллект и принятие решений. – 2009б. – №2. – С.25-33.

[Ковалев, 1991] Ковалев И.П. Алгебра модификации образов на основе теории ультрачетких множеств / И.П. Ковалев // I Всесоюз. конф. "Распознавание образов и анализ изображений: Новые информационные технологии (РОАН)": Тез.докл. Ч.1. – Минск, 1991. – С.51-55.

[Кузнецов, 1998] Кузнецов О.П. Образное мышление и быстрые процессы // Новости искусственного интеллекта. – 1998. - №2. – С.117-130.

[Кулинич, 2002] Кулинич А.А. Модель активизации мышления субъекта в системах когнитивного моделирования / А.А. Кулинич // Восьмая национальная конференция по искусственному интеллекту с международным участием: Тр. конф. Т.2. – М.: Физматлит, 2002. – С.575-584.

[Нариньяни, 2003] Нариньяни А.С. Не-факторы: Неоднозначность (доформальное исследование) (1-я часть) / А.С. Нариньяни // Новости искусственного интеллекта. – 2003. – №5. – С.47-55.

[Осипов, 2001] Осипов Г.С. Искусственный интеллект: состояние исследований и несколько слов о будущем / Г.С. Осипов // Новости искусственного интеллекта. – 2001. – №1. – С.3-13.

[Петровский, 2002] Петровский А.Б. Основные понятия теории мультимножеств / А.Б. Петровский. М.: Едиториал УРСС, 2002.

[Поспелов, 1997] Поспелов Д.А. Знания и шкалы в модели мира / Д.А. Поспелов // Модели мира. – М.: Росс.ассоциация искусственного интеллекта, 1997. – С.69-84.

[Поспелов, 1998] Поспелов Д.А. Метафора, образ и символ в познании мира / Д.А. Поспелов // Новости искусственного интеллекта. – 1998. - №1. – С.94-114.

[Поспелов, 2007] Поспелов Д.А. Амаравелла: мистическая живопись Петра Фатеева / Д.А. Поспелов. – М.: Фантом Пресс, 2007.

[Рассел, 1999] Рассел Б. Философия логического атомизма / Б. Рассел. – Томск: Водолей, 1999.

[Солсо, 1996] Солсо Р.Л. Когнитивная психология / Р.Л. Солсо. – М.: Тривола, 1996.

[Судаков, 2002] Судаков К.В. Динамические стереотипы или информационные отпечатки действительности / К.В. Судаков. – М.: ПЕРСЭ, 2002.

[Тарасов, 1998] Тарасов В.Б. Панельная дискуссия / В.Б. Тарасов // Новости искусственного интеллекта. – 1998. – №1. – С.115-136.

[Хакен, 2001] Хакен Г. Принципы работы головного мозга: синергетический подход к активности мозга, поведению и когнитивной деятельности / Г. Хакен. – М.: ПЕРСЭ, 2001.

[Цапкин 1997] Цапкин В.Н. Семиотический подход к проблеме бессознательного / В.Н. Цапкин // Бессознательное: Сборник статей. Т.1. – Новочеркасск: Сагуна, 1994. – С. 81-90.

[Юнг, 1995] Юнг К. Психологические типы / К. Юнг. – СПб-М.: Ювента, Прогресс-Универс, 1995.

[Dubois et al., 2005] Dubois D., Prade H. Fuzzy elements in a fuzzy set / D. Dubois, H. Prade // Fuzzy Logic, Soft Computing and Computational Intelligence: Eleventh International Fuzzy Systems Association World Congress. Vol.1 / Y. Liu, G. Chen, M. Ying, eds.– Beijing: Tsinghua University Press/Springer, 2005.– P.55-60.

[Paivio, 1969] Paivio A. Mental imagery in associative learning and memory / A. Paivio // Psychological Review. – 1969. – Vol.76. – P.241-263.

[Pospelov et al., 1997] Pospelov D.A., Osipov G.S. Knowledge in semiotic models / D.A. Pospelov, G.S. Osipov // Seventh Intern. conf. Artif. Intell. and Information-Control systems of robots: Second workshop on applied semiotics. – Smolenice Castle, Slovakia, 1997. – P.1-12.

[Ruspini, 1973] Ruspini E.H. New experimental results in fuzzy clustering / E.H. Ruspini // Inform. Sciences. – 1973. – Vol.6. – No.3. – P. 273-284.

[Watanabe, 2011] T. Watanabe / <http://news.mail.ru/society/7621241>

IMAGE REPRESENTATIONS OF THE EXPERT AND PROBLEM THEIR REFLECTIONS IN INTELLECTUAL SYSTEMS

Kobrinский B.A.

*Moscow Research Institute for Pediatrics and Children's Surgery
Russian National Pirogov Research Medical University
Moscow, Russia
bakob@pedklin.ru*

In work the problem of image representations of experts and inclusion of visual images in knowledge bases of intellectual systems is considered. The concept of imageseries offered by the author is analysed from different positions, including a fuzzy imageseries. Principles of formation of the knowledge base on a basis the structures similar to frames, including cortege of images are considered. The frame is represented the typical representative of aimageseries to whom there will correspond "archetype" or a pattern-prototype or the parental frame. The individual representatives of animageseries placed in slots of the frame, are accompanied by a context of verbal concepts. The fragment of the lingvo-imagined frame in the field of medical diagnostics is presented. Construction of conceptually-imagined or lingvo-image-bearing-logic intellectual system should be result. Systems of artificial intelligence with technology use frame-like imageseries can find application in medicine, ethnographies, archeology, art criticism.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

МОДЕЛЬ КЛАССОВ СЕМАНТИЧЕСКИХ СЕТЕЙ И ИХ ПРЕОБРАЗОВАНИЙ

Тимченко В.А.

*Учреждение Российской академии наук Институт автоматизации и процессов управления
Дальневосточного отделения РАН,
г. Владивосток, Россия*

rakot2k@mail.ru

В работе представлена модель для описания классов семантических сетей, а также модель для описания спецификаций преобразования семантических сетей – структурных проекций в терминах описания их классов, являющиеся инвариантными по отношению к технологическим пространствам. Модель классов семантических сетей включает в себя формализм для описания связи класса семантических сетей с элементами конкретного синтаксиса языка их текстового представления.

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

ВВЕДЕНИЕ

Вопросы преобразования разных видов информации (данных, знаний, программ на формальных языках), обрабатываемых программными системами, остаются сложными и, вместе с тем, одними из первостепенных для исследования. С такими задачами приходится сталкиваться специалистам во многих сферах профессиональной деятельности. К таким областям можно отнести, например, инженерную программную обеспечения, инженерную онтологию, разработку систем для анализа и преобразования программ, моделей и т. д.

Семантические сети успешно используются для представления различного вида данных, знаний, онтологий в разных предметных областях. Поэтому многие содержательные задачи преобразования информации, возникающие в различных технологических пространствах [Kurtev, 2002] [Bezivin, 2005], а также на их стыке, можно сформулировать в терминах преобразования семантических сетей. 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

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]. Поэтому если задача преобразования возникает на пересечении разных технологических пространств и данный класс задач не охвачен ни одной программной системой, то под этот случай необходимо разрабатывать новое специализированное средство преобразования, либо адаптировать одно из существующих.

В частности существующие модели описания метainформации (фактически классов семантических сетей) предоставляют уровень абстракции и нотацию, общепринятую в том технологическом пространстве, на которое ориентирована данная модель (например, язык MOF и его разновидности в технологическом пространстве modelware, РБНФ и другие формализмы спецификации грамматик в технологическом пространстве grammarware) [Wimmer, 2007]. Поэтому описание в терминах таких моделей метainформации из другого технологического пространства либо невозможно, либо сопряжено с большими трудностями.

В данной работе представлены модели классов семантических сетей и их преобразований, являющиеся инвариантными по отношению к технологическим пространствам. Данные модели позволяют определять как структурные

преобразования, так и преобразования из текстового представления информации в структурное (в виде семантической сети) и наоборот.

1. Основные понятия

Введем ряд определений терминов, которые будут в дальнейшем использоваться в работе.

Под *описанием класса семантических сетей информации* (S) будем понимать описание устройства информации из некоторой предметной области – ее структуры, свойств и ограничений на возможное содержание в виде семантической сети. Описания классов семантических сетей формируются в соответствии с *Моделью описания классов семантических сетей*.

Под *представлением информации в виде семантической сети* или просто *семантической сетью* (S_{inst}) будем понимать семантическую сеть, являющуюся экземпляром (информацией более низкого уровня общности) по отношению к сети S , представляющей собой информацию более высокого уровня общности – метаинформацию. Это означает, что вершины S_{inst} , представляющие понятия, входящие в объемы понятий, представленных вершинами из S , являются экземплярами этих вершин из семантической сети S , тогда как сами вершины семантической сети S являются прототипами для вершин из семантической сети S_{inst} . Таким образом, S определяет, в сущности, язык для описания множества S_{inst} .

Под *структурной проекцией* (μ) (*спецификацией преобразования*) класса семантических сетей исходной информации на класс семантических сетей целевой информации будем понимать множество правил (соответствий) $\mu = \{Rule_i\}_{i=1}^{rulescount}$. Каждое правило $Rule_i$ описывает структуру подсети семантической сети целевой информации, являющейся экземпляром некоторой подсети из описания класса семантических сетей целевой информации, сопоставляемое понятию из описания класса семантических сетей исходной информации. Структурные проекции описываются в терминах *Модели описания структурных проекций*.

Используя введенные термины, *преобразование семантических сетей* можно определить как генерацию целевой семантической сети, являющейся экземпляром класса семантических сетей целевой информации, на основе исходной семантической сети, являющейся экземпляром класса семантических сетей исходной информации, по описанию структурной проекции класса семантических сетей исходной информации на класс семантических сетей целевой информации.

2. Модель описания классов семантических сетей

Определим *модель описания классов*

семантических сетей как тройку $\langle Concepts_Network, Id_Concept, Syntax_Restrictions \rangle$. Рассмотрим далее каждый компонент данной модели.

1. $Concepts_Network (S_{nc}) = \langle Concepts, Relations, Initial_Concept \rangle$ – описание класса сетей понятий. Данное описание базируется на модели представления семантических сетей, предложенной в рамках архитектуры ИРУО [Орлов, 2006а], [Орлов, 2006б].

1.1. $Concepts = \{Concept_i\}_{i=1}^{conceptscount}$ – конечное непустое множество понятий (включая $Initial_Concept$). Каждое понятие $Concept_i$ описывается следующим образом: $Concept_i = \langle Concept_Value, Concept_Kind, Concept_TerminalType \rangle$.

1.1.1. $Concept_Value = \langle Concept_Value_Type, Concept_Value_Value \rangle$ – значение понятия описывается своим типом $Concept_Value_Type$ и значением $Concept_Value_Value$. Значение понятия есть непустая последовательность символов, которая идентифицирует определенный класс объектов описываемой предметной области, множество значений (сорт) или представляет в ней конкретное (константное) значение.

$Concept_Value_Type \in \{“Строковое”, “Целое”, “Вещественное”, “Логическое”, “Бинарные данные”\}$.

$Concept_Value_Value \in String \cup Integer \cup Real \cup Boolean$. $String$ – множество строк. $Integer$ – множество целых чисел. $Real$ – множество вещественных чисел. $Boolean$ – множество $\{“true”, “false”\}$. Значением понятия может быть последовательность символов, интерпретируемых как строковая константа ($Concept_Value_Type = “Строковое”$), целое число из множества целых чисел ($Concept_Value_Type = “Целое”$), вещественное число из множества вещественных чисел ($Concept_Value_Type = “Вещественное”$), элемент множества $\{“true”, “false”\}$ ($Concept_Value_Type = “Логическое”$), бинарные данные ($Concept_Value_Type = “Бинарные данные”$).

1.1.2. $Concept_Kind \in \{“Терминальное”, “Нетерминальное”\}$ – тип понятия. Понятие может быть терминальным ($Concept_Kind = “Терминальное”$) или нетерминальным ($Concept_Kind = “Нетерминальное”$).

1.1.3. $Concept_TerminalType \in \{“Идентификатор”, “Константа”, “Сорт”, “Не определено”\}$ – тип терминального понятия. Терминальное понятие может быть идентификатором ($Concept_TerminalType = “Идентификатор”$), константой ($Concept_TerminalType = “Константа”$), или обозначать некоторый сорт ($Concept_TerminalType = “Сорт”$). Если понятие $Concept_i$ является нетерминальным, то $Concept_TerminalType = “Не определено”$.

1.2. $Relations = \{Relation_i\}_{i=0}^{relationscount}$ – конечное, возможно пустое, множество отношений. Каждое отношение $Relation_i$ является направленным бинарным отношением (дугой), имеющим, возможно, спецификатор множественности и связывающим два понятия. Отношение описывается следующим образом: $Relation_i = \langle Relation_EndSp, Begin_Concept, End_Concept \rangle$.

1.2.1. $Relation_EndSp \in \{“Конкретность”, “Единственность”, “Множество”\}$ – спецификатор множественности (кардинальность) – характеристика отношения, определяющая способ порождения понятия-экземпляра (понятия из S_{inst}) вместе с отношением к нему по понятию-концу данного отношения ($End_Concept$ – понятие из S_{nc}). $Relation_EndSp = “Конкретность”$ (или пустой спецификатор) означает, что может быть порождено только одно понятие-экземпляр и его значение (имя) должно совпадать со значением (именем) его понятия-прототипа ($End_Concept$), иначе говоря, в S_{inst} создается в точности такое же понятие как и понятие-прототип из S_{nc} . $Relation_EndSp = “Единственность”$ означает, что по $End_Concept$ может быть порождено только одно понятие-экземпляр, но его значение (имя) обязательно должно быть задано при порождении. Порожденное понятие при этом является сущностью из класса объектов или элементом из множества значений, идентифицируемого понятием-прототипом ($End_Concept$) (строка, целое значение, вещественное значение, логическое значение). $Relation_EndSp = “Множество”$ означает, что по $End_Concept$ может быть порождено любое количество (но, по крайней мере, одно) понятий-экземпляров и их значения (имена) обязательно должны быть заданы при порождении. Каждое порожденное понятие при этом является сущностью из класса объектов или элементом из множества значений, идентифицируемого понятием-прототипом ($End_Concept$).

1.2.2. $Begin_Concept$ – понятие, из которого дуга исходит – понятие-начало отношения. $Begin_Concept \in Concepts \setminus Terminal_Concepts$, где $Terminal_Concepts$ – множество терминальных понятий, $Terminal_Concepts \subset Concepts$. Понятием-началом отношения может быть любое нетерминальное понятие.

1.2.3. $End_Concept$ – понятие, в которое дуга входит – понятие-конец отношения. $End_Concept \in Concepts \setminus \{Initial_Concept\}$. Понятием-концом отношения может быть любое понятие за исключением *начального понятия*.

1.3. $Initial_Concept$ – начальное понятие в описании класса сетей понятий, оно единственно, и через него не может быть выражено ни одно другое понятие из описываемого класса сетей понятий. $Initial_Concept \in Concepts$.

2. $Id_Concept$ – понятие, представляющее идентификатор в описании класса сетей понятий,

оно единственно и не может быть выражено через другие понятия, т.е. является терминальным понятием типа “Сорт”. $Id_Concept \in Concepts$. Понятие выделяется отдельно (если требуется текстовое представление информации) для того, чтобы в абстрактном синтаксисе отличить идентификатор от строковой константы.

3. $Syntax_Restrictions = \langle Lexica, Syntax \rangle$ – описание синтаксических ограничений. Оно может отсутствовать, если текстовое представление информации не требуется.

3.1. $Lexica = \{\langle Lexem_Type_i, Definition_i \rangle\}_{i=1}^5$, где $Lexem_Type_i \in \{“Идентификатор”, “Целое число”, “Вещественное число”, “Строковая константа”, “Ограничитель строковой константы”\}$.

3.2. $Syntax = \{\langle Concept_i, Definition_i \rangle\}$, где $Concept_i \in Concepts \setminus Terminal_Concepts$. Синтаксические ограничения задаются для нетерминальных понятий. $Definition_i$ – строка символов, включающая метасимволы и представляющая конкретное лексическое или синтаксическое ограничение для конкретного понятия или вида лексемы.

Приведенная модель для описания синтаксических ограничений имеет свое представление в виде сети понятий, в терминах которой пользователь может формально описывать способ текстового представления информации. Далее подробно рассмотрена эта общая структура, которая может быть использована для представления синтаксических ограничений любого контекстно-свободного языка (рис. 1).

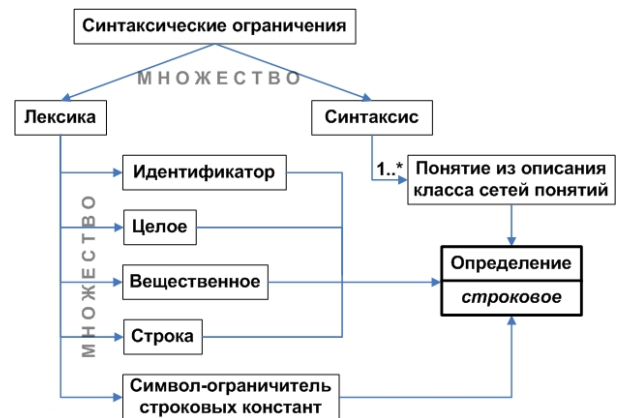


Рисунок 1 – Модель синтаксических ограничений

Синтаксические ограничения языка кроме, собственно, синтаксических определений, содержат еще и ограничения на вид лексем. Синтаксическое определение представляет собой определение объемлющего понятия из описания класса сетей понятий через объемлемые понятия и элементы конкретного синтаксиса языка текстового представления. В определении фиксируется также порядок следования объемлемых понятий и элементов конкретного синтаксиса.

Лексический словарь языка определяется

фиксированным набором вершин, соответствующих базовым типам данных – целое, вещественное и строковое и, кроме этого, определением вида идентификатора или имени. Все ограничения на вид лексем задаются с помощью регулярных выражений с использованием нотации perl5 [Christiansen, 1996]. Строка, представляющая синтаксическое определение, содержит специальные метасимволы, которые будут описаны ниже. Метасимволы не распространяются на лексические ограничения.

Синтаксические ограничения – это корневая вершина в сети понятий, соответствующей модели синтаксических ограничений. Корневой вершиной сети понятий, задающей синтаксические ограничения для конкретного вида информации, должна быть вершина-экземпляр этой вершины “Синтаксические ограничения” с точно таким же названием.

Лексика – в сети понятий, соответствующей синтаксическим ограничениям для конкретного вида информации, должна быть единственная вершина-экземпляр вершины “Лексика” с таким же точно названием. Эта вершина является корнем поддерева, задающего лексические ограничения конкретного вида информации.

Идентификатор – в сети понятий, соответствующей синтаксическим ограничениям для конкретного вида информации, должна быть единственная вершина-экземпляр вершины “Идентификатор” с таким же точно названием. Эта вершина задает ограничения на вид идентификаторов.

Строка – в сети понятий, соответствующей синтаксическим ограничениям для конкретного вида информации, должна быть единственная вершина-экземпляр вершины “Строка” с таким же точно названием. Эта вершина задает ограничения на вид строковых констант.

Символ-ограничитель строковых констант – символ, указывающий начало и конец строковой константы. При синтаксическом анализе эта информация не существенна, ограничители строки можно использовать наряду с другими элементами строковой константы в регулярном определении, например, следующее регулярное выражение задает строку символов следующим образом: “*строка, состоящая из нуля или более символов латинского алфавита (прописных и/или строчных) и также символа пробела, ограниченная одинарными кавычками, причем кавычки исключаются*”: $(?<=^)[a-zA-Z,]*(?='\$)$. При синтезе текста нужно “знать”, как выделить строковую константу в синтезируемом тексте, поэтому этот элемент нужно задать, осуществляя синтез. При анализе достаточно регулярного определения.

Целое – в сети понятий, соответствующей синтаксическим ограничениям для конкретного вида информации, должна быть единственная вершина-экземпляр вершины “Целое” с таким же

точно названием. Эта вершина задает ограничения на вид целочисленных констант.

Вещественное – в сети понятий, соответствующей синтаксическим ограничениям для конкретного вида информации, должна быть единственная вершина-экземпляр вершины “Вещественное” с таким же точно названием. Эта вершина задает ограничения на вид вещественных констант.

Синтаксис – в сети понятий, соответствующей синтаксическим ограничениям для конкретного вида информации, должна быть единственная вершина-экземпляр вершины “Синтаксис” с таким же точно названием. Эта вершина является корнем поддерева, задающего синтаксические ограничения конкретного вида информации.

Понятие из описания класса сетей понятий – это нетерминальная вершина, которой соответствует некоторое понятие из S_{nc} , синтаксически охватывающее другие понятия.

Определение – это терминальная вершина, описывающая строковый тип, которой соответствует лексическое или синтаксическое ограничение для понятия, представленного родительской вершиной. В сети понятий, описывающей синтаксические ограничения для конкретного вида информации, вершиной, соответствующей данной вершине будет являться терминальная вершина, значение которой задает в виде строки конкретное лексическое или синтаксическое ограничение для конкретного понятия из описания класса сетей понятий или вида лексем.

В сети понятий, задающей синтаксические ограничения для конкретного класса семантических сетей, у всех вершин, прототипами которых являются вершины из сети понятий, представляющей модель синтаксических ограничений, с потомком “определение”, может быть единственный потомок. Другими словами, для лексика и синтаксиса может быть определено единственное ограничение для каждого элемента.

Строка, представляющая синтаксическое определение, может содержать следующие метасимволы:

« » – пробел является символом разделителем и используется для наглядности. Наличие или отсутствие пробела не меняет смысла записи.

|| – две вертикальные черты. Являются метасимволом конкатенации синтаксических конструкций.

? – знак вопроса. Является метасимволом факультативности. Означает нуль или одно вхождение элемента, за которым он находится. Этим элементом может быть как понятие, так и элемент конкретного синтаксиса. В общем случае одно синтаксическое определение, включающее вхождение метасимвола факультативности,

эквивалентно 2^n синтаксическим определениям без этого метасимвола, где n – количество факультативных элементов в этом ограничении.

{...} – метасимвол перечисления. В фигурные скобки заключается понятие из описания класса сетей понятий (S_{nc}), *Relation_EndSp* отношения, концом которого является данное понятие, есть “Множество”. Другими словами, понятие, являющееся экземпляром данного понятия в S_{inst} , может входить в объемлющее понятие (также из S_{inst}) произвольное число раз. Первый символ после фигурной скобки рассматривается как символ-разделитель. Этим символом, помимо символа конкатенации, в тексте будут разделены повторяющиеся элементы. Если символ-разделитель состоит из нескольких литер, то такой символ заключается в двойные кавычки. То же самое можно сделать и с разделителем, записываемым одной литерой, например, {“;”operator} вместо {;operator}. Строка после символа-разделителя до второй фигурной скобки вся рассматривается как имя некоторого понятия из описания класса сетей понятий S_{nc} .

(...|...|...) – альтернатива. Определяет возможность выбора одного из множества вариантов синтеза или анализа текста. Элементы альтернативы разделяются вертикальной чертой – |. Все эти элементы, так же, как и в случае перечисления, рассматриваются как имена понятий из описания класса сетей понятий S_{nc} . Строка, содержащая символ альтернативы, не может содержать других конструкций вне скобок, относящихся к определяемой альтернативе. Многоточие “...” не является метасимволом и означает, что в этой позиции должно находиться имя некоторого понятия из описания класса сетей понятий (S_{nc}).

”” – двойные кавычки. Используются для определения элементов конкретного синтаксиса, соответствующего описываемому синтаксическому ограничению.

\ – обратная косая черта. Используется для определения элементов конкретного синтаксиса языка, которые записываются теми же литерами, что и служебные символы.

Пусть, например, в качестве метаинформации выбрано описание языка Милан [Ершов, 1977]. Описание класса сетей понятий (S_{nc}), представляющее устройство языка Милан, показано на рис. 2. Здесь “Program MILAN” – начальное понятие, “id” – понятие, представляющее идентификатор. Символ “+” представляет спецификатор множественности отношения “множество”, а символ “!” – “единственность”.

Представление программы на языке Милан в виде семантической сети (информация S_{inst}) показано на рис. 3. На рисунке также представлено текстовое представление этой программы,

соответствующее представлению в виде семантической сети понятий.

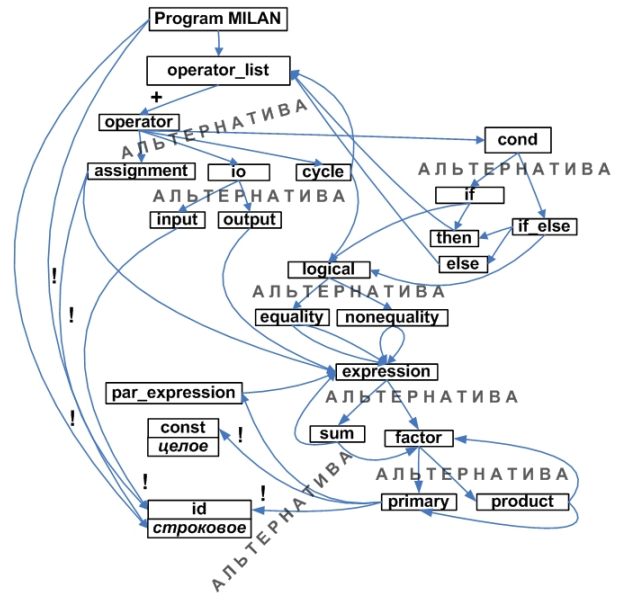


Рисунок 2 – Описание класса сетей понятий, представляющего устройство языка Милан

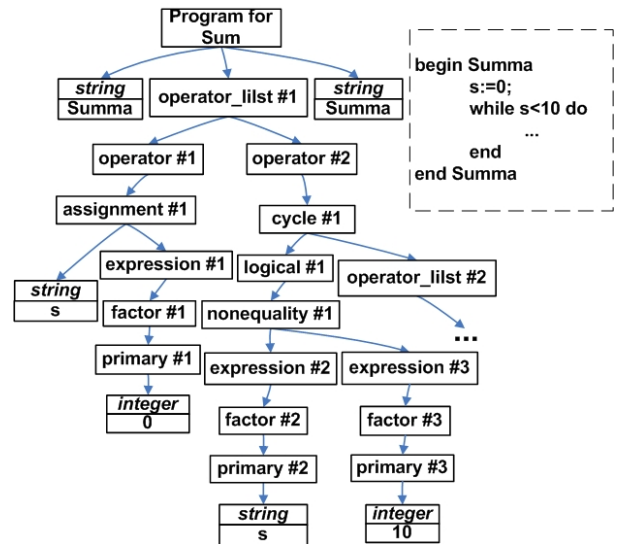


Рисунок 3 – Программа в терминах сети понятий, описывающей устройство языка Милан

Фрагмент описания синтаксических ограничений языка Милан представлен на рис. 4.

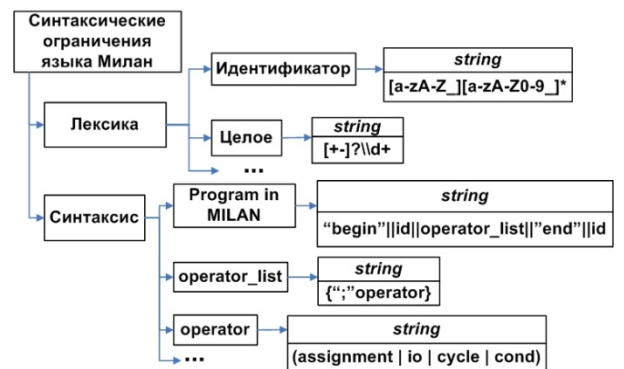


Рисунок 4 – Фрагмент описания синтаксических ограничений для языка Милан

3. Модель описания структурных проекций

В определении структурной проекции было сказано, что описание проекции μ представляет собой множество правил (соответствий) $\mu = \{Rule_i\}_{i=1}^{rulescount}$. В общем виде каждое правило $Rule_i$ можно представить следующим образом: $\alpha \rightarrow \beta$, где $\alpha \in Source_Concepts$, где $Source_Concepts$ – множество понятий из описания класса семантических сетей исходной информации SS_{nc} ;

$$\beta = \{Target_Concept_Descr_i\}_{i=1}^{conceptscount} \cup \{Target_Relation_Descr_j\}_{j=0}^{relationscount}, \text{ где}$$

$\{Target_Concept_Descr_i\}_{i=1}^{conceptscount}$ – возможно пустое множество описаний порождаемых понятий в семантической сети целевой информации (TS_{inst}). $Target_Concept_Descr_i$ есть описание порождения понятия в TS_{inst} . $\{Target_Relation_Descr_j\}_{j=0}^{relationscount}$

– возможно пустое множество описаний порождаемых отношений в семантической сети целевой информации (TS_{inst}). $Target_Relation_Descr_j$ есть описание порождения отношения в TS_{inst} . При этом $\beta \neq \emptyset$ и для любых $\alpha_1, \alpha_2 \in Source_Concepts$, если правила $\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2$ входят в μ , то $\alpha_1 \neq \alpha_2$.

Описание создания понятия в TS_{inst} ($Target_Concept_Descr$) представляет собой шестерку: $Target_Concept_Descr = \langle Proto_Concept_Name, SConceptValue, isTProtoConceptName, Storable, Loadable, Concept_Value \rangle$.

1. $Proto_Concept_Name$ – имя понятия-прототипа из описания класса сетей понятий целевой информации (TS_{nc}), которое представляет собой непустую последовательность символов.

2. $SConceptValue = \langle isSConceptValue, isTerminal \rangle$. Описывает выбор значения для создаваемого в TS_{inst} понятия на основе имени (значения) понятия из семантической сети исходной информации (SS_{inst}).

2.1. $isSConceptValue \in \{\text{“true”}, \text{“false”}\}$. Определяет, нужно ли создаваемому в TS_{inst} понятию присваивать в качестве значения имя (значение) понятия из SS_{inst} .

2.2. $isTerminal \in \{\text{“true”}, \text{“false”}\}$. Определяет, значение какого понятия из SS_{inst} нужно присвоить: экземпляра понятия, стоящего в левой части правила (“false”), или терминального понятия-потомка экземпляра понятия, стоящего в левой части правила (“true”).

3. $isTProtoConceptName \in \{\text{“true”}, \text{“false”}\}$. Определяет, нужно ли создаваемому в TS_{inst} понятию присваивать в качестве значения имя (значение) его понятия-прототипа из TS_{nc} .

4. $Storable = \langle isStorable, Alias, storeConcept \rangle$. Описывает сохранение значений в хеш-таблицу.

4.1. $isStorable \in \{\text{“true”}, \text{“false”}\}$. Определяет, нужно ли значение, которое будет присвоено создаваемому в TS_{inst} понятию, поместить (сохранить) в хеш-таблицу.

4.2. $Alias$ – строка символов, возможно пустая, представляющая псевдоним, под которым нужно сохранить значение понятия в хеш-таблицу.

4.3. $storeConcept \in \{\text{“true”}, \text{“false”}\}$. Определяет, нужно ли поместить (сохранить) в хеш-таблицу создаваемое в TS_{inst} понятие целиком или же только его значение (имя).

5. $Loadable = \langle isLoadable, Alias \rangle$. Описывает извлечение значений из хеш-таблицы.

5.1. $isLoadable \in \{\text{“true”}, \text{“false”}\}$. Определяет, будет ли значение, которое необходимо присвоить создаваемому в TS_{inst} понятию, извлекаться из хеш-таблицы.

5.2. $Alias$ – строка символов, возможно пустая, представляющая псевдоним, по которому нужно получить значение для понятия из хеш-таблицы.

6. $Concept_Value = \langle Concept_Value_Type, Concept_Value_Value \rangle$. Описывает значение (имя), присваиваемое создаваемому в TS_{inst} понятию.

6.1. $Concept_Value_Type \in \{\text{“Строковое”}, \text{“Целое”}, \text{“Вещественное”}, \text{“Логическое”}, \text{“Бинарные данные”}, \text{“Вычислимое целое”}, \text{“Вычислимое вещественное”}, \text{“Вычислимое строковое”}\}$ – тип значения.

6.2. $Concept_Value_Value$ – собственно значение понятия – строка символов, интерпретируемая в зависимости от значения типа понятия. Значением понятия может быть строковая константа ($Concept_Value_Type = \text{“Строковое”}$), целое число из множества целых чисел ($Concept_Value_Type = \text{“Целое”}$), вещественное число из множества вещественных чисел ($Concept_Value_Type = \text{“Вещественное”}$), элемент множества $\{\text{“true”}, \text{“false”}\}$ ($Concept_Value_Type = \text{“Логическое”}$), бинарные данные ($Concept_Value_Type = \text{“Бинарные данные”}$). Если тип значения понятия “Вычислимое целое” ($Concept_Value_Type = \text{“Вычислимое целое”}$), то значение понятия интерпретируется как выражение, значение которого требуется вычислить, и результат вычисления будет интерпретироваться как целое число. Переменные в этом выражении могут инициализироваться либо значением понятия из семантической сети понятий, представляющей исходную информацию SS_{inst} , либо значениями из хеш-таблицы. Если тип значения понятия “Вычислимое вещественное” ($Concept_Value_Type = \text{“Вычислимое вещественное”}$), то аналогично предыдущему случаю, но результат вычисления будет интерпретироваться как вещественное число. Если тип значения понятия “Вычислимое строковое” ($Concept_Value_Type = \text{“Вычислимое строковое”}$), то вычисленное значение будет строковой константой.

Описание порождения отношения в TS_{inst}

(*Target_Relation_Descr*) представляет собой тройку: $Target_Relation_Descr = \langle Relation_Begin_Concept, Relation_End_Concept, Number \rangle$.

1. $Relation_Begin_Concept = \langle Begin_Concept_Name, isProtoConceptName, isLoadable, Alias \rangle$. Описывает понятие-начало отношения.

1.1. *Begin_Concept_Name* – имя понятия-начала отношения, представляющее собой непустую последовательность символов.

1.2. $isProtoConceptName \in \{“true”, “false”\}$. Определяет, является ли это имя именем понятия из TS_{nc} .

1.3. $isLoadable \in \{“true”, “false”\}$. Определяет, будет ли понятие-начало отношения, извлекаться из хеш-таблицы.

1.4. *Alias* – строка символов, возможно пустая, представляющая псевдоним, по которому нужно получить понятие из хеш-таблицы.

2. $Relation_End_Concept = \langle End_Concept_Name, isProtoConceptName, isLoadable, Alias \rangle$. Описывает понятие-конец отношения. Аналогично описанию $Relation_Begin_Concept$, только *End_Concept_Name* есть имя понятия-конца отношения, которое также представляет собой непустую последовательность символов.

3. *Number* – целое положительное число – порядковый номер отношения. Определяет номер, под которым создаваемое отношение будет добавлено к отношениям, в которых уже участвует понятие-начало.

Для формирования пользователем множества правил в терминах приведенной выше модели разработан язык для описания структурных проекций. Специфицированы его абстрактный и конкретный синтаксис, а также операционная семантика.

ЗАКЛЮЧЕНИЕ

Разработана модель для описания классов семантических сетей. Данная модель базируется на формализме представления семантических сетей, предложенном в рамках архитектуры ИРУО, которая в свою очередь основана на максимально абстрактных категориях “сущность” и “отношение” и предлагает схему организации понятий, предназначенную для описания информационных моделей различных уровней общности. Расширение модели позволяет специфицировать преобразования вида “текст – семантическая сеть” и “семантическая сеть – текст” одинаковым образом (используется один и тот же формализм, нотация, вид правил и т.д.). Это означает, в частности, что при выполнении эндогенных преобразований одна и та же спецификация используется для выполнения этих двух типов преобразований. При этом в обоих

случаях никаких дополнительных описаний не требуется.

Разработаны модель и язык для описания структурных проекций, предназначенные для описания преобразований вида “семантическая сеть – семантическая сеть”. В отличие от многих других моделей и языков для описания спецификаций преобразования они не ориентированы на конкретное технологическое пространство и позволяют специфицировать преобразования в терминах модели для описания классов семантических сетей.

Работа выполнена при финансовой поддержке РФФИ, проект 10-07-00089-а.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[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.

[Wimmer, 2007] Wimmer, M. Towards Model Transformation Generation By-Example / M. Wimmer, M. Strommer, H. Kargl, G. Kramler // In Proceedings of the 40th Hawaii International Conference on Systems Science (HICSS-40), [Electronic resource], CD-ROM / Abstracts Proceedings, Big Island, HI, USA. – 2007. – 285 p.

[Орлов, 2006a] Орлов, В.А. Компьютерные банки знаний. Универсальный подход к решению проблемы редактирования информации / В.А. Орлов, А.С. Клещев // Информационные технологии. – 2006. – №5. – С. 25 – 31.

[Орлов, 2006b] Орлов, В.А. Компьютерные банки знаний. Модель процесса редактирования информационного наполнения / В.А. Орлов, А.С. Клещев // Информационные технологии. – 2006. – №7. – С. 11 – 16.

[Christiansen, 1996] Christiansen, T. PERL5 Regular Expression Description / T. Christiansen // [Electronic resource]. URL: <http://www.perl.com/doc/FMTEYEWTK/regexps.html>. – 1996. (дата обращения: 15.05.2010).

[Ершов, 1977] Ершов, А.П. О сущности трансляции / А.П. Ершов // Препринт, ВЦ Сибирское отделение АН СССР, Новосибирск. – 1977.

THE MODEL OF SEMANTIC NETWORK CLASSES AND THEIR TRANSFORMATIONS

Timchenko V.A.

Institute of Russian Academy of Sciences Institute of Automation and Control Processes Far Eastern Branch of RAS Vladivostok, Russia

rakot2k@mail.ru

The paper presents a model for semantic networks classes description, as well as a model for structure mappings description. The latter is model for semantic networks transformations definitions in terms of their classes. Both models are invariant to the different technological spaces. The model for semantic networks classes includes a formalism for describing a relationship between the semantic networks class and

the concrete syntax elements of the language for their textual representation.

INTRODUCTION

The problem of different types of information (data, knowledge, computer programs) transformation is still knotty and, at the same time, one of the crucial for research. Different specialists in their professional fields deal with such problems. These fields include, for example, software engineering, knowledge engineering, ontologies engineering, development of the systems for the analysis and transformation of programs, models, etc.

Semantic networks have been successfully used for representation of a various types of data, knowledge, ontologies in different domains. Therefore many conceptual problems of transformation information appearing in different technological spaces [Kurtev, 2002] [Bezivin, 2005], as well as at their joint, can be formulated in terms of the semantic networks transformation. 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]. Therefore, if the problem of transformation arises at the joint of different technological spaces and this class of problems does not solved with some software system, it is necessary in this case to develop a new specialized transformation tool or to adapt an existing one.

In particular, the existing models for the definition of a meta-information (in fact classes of semantic networks) provide a level of abstraction and notation generally accepted in the technological spaces on which these models are oriented. For example, the MOF language and its dialects are oriented on the modelware technology space, EBNF and other formalisms for the grammars specification are oriented on the grammarware technological space [Wimmer, 2007]. Therefore, the definition in terms of such models the meta-information from different technological spaces is either impossible or very difficult.

The paper presents the model of semantic networks classes and the model for their transformations. Both models are invariant to the different technological spaces. These models allow to define both structural transformations and transformations from textual representation of information into structural representation (in the form of a semantic network) and vice versa.

MAIN PART

The first section introduces the basic terms used further and their definitions: *semantic networks class*, *semantic network instance*, *the specification of the of semantic networks transformation*, *the transformation of semantic networks*.

The second section presents the model for semantic networks classes description. The model is based on the formalism for semantic networks representation proposed in the IDEA framework. This model includes a formalism for describing a relationship between the semantic networks class and the concrete syntax elements of the language for their textual representation – the syntax restrictions. Each syntax restriction defines a nonterminal concept from a semantic networks class through others nonterminal or terminal concepts from the semantic networks class and the concrete syntax elements. The definition fixes the sequence of the concepts and elements of the concrete syntax. The lexeme corresponds to one of basic data types – integer, real and string or to identifier. All the restrictions on the type of CF language lexemes are specified with regular expressions using the perl5 notation.

The third section presents the model for semantic networks transformations definitions in terms of their classes. Each rule matches the concept from source semantic networks class and the description of target semantic network subnet structure generation. This subnet is an instance of a target semantic networks class subnet. The structure of the target semantic networks subnet in each rule is defined by the description the set of concepts and relationships generation. The techniques of the concepts (relations) generation in the target semantic network from their meta-concepts (meta-relations) are described.

CONCLUSION

The model for semantic networks classes description is developed. This model is based on the formalism for semantic networks representation proposed in the IDEA framework. The formalism is in turn based on the most abstract categories of “essence” and “relation” and provides a possibility for data models of different levels of generality definition. The extension of the model allows specifying “text-to-semantic network” and “semantic network-to-text” transformations in the same way (using the same formalism, notation, rules, etc.). It means in particular that if an endogenous transformation is performed the same specification is used for these two types of transformations. In both cases, no additional definitions are required.

The model and language for structure mappings description are developed. They are used for “semantic network-to-semantic network” transformations definition. Unlike many other models and languages for the transformation specifications definition, they are invariant to the different technological spaces and allow specifying the transformations in terms of the model for semantic networks classes description.

The paper was written with financial support from RFBR, project 10-07-00089-a.



УДК 519.711.74

ДИСКРЕТНЫЕ МОДЕЛИ РАССЕЙЯНИЯ НА ГРАФАХ

Жилякова Л.Ю.

*Институт проблем управления им. В.А. Трапезникова РАН,
г. Москва, Россия*

zhilyakova.ludmila@gmail.com

Рассматриваются две модели рассеяния на графах: chip-firing game – целочисленная пороговая модель, и неоднородная ресурсная сеть. Рассматриваются сходства и различия этих моделей, а также области их применения.

Ключевые слова: граф, дискретная модель, рассеяние, ресурс.

ВВЕДЕНИЕ

Модели рассеяния ресурсов на графах используются для решения различных задач в широком кругу предметных областей. Показано, что явления самоорганизующейся критичности [Bak et al., 1988], [Dhar, 1990], такие как сход лавин, можно имитировать достаточно простой целочисленной моделью. Модели катастроф: «лавина» и «абелева куча песка», модели балансирования нагрузки в распределенных сетях и т.д. при некоторых упрощениях математически эквивалентны играм «выстреливания фишек» (chip-firing games) (см. напр., [Bjorner et al., 1991, 1992], [Lovasz et al., 1995]).

Ресурсная сеть, предложенная в [Кузнецов, 2009], также является моделью рассеяния ресурсов. Однако она обладает несколькими отличительными особенностями, позволяющими с ее помощью моделировать более разнообразные процессы, такие, к примеру, как распространение вещества в водной среде [Жилякова, 2011].

1. Chip-firing game и связанные с ней модели

Пусть G – ориентированный граф, который может иметь петли и кратные дуги. В начальном состоянии в некотором множестве вершин графа содержатся стопки фишек. «Ход» состоит в выборе вершины, число фишек в которой не меньше числа ее исходящих дуг, после чего эта вершина выстреливает, т.е. передает по одной фишке по каждой исходящей дуге всем своим соседям. Игра продолжается до тех пор, пока остается хотя бы одна вершина, способная выстрелить.

Ключевое свойство таких игр – то, что из данной позиции все последовательности выстрелов ведут себя одинаково: либо все они будут длиться бесконечно, либо закончатся за одинаковое количество шагов с одной и той же конечной позицией [Lovasz et al., 1995].

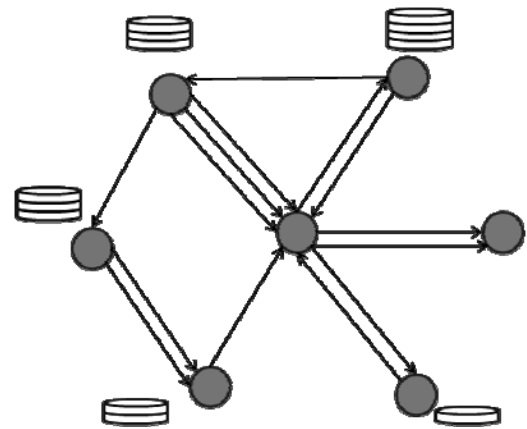


Рисунок 1 – Распределение фишек в chip-firing game с ориентированным графом

Для игры в выстреливания (chip-firing game) можно задать ряд естественных вопросов: будет ли процесс конечным или бесконечным? Если конечным, как долго он будет длиться? Если бесконечным, как скоро он заикнется? Сколько фишек необходимо для бесконечной игры? Как конкретная начальная позиция может быть трансформирована с помощью выстреливаний в другую позицию?

В случае неориентированных графов на эти вопросы получены ответы. Например, если n –

количество вершин, m – количество дуг в графе, то конечная процедура заканчивается за $O(n^4)$ шагов. Кратчайший период в периодической игре равен n ; минимальное число фишек для бесконечной игры равно m , количеству дуг. Существуют алгоритмы полиномиальной сложности для определения, задает ли начальная позиция конечную или бесконечную игру, а также для определения, достижима ли одна позиция из другой. Орграфы более трудны для изучения, и сложность некоторых вопросов для них пока неизвестна.

1.1. Модель «абелева куча»

Игрой в выстреливания на орграфе может быть представлена модель самоорганизующейся системы, называемая «абелева куча песка» или «лавины». В этой модели каждый узел представляет собой место, в котором аккумулируется снег. Специальный узел s отвечает за «внешний мир». Это узел, из которого идет снег. Он связан со всеми остальными узлами сети. Когда в каком-либо месте количество снега превышает порог, узел «обрушивается», посылая по единице снега каждому исходящему соседу, которые, в свою очередь, тоже могут обрушиться, и т.д., начиная лавину.

Снегопад включается в модель. Для этого узел s соединяется с каждым узлом i посредством a_i дуг. Снегопад соответствует выстреливанию узла s . Предполагается, что снега в s «достаточно» для того, чтобы он стрелял всегда.

Последовательность лавина-снегопад-лавины-снегопад-... задает бесконечную игру в выстреливания на графе с дополнительным ограничением: узел s стреляет только тогда, когда ни один другой узел не стреляет. Когда лавина начинает сходить, очередность выстреливания узлов может быть различной, однако количества выстреливаний каждого узла и конечное распределение снега определены однозначно.

Недостатком модели является ее целочисленность и отсутствие параллельности. За один ход выстреливает только один узел.

1.2. Балансирование нагрузки в сетях

Балансирование нагрузки в распределенных сетях тесно связано с рассматриваемыми процессами на графах [Spencer, 1986]. В этой модели каждый узел (обычно регулярного неориентированного) графа соответствует процессору, и каждый процессор i имеет определенное количество рабочей нагрузки w_i . Процессоры хотят передать нагрузку другим процессорам по дугам, с тем чтобы их нагрузка приблизительно выровнялась. По духу это близко случайному блужданию на регулярном графе, где «вероятность» передается по ребрам и, в конечном счете, выравнивается.

Конечно, нижняя и верхняя границы времени, необходимого для выравнивания нагрузок, включают в себя параметры, знакомые из теории случайных блужданий: скорость распространения,

проводимость, зазор между собственными значениями и т.д. [Aiello et al., 1993].

2. Ресурсные сети

Ресурсная сеть – динамическая модель, представленная взвешенным графом с вершинами $v_i \in V$, функционирующая в дискретном времени t .

Ребрам (v_i, v_j) приписаны неотрицательные числа r_{ij} , постоянные во времени и называемые *пропускными способностями*. Сеть называется *двусторонней*, если любые две смежные вершины связаны парой противоположно ориентированных ребер. Будем называть сеть *сетью с петлями*, если множество ребер вида (v_i, v_i) непусто.

Ресурсами $q_i(t)$ называются неотрицательные числа, приписанные вершинам v_i и изменяющиеся в дискретном времени t . Вершины v_i могут хранить неограниченное количество ресурса.

$Q(t) = (q_1(t), \dots, q_n(t))$ – состояние сети в момент времени t .

Состояние сети Q^* называется *предельным*, если оно либо устойчиво и достижимо из $Q(0)$ за конечное время, либо асимптотически достижимо из $Q(0)$.

$R = (r_{ij})_{n \times n}$ – матрица пропускной способности сети; $r_{sum} = \sum_{i=1}^n \sum_{j=1}^n r_{ij}$ – суммарная пропускная

способность сети; $r_i^{in} = \sum_{j=1}^n r_{ji}$ и $r_i^{out} = \sum_{j=1}^n r_{ij}$ –

входная и выходная пропускные способности вершины соответственно.

Суммарный ресурс, находящийся во всех вершинах, обозначим через W . В сети выполняется *закон сохранения*: при ее функционировании ресурс остается постоянным (не поступает извне и не расходуется).

Правила распределения ресурса. В момент t вершина v_i отдает в смежную ей вершину v_m :

r_{im} единиц ресурса, если $q_i(t) > r_i^{out}$ (правило 1);

$\frac{r_{im}}{r_i^{out}} q_i(t)$ – в противном случае (правило 2).

Ресурс, выходящий из вершины v_i по ребру (v_i, v_m) в момент t , приходит в вершину v_m в момент $t+1$.

Множество вершин с ресурсом $q_i(t)$, не превосходящим r_i^{out} , назовем зоной $Z(t)$. Вершины из $Z(t)$ функционируют по правилу 2.

$Z^+(t)$ – множество вершин, ресурс которых больше их выходной пропускной способности, они

функционируют по правилу 1. Для предельного состояния Q^* обозначим эти зоны через Z^* и Z^{+*} .

T – пороговое значение ресурса, такое что при $W \leq T$ все вершины начиная с некоторого t' переходят в зону $Z(t)$; при $W > T$ зона $Z^+(t)$ пуста начиная с некоторого t'' . Доказаны существование и единственность T для любой связной двусторонней несимметричной сети с петлями.

При $W \leq T$ процесс распределения ресурса представляет собой регулярную сеть Маркова. Предельное состояние существует и единственно для любой матрицы пропускной способности R , соответствующей двусторонней сети с петлями, и любого начального распределения ресурса $Q(0)$.

Процессы распределения в ресурсных сетях были исследованы в ряде работ (О.П. Кузнецовым и автором). Доказана стабилизация состояний и потоков при любых значениях ресурса в сети; сформулированы достаточные условия единственности предельного состояния. Произведена классификация сетей в соответствии с их топологией и выявлены особенности функционирования каждого класса сетей при больших значениях ресурса.

Из определений следует, что в ресурсной сети, в отличие от chip-firing game, мульти-ребра заменяются одним ребром с соответствующей пропускной способностью. Целочисленность модели chip-firing game накладывает на нее ряд серьезных ограничений, которые отсутствуют в ресурсной сети.

Главное отличие этих двух моделей состоит в их функционировании при больших t . Модель выстреливаний останавливается или заикливается, проходя ряд последовательных состояний – в зависимости от количества фишек и их начального распределения. Ресурсная сеть из любого начального состояния стремится к предельному, причем процесс перераспределения ресурса продолжается бесконечно и поток ресурса стабилизируется. (Формулы для координат вектора предельного потока найдены для любой топологии сетей). Определение потока в ресурсных сетях является обобщением понятия потока в графовых моделях. В классической модели поток имеет направленность от источника к стоку; в ресурсной сети поток не направлен, источники и стоки отсутствуют.

2.1. Потоки в ресурсных сетях

Традиционное понятие потока [Форд и др., 1966], [Ahuja et al., 1993] для ресурсных сетей определяется следующим образом. Ресурс, выходящий из вершины v_i по ребру (v_i, v_j) в момент t , приходит в вершину v_j в момент $t + 1$. Соответственно, будем считать, что этот ресурс на интервале $(t, t + 1)$ находится на ребре (v_i, v_j) . Его величину назовем *потоком* $f_{ij}(t)$.

$$\sum_{j=1}^n f_{ij}(t) = f_i^{out}(t) \quad - \text{выходной поток из}$$

вершины v_i в момент t .

$$\sum_{i=1}^n f_{ij}(t) = f_j^{in}(t + 1) \quad - \text{входной поток в}$$

вершину v_j ; кроме того, положим $f_j^{in}(0) = 0$.

Общий поток $F(t)$ описывается матрицей $F(t) = (f_{ij}(t))_{n \times n}$.

Величиной потока будем называть сумму:

$$f_{sum}(t) = \sum_{i=1}^n \sum_{j=1}^n f_{ij}(t).$$

Сети разных топологий имеют несходные свойства. Рассмотрим некоторые классы ресурсных сетей отдельно.

2.2. Однородные полные сети

Однородными называются сети с равной пропускной способностью ребер. Пусть пропускная способность каждого ребра равна r .

В полной двусторонней сети с петлями динамический процесс распределения ресурсов сходится к предельному состоянию следующим образом:

2.2.1. Если суммарный ресурс W сети не превосходит порога $T = rn^2$, то при любом начальном состоянии сети ее предельным

состоянием является вектор $(\frac{W}{n}, \frac{W}{n}, \dots, \frac{W}{n})$. Для

таких сетей предельное состояние достигается за конечное время.

2.2.2. Если $W > rn^2$, то при любом начальном состоянии сети, в котором не все ресурсы вершин равны, происходит следующее. Будем говорить, что вершина v_i находится в зоне Z в момент t , если $q_i(t) \leq rn$, и в зоне Z^+ в противном случае. Тогда вершины, попавшие в зону Z , не смогут из нее выйти, а в предельном состоянии они имеют ресурс rn ; ресурс вершин, оставшихся в Z^+ , отличается от их начального ресурса на одну и ту же величину.

2.3. Несимметричные сети

Несимметричные ресурсные сети – частный случай неоднородных сетей, в которых пропускные способности произвольны. Введем обозначение $\Delta r_i = r_i^{in} - r_i^{out}$.

В общем случае вершины сети делятся на три класса:

- 1) вершины-приемники, для которых $\Delta r_i > 0$;
- 2) вершины-источники, $\Delta r_i < 0$;
- 3) нейтральные вершины, $\Delta r_i = 0$.

Несимметричной назовем сеть, которая обладает хотя бы одним источником и одним приемником.

Зона $Z(t)$ для таких сетей – это множество вершин, для которых $q_i(t) \leq r_i^{out}$.

Для несимметричных двусторонних сетей с петлями справедливы следующие утверждения:

2.3.1. Для любого начального состояния и любого W существует такой момент t' , что для всех $t > t'$ все источники и некоторые нейтральные вершины окажутся в зоне Z .

2.3.2. При любом W , при котором, начиная с некоторого момента t' , все вершины переходят в зону $Z(t)$, для любого $Q(0)$ предельное состояние существует и единственно.

2.3.3. Существует порог T , такой, что при $W \leq T$ все вершины, начиная с некоторого t' , переходят в зону $Z(t)$; при $W > T$ зона $Z^+(t)$ непуста, начиная с некоторого t'' . Для любой сети T единственен и не зависит от суммарного ресурса W и его начального распределения $Q(0)$. Кроме того, в несимметричных сетях T всегда меньше суммарной пропускной способности сети.

2.3.4. Для случая $W \leq T$ предложен метод вычисления предельного состояния. Он основан на том, что в предельном состоянии для любой вершины входной поток равен выходному.

2.3.5. Предельный поток существует при любом ресурсе W . Причем, при $W \leq T$ $f_{sum}^* = W$; при $W > T$ $f_{sum}^* = T$.

2.3.6. Для случая $W > T$ и любом начальном распределении ресурса предельное состояние Q^* существует.

2.3.7. При $W > T$ для некоторого множества вершин существуют начальные состояния, из которых эти вершины оказываются в зоне Z^{+*} . Такие вершины будем называть *потенциальными аттракторами*. Потенциальными аттракторами могут быть некоторые вершины-приемники, а также некоторые нейтральные вершины. Если в сети более одного потенциального аттрактора, ее предельное состояние не единственно и зависит от начального состояния.

2.3.8. Найден критерий аттрактивности вершин. Вершина является потенциальным аттрактором тогда и только тогда, когда при $W = T$ она в предельном состоянии имеет ресурс, равный своей выходной пропускной способности: $\tilde{q}_k = r_k^{out}$.

2.4. Симметричные и квазисимметричные сети

Сети, в которых для каждой вершины величины r_i^{in} (сумма пропускных способностей ребер, входящих в вершину v_i) и r_i^{out} (сумма пропускных способностей ребер, выходящих из вершины v_i) равны, делятся на два класса с похожими свойствами: симметричные сети (с симметричной

матрицей пропускной способности) и квазисимметричные сети, для которых матрица пропускной способности несимметрична, но для каждой вершины выполняется: $r_i^{in} = r_i^{out}$. В таких сетях (как и в однородных) все вершины являются потенциальными аттракторами.

2.4.1. В симметричных и квазисимметричных сетях пороговое значение T равно суммарной пропускной способности сети: $T = r_{sum}$

2.4.2. Для случая $W \leq T$ предельное состояние единственно и задается вектором:

$$Q^* = \left(\frac{r_1^{out}}{r_{sum}} W, \frac{r_2^{out}}{r_{sum}} W, \dots, \frac{r_n^{out}}{r_{sum}} W \right).$$

2.4.3. Для случая $W > T$ вершины, попавшие за конечное число тактов в зону Z , не смогут из нее выйти, а в предельном состоянии они имеют ресурс $r_i^{in} = r_i^{out}$:

$$Q^* = (r_1^{out} + c_1^*, \dots, r_m^{out} + c_m^*, r_{m+1}^{out}, \dots, r_n^{out}).$$

2.5. Динамика потоков в ресурсных сетях

Потоки в двусторонних сетях с петлями стабилизируются при любом количестве ресурса и его начальном распределении. Однако в зависимости от начального состояния динамика потока может быть различной. Суммарный поток может быть как монотонным, так и немонотонным, может возрастать, убывать, а также иметь промежутки временной стабилизации. При этом, в зависимости от начального распределения одного и того же суммарного ресурса в одной и той же сети, потоки могут быть различными (хотя предельный поток будет одинаковым).

На рисунках представлена динамика потока в несимметричной сети с пятью вершинами при разном начальном состоянии.

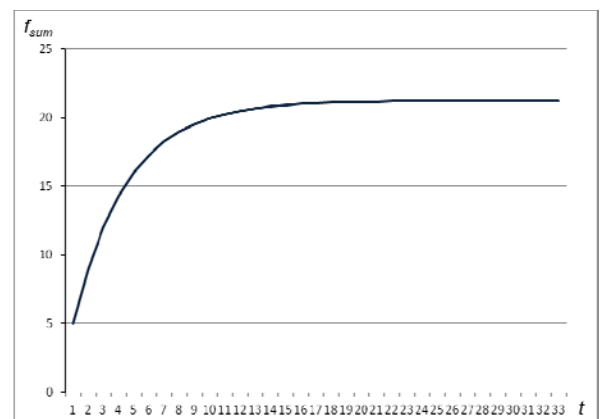


Рисунок 1 – Монотонно возрастающий поток

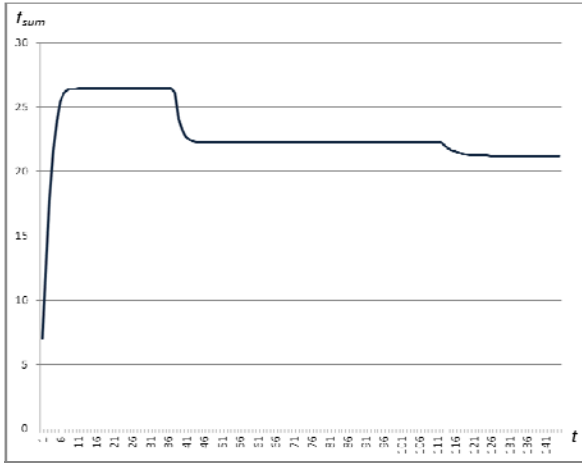


Рисунок 2 – Немонотонный поток с несколькими плато

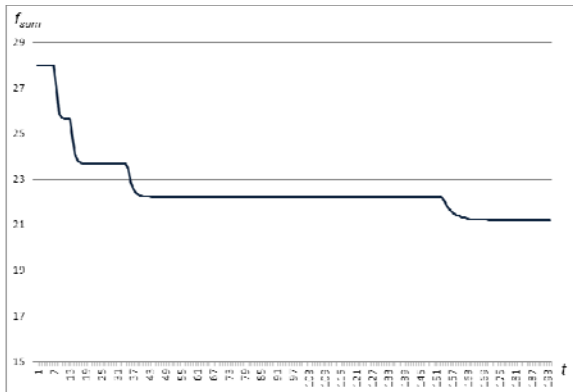


Рисунок 3 – Неубывающий поток с несколькими плато (в первые семь тактов $f_{sum} = r_{sum}$)

2.6. Динамика состояний в ресурсных сетях

Исследование изменений координат вектора предельных состояний выявило зависимость распределения ресурса от начального состояния и топологии сети. Чаще всего ресурсы в вершинах изменяются монотонно. Однако в некоторых случаях в сети возможны значительные колебания (рис. 4).

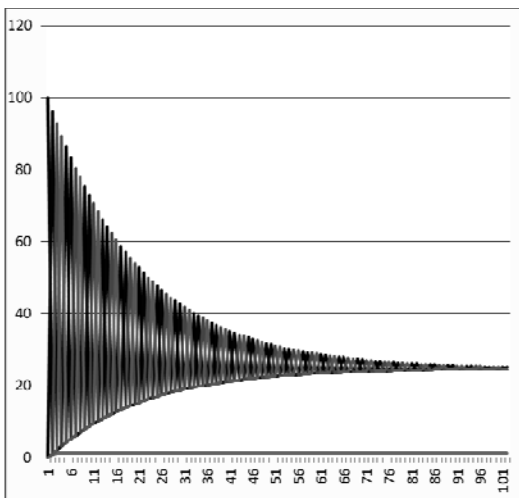


Рисунок 4 – Колебания и стабилизация ресурса в квазисимметричных сетях

Доказано (и видно из рисунков), что при любом изменении вектора предельных состояний имеет место асимптотическая сходимость к предельному состоянию.

3. Сравнительные характеристики двух моделей

Модель выстреливания фишек и ресурсная сеть имеют много общего. Общим является принцип функционирования: хранение ресурса в вершинах и распределение его по ребрам в смежные вершины. Однако имеется ряд существенных различий, которые определяют аппараты их исследования и предметные области, описываемые этими моделями.

Сходства и различия представим в виде сравнительной таблицы.

Таблица 1 – Характеристики модели «chip-firing game» и ресурсной сети

	Chip-firing game	Ресурсная сеть
Параллельность	–	+
Накопление ресурса до порогового значения	+	– пороговое значение определяет правило функционирования вершин
Распростр. малых ресурсов	–	+
Делимость ресурса	–	+
Асимптотич сходимость	– (либо остановка, либо бесконечный цикл)	+

Chip-firing game моделирует лавины за счет своей способности накапливать ресурс в узлах до момента выстреливания. В ресурсной сети вершины отдают ресурс, имея любое его количество. Таким образом, самоорганизующаяся критичность не может быть описана ресурсной сетью с набором правил, описанным в данной работе.

Ресурсная сеть подходит для моделирования иных процессов, в которых перераспределение должно происходить постоянно. На основе ресурсной сети построена модель распространения загрязняющих веществ в водной среде [Жилякова, 2011]. В этой модели сеть задана регулярной двумерной решеткой, наложенной на некоторую площадь водоема. Узлы хранят в себе количество вещества, ребра отвечают за перетоки между

смежными районами. Пропускные способности соответствуют полям течений, которые зависят от метеорологических и гидрологических параметров и считываются из базы данных. Пропускная способность петли отвечает за скорость оседания вещества.

Модели социальных процессов в сетях мобильных операторов, позволяющие предсказывать динамику оттока клиентов из данной сети (см. напр. [Dasgupta et al., 2008]) также можно описать в терминах ресурсных сетей. На основе данных о звонках и смс-сообщениях клиентов сети строится граф, узлы которого представляют собой телефонные номера, а двусторонние пары – обоюдные звонки. Веса ребер соответствуют количеству и продолжительности разговоров. Делается предположение, что решимость абонента сменить сотового оператора зависит от количества его друзей (смежных узлов), сменивших оператора за предыдущий период времени. В узлы, перешедшие в другую сеть, помещается некоторое количество ресурса, названное в данной модели «энергией». Энергия в соответствии с правилом 2 ресурсной сети распространяется по инцидентным ребрам в смежные узлы. После наступления стабилизации узлы, имеющие энергию, превосходящую некоторое пороговое значение, считаются потенциально готовыми к уходу из сети.

Третьей моделью, в основе которой лежит ресурсная сеть, является модель ассоциативной памяти [Жилиякова, 2010]. В ней узлы соответствуют понятиям, а двусторонние пары ребер – взаимным ассоциациям между ними.

ЗАКЛЮЧЕНИЕ

В работе описаны две модели рассеяния на графах: целочисленная последовательная пороговая модель chip-firing game и параллельная модель с бесконечно делимым ресурсом – неоднородная ресурсная сеть.

Рассмотрены свойства обеих моделей, описываемые ими предметные области и границы применимости каждой из них.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Жилиякова, 2010] Жилиякова Л.Ю. Реализация рекурсивных запросов в динамической ассоциативной ресурсной сети // Двенадцатая национальная конференция по искусственному интеллекту с международным участием КИИ'2010. Труды конференции, том 1. М. – Физматлит, 2010. с. 335-343.

[Жилиякова, 2011] Жилиякова Л.Ю. Применение ресурсных сетей для моделирования распространения веществ в водной среде // Проблемы управления, 2011, № 2, с. 46-51

[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., Жилиякова Л.Ю. Двусторонние ресурсные сети – новая потоковая модель // Доклады АН, 2010, том 433, №5, с.609-612.

[Форд и др., 1966] Форд Л.Р., Фалкерсон Д. Потоки в сетях. М.: Мир, 1966. 276 с.

[Aiello et al., 1993] W. Aiello, B. Awerbuch, B. Maggs and S. Rao, Approximate load balancing on dynamic and asynchronous

networks, Proc. 25th ACM Symp. of Theory of Computing (1993), 632–634.

[Ahuja et al., 1993] Ahuja R.K., Magnati T.L., Orlin J.B. Network Flows: Theory, Algorithms and Applications. Prentice Hall, New Jersey, 1993.

[Bak et al., 1988] P. Bak, C. Tang and K. Wiesenfeld, Self-organized criticality, Physical Review A 38 (1988), 364–374.

[Bjorner et al., 1991] A. Bjorner, L. Lovasz and P. Shor, Chip-firing games on graphs, Europ. J. Comb. 12 (1991), 283–291.

[Bjorner et al., 1992] A. Bjorner and L. Lovasz, Chip-firing games on directed graphs, J. Algebraic Combinatorics 1 (1992), 305–328.

[Dasgupta et al., 2008] K. Dasgupta, R. Singh, B. Viswanathan, D. Chakraborty, S. Mukherjee, A.A. Navati. Social Ties and their Relevance to Churn in Mobile Telecom Networks // Proceedings of the 11th international conference on Extending database technology EDBT'08: Advances in database technology ACM New York, NY, USA, 2008.

[Dhar, 1990] D. Dhar, Self-organized critical state of sandpile automaton models, Physical Review Letters 64 (1990), 1613–1616.

[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.

[Spencer, 1986] J. Spencer. Balancing vectors in the max norm. Combinatorica, v. 6 (1986), 55–66.

DISCRETE MODELS OF DIFFUSION ON GRAPHS

Zhilyakova L. Yu.

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Profsoyuznaya ul., 65, Moscow, 117997 Russia
zhilyakova.ludmila@gmail.com

Two models of diffusion on graphs are considered: chip-firing game – an integer threshold model, and nonhomogeneous resource network. Their main properties are described. The similarities and differences between these models are analysed, and their applications are specified.

INTRODUCTION

There exist a number of graph-theoretic models, involving various kinds of diffusion processes. These models include: random walks on graphs; the “avalanche” or “sandpile” model of catastrophic events, which is mathematically equivalent to “chip-firing” games; load balancing in distributed networks; and, the our object of our study: nonhomogeneous resource networks.

CONCLUSION

In this paper we compare the two models: chip-firing game and nonhomogeneous resource network. In spite of many common properties they need different methods of research and have rather different applications.



УДК 004.822:514

ТЕРМИНОЛОГИЧЕСКИЕ СЕТИ

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 анализируется понятие терминологической сети, обсуждаются их свойства и особенности, приводится типовая технология построения таких сетей. В качестве конкретной терминологической сети описывается универсальное терминологическое пространство, использующее ограниченное количество типов бинарных отношений. Обсуждаются вопросы практического применения терминологических сетей на примере веб-ресурса www.glossary.ru; рассматриваются пользовательские сервисы, основанные на использовании свойств терминологических сетей.

Ключевые слова: глоссарий, определение, семантическая сеть, термин.

ВВЕДЕНИЕ

Человек в своих рассуждениях неизбежно оперирует терминами. Свободное владение специальной терминологией – верный признак профессионализма. Терминологические системы проблемных областей отражают соответствующий понятийный аппарат и известные отношения между понятиями. Подавляющая часть определений терминов зафиксирована в толковых словарях, глоссариях, ГОСТах и энциклопедиях. Существует и развивается терминоведение [Лейчик, 2009] – отрасль филологии, изучающая закономерности существования терминологии. В настоящей работе рассматриваются прикладные вопросы систематизации терминов и их определений в интересах развития интеллектуальных информационных технологий.

1. Структурирование терминологии

Возьмем с полки любой толковый словарь, скажем, «Новейший справочник школьника по математике» [Якушева, 2007], и откроем его на 363 странице. Среди прочих терминов на «У» здесь приводится определение угла.

Угол – геометрическая фигура, состоящая из двух лучей с общим началом. (Здесь и далее в примерах сохранен стиль источника.)

Из приведенного определения образованный человеком легко извлекает три задействованные в нем термина: «Геометрическая фигура», «Луч» и «Начало луча». Определения первых двух терминов

обнаруживаются в том же словаре на страницах 48 и 169:

Геометрическая фигура – конечное или бесконечное множество точек.

Луч – часть прямой, состоящая из всех точек этой прямой, лежащих по одну сторону от данной точки.

И хотя отдельного определения для «Начала луча» в словаре не обнаружено, его можно достаточно просто восстановить из контекста упоминания термина «Луч».

Начало луча – точка на луче, относительно которой все остальные его точки расположены по одну сторону.

Результаты интеллектуальных усилий по анализу определения «Угол» можно представить в материальном виде: четыре картонные карточки с определениями, соединенные тремя наклейками-стикерами с именами бинарных отношений (см. рисунок 1).

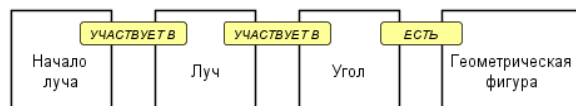


Рисунок 1 – Связи между определениями

В рассмотренном примере карточки-определения выстроились в линейную последовательность, однако так бывает не всегда. Существенно, что три определения «Луч», «Угол» и «Геометрическая фигура» имеют некоторое внутреннее устройство:

- «Луч» определяется с использованием «Начала луча»;
- «Угол» определяется с использованием «Луча»;
- «Геометрическая фигура» имеет подвид «Угол».

Наличие у определения внутреннего устройства свидетельствует о том, что соответствующий термин описывает некоторое понятие проблемной области. Фактически определение, обладающее внутренним устройством, задает типичного представителя [Нильсон, 1985] некоторого класса объектов. Наименование выявленного таким образом понятия получается переформулированием имени его типичного представителя. Как правило, переформулирование сводится к образованию множественного числа. Так, в примере из рисунка 1

- определению «Луч» соответствует понятие «Лучи»;
- определению «Угол» соответствует понятие «Углы»;
- определению «Геометрическая фигура» соответствует понятие «Геометрические фигуры».

Отметим, что в самых разных науках словосочетание «раскрыть понятие» подразумевает выявление подвидов этого понятия и описание отношений между ними с привлечением некоторого количества других концептов.

Понятия-классы и остальные определения можно рассматривать в качестве вершин семантической сети. Договоримся обозначать на рисунках понятия – овалами, а остальные определения – прямоугольниками. Набор бинарных отношений, связывающих определения, обязан содержать отношение *это-есть* (*есть*) для представления родо-видовых отношений; представительство остальных отношений определяется целью структурирования терминологии. С учетом принятых соглашений результат анализа определения «Луч» представляет собой фрагмент семантической сети, изображенный на рисунке 2.

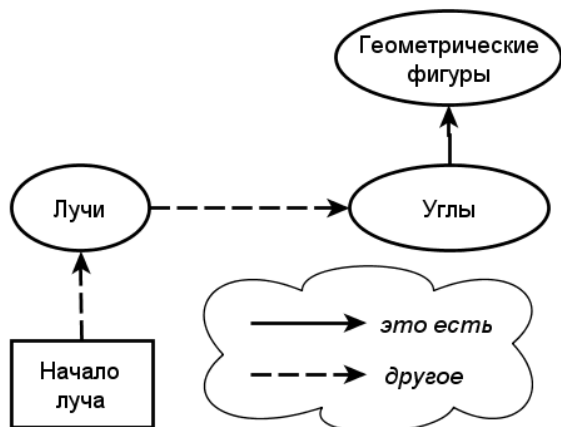


Рисунок 2 – Фрагмент семантической сети

Аналогичный анализ определения «Ломаная» порождает иной фрагмент семантической сети (рисунок 3), содержащий, в том числе, некоторое количество ранее встретившихся вершин-определений.

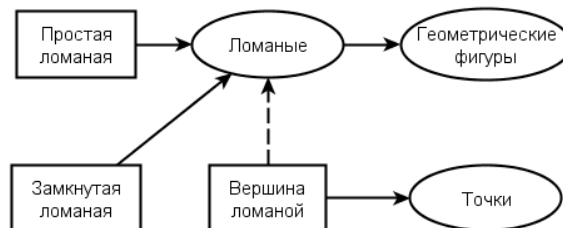


Рисунок 3 – Второй фрагмент семантической сети

Присоединение вновь построенного фрагмента (рисунок 3) к имеющейся сети (рисунок 2) порождает более сложную семантическую сеть, приведенную на рисунке 4. Идея операции присоединения состоит в том, что каждому определению в сети должна соответствовать ровно одна вершина.

Продолжая процесс построения новых фрагментов и присоединяя их к уже имеющейся сети, можно построить весьма сложную семантическую сеть, фиксирующую бинарные связи между понятиями и терминами проблемной области. Сети такого рода будем называть терминологическими сетями. Вершинам терминологической сети обязательно сопоставляются определения терминов; нарушаться это правило может только в исключительных случаях принципиального отсутствия определений. Для неопределяемых понятий в терминологической сети создается определение-заглушка.

Терминологическая сеть – семантическая сеть, вершинами которой являются определения терминов, связанные бинарными отношениями заданных типов.

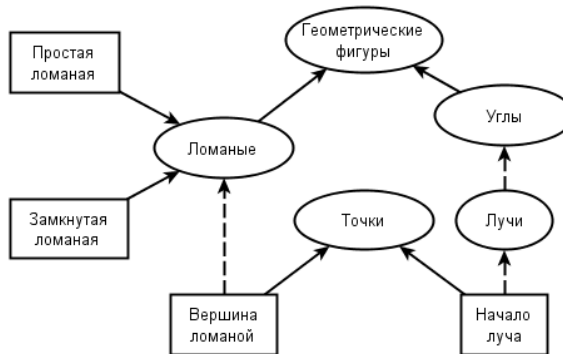


Рисунок 4 – Результат присоединения

Рассмотренная на конкретном примере технология систематизации определений, порождающая терминологическую сеть, относится к сфере интеллектуальной деятельности человека, который в этом контексте именуется научным редактором. Именно редактор в меру своей

квалификации и образования выявляет фрагменты в заданном массиве определений, а затем присоединяет их к ранее построенной сети. Вместе с тем, производительность труда научного редактора вполне поддается оценке и прогнозированию, что позволяет (хотя бы гипотетически) ставить вопрос об организации “терминологического бизнеса”.

Важно, что реальная деятельность по систематизации терминологии возможна только при наличии у научного редактора адекватного программного инструментария.

2. Универсальное терминологическое пространство

В 2000 году начались работы по реализации конкретного варианта терминологической сети. Проект “Универсальное терминологическое пространство” (УТП) [Мальковский и др., 2002] проект ориентирован на интеграцию в единой сети всех терминов научной и деловой лексики. Цель проекта состоит в организации самодостаточного процесса систематизации терминологии.

По состоянию на начало 2012 года УТП содержит 53'477 определений, 9'044 из которых являются понятиями. Суммарно в УТП представлены 95'252 термина и их синонима из астрономии, биологии, географии, информатики, математики, физики, физиологии, экологии, экономической теории, внешней торговли, страхового дела, управления и из десятков других отраслей науки и видов деятельности. Между определениями УТП установлены 75'569 бинарных связей двух типов:

- отношение *это-есть* (34'566 экземпляров.); и
- отношение *относится-к* (41'003 экземпляра), включающее в себя все типы отношений, отличающиеся от отношения *это-есть*.

Наличие только двух типов отношений объясняется необходимостью иметь технологичный программный инструмент, рассчитанный на реальные возможности научных редакторов. Отчасти это ограничение компенсируется специальными шаблонами в определениях терминов. Так, определение

Почвенный раствор – жидкая часть почвы; вода с растворенными газами, минеральными и органическими веществами.

в явном виде информирует пользователя, что связь *относится-к*, установленная между «Почвенный раствором» и «Почвой», на самом деле является связью типа *часть-целое*.

Каждый термин, представленный в УТП, может иметь несколько синонимов и аббревиатур. определение может содержать этимологическую справку, ссылку на некоторый веб-ресурс (URL) и указатель на файл-изображение. Определение термина задается в виде линейного текста, возможно разделенного на абзацы, и/или содержащего списки перечислений.

По построению в УТП все понятия имеют различные наименования, остальные термины могут иметь «двойников». Опыт существования УТП показывает, что такое ограничение не является существенным. Как оказалось, для каждого понятия без труда находится уникальное наименование, отличающее его от других понятий собственной и других проблемных областей.

Топология текущей версии УТП с указанием наименований понятий и терминов, приписанных вершинам, находится на странице

http://www.glossary.ru/_netwrk_.htm

и доступна в формализованном электронном виде всем исследователям.

Характерная для УТП ориентация на неограниченный круг терминов отделяет от общего класса терминологических сетей собственный подкласс универсальных терминологических сетей.

Универсальная терминологическая сеть – терминологическая сеть, предназначенная для систематизации неограниченного количества терминов.

Выдающимся, но не единственным примером универсальной терминологической сети являются клинические коды Дж. Рида [Емелин, 2000] RCC (Read Clinical Codes; Clinical Terms Version 3; CTV3). Разработка каждой универсальной сети начинается с постулирования некоторых ее свойств, которые, с одной стороны, направляют дальнейшее развитие сети, а с другой стороны – играют роль ограничителей.

3. Особенности универсальных терминологических сетей

Накопленный опыт ведения УТП позволяет говорить о некоторых принципиальных свойствах и проблемах универсальных терминологических сетей.

Проблемы редактирования | При работе с конкретной терминологией возникает большое количество ситуаций, в которых редактор вынужден принимать решения. В частности, серьезные затруднения вызывают:

- качество исходных определений;
- отсутствие определений;
- наличие альтернативных определений.

О качестве определений | Значительная часть исходных определений не выдерживает критики с точки зрения стиля и полноты; часто определения излагаются в виде адаптированного под интересы некоторой проблемной области.

Об альтернативных определениях | В некоторых науках мирно сосуществуют несколько несводимых определений одного и того же понятия: «Почва по В.В.Докучаеву», «Почва по П.А.Костычеву», «Почва по В.А.Вернадскому» и др. Каждый такой случай требует от научного редактора

индивидуального подхода для корректного представления альтернативных определений в терминологической сети.

О размере определений | Наилучшее по объему определение содержит 2-3 предложения. Наличие в определении пяти и более предложений сигнализирует о необходимости разделить текст определения между несколькими терминами.

О достоверности связей | При построении терминологической сети части терминологии, попадающей на стык проблемных областей, приходится структурировать многократно. Второе и последующие структурирования в значительной степени перепроверяют ранее проделанную работу. По результатам этих проверок можно с уверенностью утверждать, что количество ошибок в УТП мало.

О достаточности связей | В весьма редких случаях возникает необходимость представить в УТП бинарное симметричное отношений между терминами. Одно из решений этой задачи состоит в том, чтобы разместить оба термина в окрестности их родительского понятия.

Проблема недостаточности | В любом источнике терминологии всегда обнаруживается некоторое количество терминов, которые используются в определениях других терминов, но сами в этом источнике не определяются. Это обстоятельство делает неизбежным использование в процессе построения универсальной терминологической сети различных терминологических источников.

Связность | Универсальная терминологическая сеть, построенная без привлечения общенаучной, религиозной и философской терминологии, является связанной. В структуре такой сети явно выделяются отрасли науки и виды деятельности, играющие объединительную роль: математика, бухгалтерский учет, страховое дело и др.

Цикличность | В терминологической сети присутствуют циклы. Как правило, существование циклов связано с различиями в ролях, которые играют крупные, сложно устроенные понятия. Типичным примером такого понятия является «Государство», которое в разных контекстах выступает и продавцом, и покупателем и регулятором рынка.

Проблема отбора | Часть терминологии, находящейся в общечеловеческом обороте, на самом деле связана с недостаточно обоснованными гипотезами. Наличие такой терминологии в универсальной терминологической сети способно серьезно исказить картину мира.

Проблема старения | В терминоведении факт старения и выхода терминологии из оборота установлен и описан. Разработка адекватных программных и изобразительных средств,

автоматически поддерживающих процесс старения терминологии, составляет открытую проблему.

Проблема изменчивости | С течением времени отдельные термины («Ноосфера» и др.) претерпевают существенные изменения в своем содержании. Другими словами в разные периоды времени одному и том же термину отвечают различные понятия. При этом изменяются и связи термина. Выявление и учет этого обстоятельства составляют отдельную, открытую проблему терминологических сетей.

Проблема мироустройства | Конструирование терминологической сети, состоящее из операций порождения и присоединения новых фрагментов аналогично восходящей схеме проектирования программ [Йодан, 1979]. Проблема лишь в конфигурации «верхних этажей» сети. Можно, скажем, полагать первичность материи или исходить из существования Бога. Как это ни странно, но вопросы мироустройства необходимо зафиксировать в качестве инженерного решения.

Инварианты | Существуют несколько инфоматрических характеристик УТП, неизменных на всех этапах его существования. Эти характеристики не зависят от количества терминов в УТП и нуждаются в объяснении.

4. Веб-ресурс www.glossary.ru

Практическую востребованность концепции терминологических словарей подтверждает веб-ресурс Служба тематических толковых словарей – www.glossary.ru [Мальковский и др., 2002]. Информационное наполнение ресурса получается формальным преобразованием очередной версии УТП в структуры, удобные для построения веб-сайта. Веб-ресурс www.glossary.ru предоставляет пользователям определения терминов научной и деловой лексики, с группированные в стандартные, относительно компактные проблемно-ориентированные глоссарии.

Стандартный глоссарий определяется уникальным наименованием понятия и состоит из терминов, непосредственно связанных бинарными отношениями с соответствующей понятийной вершиной. По сути дела стандартный глоссарий содержит минимум терминологии, раскрывающий то или иное понятие.

При построении веб-ресурса существенно используется структура УТП для навигации в пространстве терминов, а также для разработки сервисов поддержки интеллектуальной деятельности. Интеллектуальные сервисы ориентированы на предоставление пользователю дополнительных возможностей по выявлению связей между терминами.

Основными интеллектуальными сервисами в проекте www.glossary.ru являются:

- сервис пополнения стандартных глоссариев;

- сервис тематической группировки;
- сервис визуализации;
- сервис формирования поисковых запросов;
- сервис веб-карт;
- сервис формирования словарей;
- сервис сортировки терминов.

Сервис пополнения позволяет включать в состав проблемно-ориентированного глоссария некоторые дополнительные термины. Например, стандартный глоссарий «Углы» не содержит определение «Начало луча», хотя этот термин и участвует в определении термина «Угол». За счет сервиса пополнения заинтересованный пользователь может получить более широкое множество терминов и их определений, включающее, в частности, и термин «Начало луча». Алгоритм пополнения [Мальковский и др., 2002] существенно использует родо-видовые связи и реализует механизм наследования свойств.

Сервис тематической группировки понятий конструирует последовательность окрестностей заданного понятия. Каждая окрестность представляет собой набор понятий, отстоящих от заданного, на известное расстояние. При группировке используются связи УТП без учета их типов. Сервис тематической группировки вскрывает «близкородственные», порой достаточно неожиданные связи между проблемными областями.

Сервис визуализации позволяет представить в наглядном виде фрагмент семантической сети, охватывающий терминологию отдельного стандартного глоссария и связанных с ним понятий [Соловьев, 2008a]. При построении графического образа существенно используется наличие в УТП ровно двух типов отношений.

Сервис интерактивного формирования поисковых запросов предоставляет пользователю многовариантную форму генерации обращений к ИПС в связи с некоторым конкретным термином. При построении формы используется содержащийся в УТП набор связей выбранного термина [Мальковский и др., 2004]. Отмеченные пользователем варианты транслируются сервисом в запрос на расширенном языке, а затем запрос передается системе Яндекс на исполнение.

Сервис веб-карт представляет пользователю веб-ресурса систему терминологии в обозримом укрупненном виде [Соловьев, 2008b], и тем самым обеспечивает быструю навигацию в терминологической сети на уровне отраслей и родов. Алгоритм построения веб-карт решает задачу наглядного представления сильно разреженного графа («почти дерева»).

Сервис формирования ассоциативных терминологических словарей позволяет представить в линейном виде достаточно большую совокупность родственных терминов. Алгоритм формирования преобразует семантические связи УТП в разделы и

порядок расположения терминов, а также в систему перекрестных ссылок.

Сервис сортировки терминов призван повысить «читабельность» глоссария. Существование сервиса объясняется феноменом обратной зависимости между объемом толкового словаря и привлекательностью для пользователя последовательного чтения определений. Установлено, что узкие специализированные глоссарии человек может читать почти как связный текст. Однако с ростом количества терминов и расширением проблемной области глоссарий теряет это свойство и превращается в хранилище разрозненных статей. Для иллюстрации этого феномена приведем отрывок из фундаментальной кукольной энциклопедии [Голдовский, 2004].

Ножная кукла – перчаточная кукла, которой актер управляет с помощью ног.

Ножной брусок – элемент управления марионетки, куда крепятся ноги куклы.

Нос – комический кукольный народный герой традиционного голландского театра кукол.

Носек – чешский художник-кукольник начала XX века.

Понятно, что в приведенном примере каждое следующее определение радикально изменяет контекст существования предыдущего определения, разрывая какие бы то ни было связи между соседними терминами.

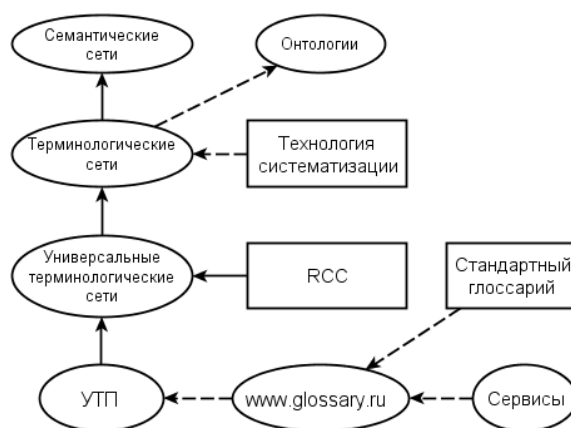


Рисунок 5 – Терминологическая сеть как способ представления знаний

Алфавитный принцип следования терминов противодействует образовательной функции толкового словаря. В обширных глоссариях [возникающих, например в результате пополнения] существует настоятельная необходимость в таком изменении порядка следования терминов, при котором они выстраиваются в взаимосвязанные цепочки. Вместе с тем возможный в УТП формальный переход к алфавитно-гнездовой системе расположения терминов оправдан далеко не всегда, поскольку имеющиеся семантические связи между терминами обеспечивают более высокий уровень группировки терминов. В некоторых случаях более удобными оказываются формально

построенные идеографические словари [Морковкин, 1970]. Исходя из этого, сервис сортировки терминов предлагает пользователю на выбор один из трех различных способов расположения терминов.

Приведенный перечень интеллектуальных сервисов, основанных на УТП, не является исчерпывающим. Значительный интерес для инженерии знаний представляет сервис генерации прототипов онтологий. Нерешенной остается задача применения УТП в информационном поиске вообще и в информационном поиске для проекта www.glossary.ru в частности.

ЗАКЛЮЧЕНИЕ

На рисунке 5 показано место терминологических сетей в системе способов представления знаний. Конечно, сети такого рода не позволяют представлять совершенно любые декларативные знания из-за ограничений на размерность отношений и из-за достаточно жесткой привязки к терминологии, однако терминологические сети существуют реально, они полезны конечным пользователям, а, кроме того, порождают и позволяют решать новые интересные задачи.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Голдовский, 2004] Голдовский, Б.П. Куклы: Энциклопедия / Б.П. Голдовский // – М.: Время, 2004. – 496 с.
- [Емелин, 2000] Емелин, И.В. Интеграция стандартов медицинской информации / И.В. Емельянов // Кремлевская медицина. Клинический вестник, No. 4.- 2000. С.55-62.
- [Йодан, 1979] Йодан, Э. Структурное проектирование и конструирование программ / Э. Йодан // – М.: Мир, 1979. – 415 с.
- [Лейчик, 2009] Лейчик, В.М. Терминоведение: Предмет, методы, структура / В.М. Лейчик // – М.: Книжный дом «ЛИБРОКОМ», 2009. – 256 с.
- [Мальковский и др., 2002] Мальковский, М.Г. Универсальное терминологическое пространство / М.Г. Мальковский, С.Ю. Соловьев // Диалог'2002. Труды международного семинара. Т 1, С. 266-270
- [Мальковский и др., 2003] Мальковский, М.Г. Методы формирования глоссариев в универсальном терминологическом пространстве. / М.Г. Мальковский, С.Ю. Соловьев // Труды международной конференции “Компьютерная лингвистика и интеллектуальные технологии” – М.: Наука, 2003, С.438-440.
- [Мальковский и др., 2004] Мальковский, М.Г. Структурный метод формирования запросов к информационной системе. / М.Г. Мальковский, С.Ю. Соловьев // Труды международной конференции “Компьютерная лингвистика и интеллектуальные технологии” – М.: Наука, 2004, С.612-613.
- [Мальковский и др., 2010] Мальковский, М.Г. Системы поддержки творческих процессов / М.Г. Мальковский, С.Ю. Соловьев, А.Н. Сотников // Программные продукты и инструменты, под ред. Л.Н.Королева – М.: ВМК МГУ; МАКС Пресс, 2010, No.10. С.74-85.
- [Морковкин, 1970] Морковкин, В.В. Идеографические словари / В.В.Морковкин // – М.: Изд-во Моск.ун-та, 1970. – 72 с.
- [Соловьев, 2008a] Соловьев, С.Ю. Образные представления терминологической сети / С.Ю. Соловьев // Сб. Прикладное программное обеспечение – М.: МИРЭА, - 2008. С.55-69.
- [Соловьев., 2008b] Соловьев, С.Ю. Об одном методе генерации страниц-карт для веб-сайтов. / С.Ю. Соловьев А.Н. // Информационные процессы Том 8, No.1, 2008, С.24-39.
- [Нильсон, 1985] Нильсон, Н. Принципы искусственного интеллекта / Н. Нильсон // – М.: Радио и связь, 1985. – 373 с.
- [Якушева, 2007] Якушева, Г.М. Математика. Новейший справочник школьника / Г.М. Якушева // – М.: СЛОВО, Эксмо, 2007. – 479 с.

TERMINOLOGICAL NETWORKS

Malkovsky M.G., Soloviev S.Y.

Lomonosov MSU CS department, Moscow, Russia

malk@cs.msu.su

soloviev@glossary.ru

In the work the notion of terminological network and provides the technology for constructing such networks.

INTRODUCTION

In this paper we consider applied problems of systematization of terms for the development of intelligent information technologies.

MAIN PART

Technology systematization of terminology describes a specific example of geometric terms. The technology is based process of generating fragments of the network and the process of joining a fragment to its existing network. Semantic network with binary relations between the definitions of the terms is a result of the systematization. Such networks are called terminological networks. Terminological networks are a subclass of semantic networks.

Building a terminology network is a kind of intellectual activity. On the one hand, the work of the editor to build a terminological network is difficult. On the other hand the productivity of the editor be accurately measured. These circumstances make it possible to raise the issue of constructing a terminological network as a real business.

Software tools play a key role in building a terminological networks. Availability of software tools and standards for the construction of a terminological network impose constraints on the class of terminological networks. The "universal terminological space" is a successful example of building a large network terminology. Practical application of this project is a web-resource www.glossary.ru.

Terminological network allows us to construct non-trivial embedded application for users. Examples of intelligent services are services:

- the replenishment service standard glossary,
- the thematic grouping service,
- the service rendering service form search queries,
- the service web-card,
- the service forming vocabularies,
- the service sort of terms.

CONCLUSION

Terminological networks do not allow to represent any declarative knowledge of the restriction on the dimension of the relationship and because of the rather restricted to specific terminology, but terminology networks exist in reality and allow us to solve interesting problems.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.81:82:032.26

АВТОМАТИЧЕСКОЕ ПОСТРОЕНИЕ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ПРОИЗВОЛЬНЫХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ НА БАЗЕ НЕЙРОСЕМАНТИЧЕСКОГО ПОДХОДА

Бодякин В.И.

Институт проблем управления им. В.А. Трапезникова РАН, Москва

body@ipu.ru

Рассматривается актуальность комплекса проблем, связанных с обработкой сверхбольших объемов неструктурированной информации произвольных предметных областей и автоматического построения их семантических моделей. Анализируются компоненты: нейроподобный N-элемент, иерархически-сетевая структура из N-элементов, а также базовые свойства нейросемантического подхода – семантическая автоструктуризация, которая достигается при минимизации ресурсных затрат в процессе отображения предметной области на структуру N-элементов.

Ключевые слова: автоструктуризация, физический процесс, образ, информационно-управляющие системы, нейроподобные среды.

ВВЕДЕНИЕ

Актуальность комплекса проблем, связанных с обработкой сверхбольших объемов неструктурированной информации, постоянно нарастает. Современный этап развития научно-технической цивилизации характеризуется небывалым инновационным ускорением по всему спектру жизнедеятельности человека. Более половины открытий, научных теорий, технических и технологических новшеств из многотысячелетней истории цивилизации приходится на последние ее тридцать лет. В течение одного поколения мы удвоили цивилизационный потенциал и, одновременно, человечество не располагает средствами, чтобы оценить все возможные последствия нахлынувших на него инноваций.

Надежды на успешное применение методов искусственного интеллекта (ИИ) и нейронных сетей (НС) для автоматизации построения крупномасштабных информационно-управляющих систем (ИУС) принципиальным успехом не увенчались. Проектируемые сегодня ИУС качественно не отличаются от ИУС 60-х годов. Различие выражается лишь в быстродействии и информационных объемах, что является следствием технического развития электронной элементной базы. Тогда как, без качественного рывка, современные ИУС «захлебнутся» от экспоненциально нарастающих информационных потоков. Необходима ревизия, а также анализ методов и подходов в ИИ и НС, становление которых произошло более полувека назад, на базе экспериментальных данных психологии и нейрофизиологии того времени [Глушков, 1981, Бир, 1993].

Исследование и анализ когнитивных функций чрезвычайно сложно, в силу большого морфологического и функционального многообразия «мозговых» структур, в которых, одних только синаптических связей более триллиона. Поэтому, предлагается от исследования отдельного нейрона и мозга перейти к «теоретической когнитологии», когда выдвигается и моделируется на компьютере некоторая модель когнитивных процессов, результаты которой сравниваются со свойствами прототипа. По такому пути уже давно идут теоретическая физика, химия, фармакология, биология (благодаря генетике) и другие бурно развивающиеся области знаний.

Наша группа, идя по тому же пути «теоретизации», пришла к выводу, что создание глобальной ИУС для управления социально-экономическими и технологическими процессами в обществе и на производстве, характеризующимися все нарастающей сложностью, возможно только на пути реализации механизмов ее самообучения, функционально не уступающих механизмам человеческой психики.

Для этого мы разработали принципиально новую формальную модель нейроподобного N-элемента, структуру нейросети (НСС) из N-элементов и рекурсивную архитектуру ИУС из сформированных НСС.

Это направление мы назвали «нейросемантикой» [Бодякин, 2006]. Обсуждение со специалистами нейрофизиологами показало большую корреляцию современных нейрофизиологических данных, с характеристиками предлагаемой нами теоретической НСС-модели, по сравнению с характеристиками «классических» НС [Бодякин, 2006, 1998, 2009] [Грекул, 2008].

Кратко перечислим базовые утверждения нейросемантики:

1. Сложность и устойчивость используемых природой процессов при эволюционном конструировании homo sapiens укладывается в набор арифметических и графовых операций, т.е. «природа не мудрее четвероклассника».
2. При минимизации ресурсных затрат (памяти) на отображение в нейрореподобной специализированной среде (НСС) причинно-следственных процессов исследуемой предметной области (ПО) в этой нейрореподобной среде автоматически формируется словарь образов-процессов этой предметной области.
3. Архитектура ИУС состоит из трех НСС и объединяет в триплет образы: <ситуация ПО> <реакция ИУС> <оценка>, что позволяет отображать условные рефлексы и, посредством (само)обучения, формировать из них прагматическую модель поведения ИУС в произвольной предметной области.
4. Если рассматривать информационные процессы в самой НСС с помощью другой ортогональной (рекурсивной) ей НСС, как некую новую виртуальную предметную область, то оказывается, что в ней можно найти физические процессы, эквивалентные различным абстрактным понятиям, порождаемым человеческой психикой, что в результате позволит ИУС перейти к оперированию уже множествами (классами) образов.

1. Нейросемантика

Нейросемантический подход, это принципиально новый, эволюционный взгляд на (само)формирование и (само)функционирование ИУС. Прежде всего, это непосредственная («генетическая») связь ИУС со свойствами, изначально ее порождающей ПО. Это, отражение всех основных характеристик физических процессов ПО в ИУС, как предтечу ее когнитивных функций. Свойства ИУС вторичны по отношению к физическим процессам ПО, т.к. ИУС формируется в самой ПО на уже устоявшихся ее свойствах, см. рис. 1.

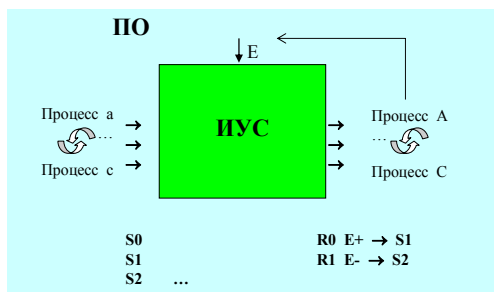


Рис. 1. Взаимоотношение ПО и ИУС

Отметим, что для физических процессов ПО характерны: детерминизм, целостность и причинно-следственность, иерархичность (фрактальность), энергетическая многомерность, параметрическая

количественно-качественная устойчивость, стремление к минимуму потенциальной энергии, функциональная эквивалентность аналоговой и дискретной форм представления и др.

Базовыми понятиями нейросемантического подхода являются:

- процесс;
- образ ИУС;
- взаимоотображение физического процесса и образа ИУС;
- текстовая форма (ТФ), как универсальный внутренний код ИУС и связующая форма взаимодействия ИУС и ПО.

Любой процесс рассматривается как некая целостность детерминированного изменения во времени вектора каких-либо физических параметров (характеристик). Процесс объединяет в себе как квази-устойчивую компоненту – параметры (данные), так и квази-динамическую компоненту – причинно-следственные детерминированные изменения параметров (функции). Такая модель процесса очень близка структуре объектно-ориентированных данных [Грекул, 2008].

Процесс также рассматривается нами как некая материальная субстанция, объединяющая в себе две разноплановые проекции: категорию вещество, обычно ассоциируемое с понятием – объект и категорию энергия, ассоциируемое с понятием – движение.

Объединение во вводимом понятии – «процесс», двух, воспринимаемых на сегодняшний день как разносубстанционных категорий, существенно упрощает концептуальную схему ИУС, т.к. пропадает необходимость в их обязательной «ручной» увязке для целостного функционирования в некой системе (напр., технической).

Любой общепринятый объект ПО, на более длительном временном интервале его наблюдения, всегда сводится к процессу. Например, придорожный камень на интервале миллионов лет, или человек от рождения и до его смерти – это все процессы, хотя каждый со своим темпом времени.

Все процессы переводимы в текстовую форму (ТФ), т.е. непрерывное, аналоговое изменение во времени компонент вектора физических параметров любого процесса может быть дискретизировано, или оцифровано («отекстовано») с любой требуемой степенью точности, как для отображения в форме образа, так и для последующего восстановления исходного физического процесса. Элементарные процессы отображаются знаками (символами) некоторого алфавита ТФ (например, A: <abazcw>).

Вообще, непрерывная и дискретная формы представления процесса (сигнала) принципиально различны. Непрерывная точна (до микроуровня квантовых процессов), но подвержена энтропии, т.е. деградации со временем. Напротив, дискретная форма сигнала точна лишь в на уровне шага дискретизации, но зато устойчива к действию энтропии в пределах этого же шага дискретизации.

Допустимость дискретизации физических процессов в ТФ следует из принципа

параметрической локальной однородности ПО (сведение континуума квантовых значений произвольного физического параметра к ограниченному числу качественно-различных образов. Например, вся температурная шкала от $0,03^\circ \text{K}$ до $10\,000,00^\circ \text{K}$ (10^6 значений), для любого вещества разбивается всего на четыре класса образов его состояния: «твердое», «жидкое», «газообразное», «плазма».

Из этого же принципа следует и возможность адаптивной подстройки в ИУС шага и равномерности дискретизации. В наиболее значимом для ИУС отрезке шкалы некоторого параметра происходит переход к более детальной дискретизации, тогда как на менее значимом интервале происходит закругление шага дискретизации. Тем самым происходит оптимизация начального представления ИУС информационного ресурса.

ТФ является универсальной формой для представимости разноуровневых процессов. Например, взаимодействия элементарных частиц – $\langle ababcw \rangle$ (микроуровень); процессов макроуровня – $\langle \langle \text{Привет, как дела?} \rangle - \langle \text{Нормально!} \rangle \rangle$; и образования двойной звезды из двух астрообъектов – $\langle zzbbaabj \rangle$ (на гигауровне).

Таким образом, из вышеизложенного следует, что понятие процесс является универсально-базовым и самодостаточным в нейросемантическом подходе.

ТФ для ИУС является источником ее информационного ресурса. Компонентами информационного ресурса являются: сигнал, образ, информация и знание. Сигнал – произвольная цепочка знаков ТФ. Образ – определенная знаковая последовательность ТФ, тождественная целому числу причинно-связанных процессов.

1.1. Нейроподобный N-элемент

Базовым функциональным элементом нейроподобной специализированной среды (НСС) является конечный автомат с памятью – N-элемент. Нейроподобный N-элемент имеет – k упорядоченных входов и один условно-рефлекторный выход и несколько рекурсивных выходов, отражающих его внутреннее состояние. N-элемент осуществляет функцию идентификации собственного образа с сигналами в потоке из предметной области (ПО), по некоторому активированному функционалу сходства. В случае положительной идентификации собственного образа в потоке сигнала, N-элемент переходит в активное состояние, подавая на условно-рефлекторный выход $U(t)=1$, в противном случае открывается доступ к рекурсивным выходам, отражающим его внутреннее состояние. Собственный образ N-элемента определяется в момент его формирования из неидентифицированного ни одним из уже существующих N-элементов символического участка информационного потока из ПО. Таким образом,

данный механизм формирования слоя N-элементов позволяет отображать в НСС весь поток из ПО без пропусков (без потерь) и, в тоже время, избегать дублирования среди собственных образов N-элементов, что дает возможность строить ассоциативную память.

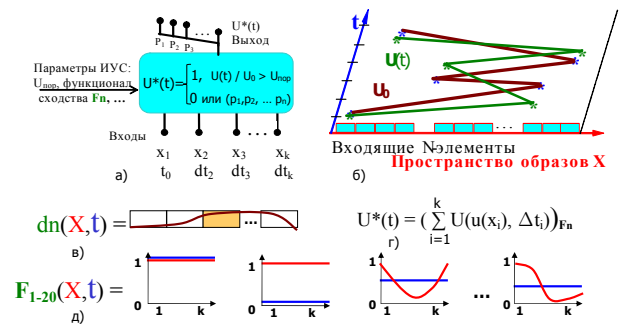


Рис. 1. а) Схематический вид N-элемента. Переход в активное состояние $U^*(t)=1$, при $F_i(U(t), U_0) > U_{пор}$, в остальных случаях доступны параметры внутренних состояний: P_1, P_2, \dots, P_n . б) Пространственно-временной образ N-элемента – U_0 и текущий входной сигнал – $U(t)$. в) Динамика проводимости входов в процессе обработки сигнала. г) Вычисление (свертка) $U^*(t)$ в процессе обработки сигнала. д) Пример набора функционалов сходства, отображающих максимальную проводимость с 1-го по k входы для X и t.

N-элемент принципиально отличается от общепринятого классического формального нейрона Маккалока и Питтса, разработанного ими в 1943 году [Бодякин, 2006]. Математический нейрон Маккалока и Питтса обычно представляют как некоторую пороговую или нелинейную функцию (функцией активации) от алгебраической суммы всех входных сигналов. Если суммы превышает некоторый параметрический порог, то на единственный выход нейрона подается «1», в остальных случаях – «0».

В N-эlemente происходит постоянное сравнение его пространственно-временного образа, сформированного в момент его формирования в сети, с текущим входным пространственно-временным сигналом. Если в результате вычисления сходства по $U^*(t) = (\sum_{k=1}^k U(u(x_i), \Delta t_i)) F_n$, при текущем функционале F_n , $U^*(t) > U_{пор}$, то на выходе N-элемента будет «1», а в остальных случаях «0». Но при этом, в случае «0» становятся доступны текущие значения «микровыходов» (параметров внутреннего состояния N-элемента) – P_1, P_2, \dots, P_n , для других N-элементов специализированной рекурсивной «ортогональной» нейросети (НСС_{1...5}).

Таким образом, функционально N-элемент становится включенным в два информационно-вычислительных процесса нейросемантической ИУС. Если на выходе будет «1», то N-элемент включается в простейший условно-рефлекторный

контур ИУС. При этом, N-элементы ИУС обрабатывают только образы, отражающие процессы внешней среды. Т.е. в этом случае, вся семантика вычислительных процессов ИУС структурно разлагается на совокупность признаков («язык») процессов предметной области и, например, программисту при написании какой-либо программы достаточно использования данной совокупности признаков.

Если же на выходе не будет ожидаемой «1», то N-элемент включается уже в рекурсивный контур ИУС. При этом, N-элементы «ортогональной» нейросети (НСС_{1...5}) обрабатывают помимо образов, отражающих процессы внешней среды еще и образы внутренних состояний N-элементов (P_1, P_2, \dots, P_n) условно-рефлекторного контура, которые являются следствиями внутренних процессов в N-лементах при первичной обработке внешней среды. В этом случае, N-элементы обрабатывают в том числе и состояния самих N-элементов.

Такая своеобразная рекурсивная обработка N-элементов существенно (эмерджентно) расширяет пространство признаков в ИУС о процессах в предметной области. Так, ни в одной реальной (физической) предметной области не существует процесса (признаков) характеризующих абстрактные понятия, например, число «два». Хотя могут быть «два яблока», «два кубика», но «два» в физической природе нет. И ни один программист (без использования нейросемантического подхода) не может написать программу, которая бы узнавала два совершенно новых процесса (объекта). Для решения поставленной задачи программисту нужно знать характерные признаки этих, априорно неизвестных процессов. Кстати, высшие биологические ИУС, в частности, человек, легко справляются с решением этой задачи.

Легко с этой задачей справляется и нейросемантическая модель рекурсивной ИУС. Благодаря автоструктуризации, происходит взаимоотображение структуры физических процессов предметной области (внешней среды) и структуры нейросемантических образов в ИУС [Бодякин, 2011]. Каждому новому процессу в нейроподобной среде выделяется новый N-элемент, в том числе и для априорно неизвестного процесса (объекта).

Конструктивно N-элементы устроены так, что внутренний параметр (P_1) величины выходного сигнала после его активации ($U^*(t)=1$) постепенно спадает по экспоненте. Если этот же N-элемент активируется повторно, то величина P_1 будет уже больше «1» и например, параметр «температура» поднимется до $P_1 = 1.5$ [Бодякин, 2011]. Значение этого параметра и будет характерным признаком состояния N-элементов для случаев графических изображений: «два яблока» и «два кубика».

Если же теперь на вход нейросемантической ИУС поступит пара совершенно новых для нее графических объектов-процессов, например, «два

ёжика», то контур условно-рефлекторной модели ИУС здесь «промолчит» ($U^*(t)=0$). Не получив реакции, эта задача автоматически поднимется на рекурсивный уровень контур обработки в НСС_{1...5}, где будет найдена ассоциация, что у новообразованного N-элемента сигнал $P_1 = 1.5$, так же как и у тех N-элементов, которые получили положительную оценку, выдав реакцию «два» на парные графические образы-процессы, и эта их ассоциация активируется. Таким образом, в ИУС автоматически сформируется новый абстрактный класс – «два», который объединяет все парные графические образы-процессы с внутренним параметром N-элемента «температура» $P_1 = 1.5$ и теперь, любая пара совершенно новых процессов будет определяться ИУС как – «два».

Этот пример показывает, как отсутствие рефлекторной реакции на N-элементах ($U^*(t)=0$) у нейросемантической ИУС включает ее механизмы «подсознания», для поиска новых ассоциаций уже в расширенном внутренними параметрами N-элементов (P_1, P_2, \dots, P_n) пространстве признаков. В этом расширенном пространстве признаков на параметрах P_1, P_2, \dots, P_n появляется возможность выявления таких принципиально новых признаков, как различные «отношения» между реальными пространственно-временными процессами предметной области. Параметры P_1, P_2, \dots, P_n N-элементов полностью задаются конструктором или природой.

Из вышеизложенного можно выдвинуть гипотезу, что для любого когнитивного отношения всегда можно сконструировать такой физический процесс на N-элементах, который будет отражать требуемое отношение-абстракцию. Например, абстрактное понятие «число» может отражаться в виде параметра некой обобщенной «температуры» образа, вызванного неоднократной активацией его N-элемента.

Также возможно формирование и других базисных образов-абстракций для ИУС: больше-меньше; фон; симметрии; повороты на угол; обороты; верх-низ-лево-право; большой-маленький; легкий-тяжелый; образы естественного языка: прилагательные, наречия, глаголы, ... и т.д. В человеческой культуре известно всего несколько десятков тысяч образов-абстракций, что соответствует мощности естественных языков.

Таким образом, снимая рабочие процессы в N-элементах многочисленными датчиками и структурируя их с помощью НСС, мы можем объективно определять формируемые абстрактные понятия.

Абстрактные понятия-образы позволяют сгладить-заполнить признаковое пространство между конкретными примерами, сразу покрывая весь класс (множество элементов) данного отношения.

Из этого факта и модели нейросемантической рекурсивной ИУС следует гипотеза, что

нейроморфология мозга высших животных должна эволюционно изменяться относительным нарастанием числа межнейронных контактов типа: «синапс – тело нейрона», к «синапс – аксон».

1.2. Нейросемантическая структура

Еще одна отличительная черта нейросемантического подхода заключается в том, что если в классических нейронных сетях изначально устанавливаются связи между нейронами («каждый с каждым») со случайными весами и затем в процессе обучения (тренинга) подстраивают веса этих связей, таким образом, чтобы вся нейросеть на входной вектор сигнала выдавала требуемый вектор реакции, то в нейросемантической ИУС процессы запоминания входного информационного потока и обучения объединены в единый процесс.

Все N-элементы объединены в слои и образуют многоуровневый иерархически-сетевой граф, который будем называть – нейросемантической структурой (НСС). Вершины графа (N-элементы) представляют образы отображаемых физических процессов текущей задачи и предметной области, а порядок входящих в вершину дуг отражает причинную связанность этих процессов. Межслойные связи в НСС формируются в режиме реального времени восходящей активностью ($U(t)=1$) N-элементов, вызванной обработкой ими символического потока из предметной области. Число слоев сформированной НСС определяется как $\log_k(TФ)$ от объема обрабатываемого потока. Размер каждого слоя НСС определяется разнообразием образов данного иерархического уровня, отображающих процессы в предметной области, см. рис. 3.

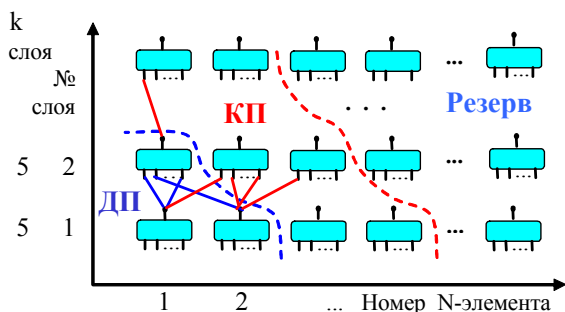


Рис. 3. Динамика роста нейросемантической структуры.

Процесс заполнения связями (памяти) НСС идет слева направо и снизу вверх (долговременная память – КП, промежуточная память – КП и резерв). В слое N-элементы вычислительно независимы, следовательно, возможна максимальная распараллеливаемость их обработки на ЭВМ.

Автоструктуризация (взаимоотображение физического процесса и образа в ИУС). Одним из значимых свойств НСС, является возможность автоматического формирования в ней графа, гомоморфного причинно-следственной структуре отображаемых физических процессов задачи для любой предметной области. Для этого необходимо

лишь минимизировать суммарные ресурсные затраты памяти, расходуемые на формирование вершин (N-элементов) и дуг (связей) в НСС при отображении в ней неограниченного символического информационного потока ТФ из произвольной ПО [Бодякин, 2006].

Если

$$\lim_{t \rightarrow \infty} \frac{\text{Объем НСС (в битах)}}{\text{Объем ТФ (в битах)}} \rightarrow 0, \quad (1)$$

то НСС будет гомоморфна (2)

структуре процессов ПО и будет представлять собой информационную (семантическую) модель данной ПО. Семантическая модель ПО – это совокупность ее причинно-связанных процессов.

Можно предложить множество несложных вычислительных процедур по минимизации ресурсных затрат НСС на базе вариационных методов, например, метода «частотной конкуренции образов словаря».

Рассмотрим данный метод на упрощенном примере. Допустим, что ПО состоит из трех «объектов» – трех шаров: «чугунного», «алюминиевого» и «деревянного». При попарных и тройном соударениях они порождают отекстованные образы четырех (n) процессов: «МАК», «ЛЕН», «РИС», «ДУБ», которые с равной вероятностью ($1/n$) без перекрытий, пауз и каких-либо разделителей поступают в поток ТФ, см. рис. 4. Задача ИУС заключается в том чтобы, анализируя только данный текстовый поток, построить информационную модель ПО. Для начала ИУС следует построить словарь «образов-процессов», в котором для упрощения демонстрации предположим, что все формируемые образы словаря имеют длину, соответствующую длинам процессов.

Метод «частотной конкуренции образов» базируется на частотном различии активации образов словаря принадлежащих двум множествам: «образов-процессов» и «образов-кентавров». «Образы-процессы» состоят из последовательности символов одного причинно-связанного процесса ПО, тогда как «образы-кентавры» состоят из двух причинно-связанных процессов: отображающих последовательность символов окончания одного процесса и последовательность символов начала следующего процесса, см. рис. 4.



Образ-процесс $p_p = 1/n$. $\omega_p = 1/p_p$
 Образ-кентавр $(1/n * 1/n)$ $p_k = 1/n^2$ $\omega_k = 1/p_k$ $\omega_p/\omega_k = n$
 Число образов-кентавров / число образов-процессов = $(L-1)*n$

Рис. 4. Образы-процессы и образы-кентавры при формировании словаря.

Из рисунка 4 легко понять, что образ-процесс имеет частоту активации (w_p), соответствующую частотам процессов ($1/n$) на интервале ТФ в 12 символов, т.е. $w_p = 1/n$. Соответственно, так как образ-кентавр построен из двух процессов ПО, то его частота активации (w_k) на интервале ТФ в 12 символов, определяется произведением вероятностей составляющих его процессов ($1/n * 1/n$), т.е. в нашем случае, $w_k = 1/n^2$.

Этот частотный признак и служит для предварительной маркировки формируемых N-элементов слоя как образов словаря в промежуточной памяти ИУС, см. рис.3. Большая частота активации N-элемента соответствует большей вероятности, что его образ принадлежит к классу образов-процессов. В итерационном алгоритме формирования словаря-слоя, N-элементам с большей частотой активации дается приоритетное право закрашивать ТФ и они постепенно вытесняют N-элементы с образами-кентаврами из формируемого словаря.

В соответствии (1), еще одной индикативной характеристикой при формировании информационной модели, гомоморфной причинно-следственной структуре процессов ПО, является размер формируемого словаря. Слои-словарь НСС, построенный только из образов-кентавров будет, как минимум, в n-раз больше слоя-словаря образов-процессов (3). Что при больших n алгоритмически (программно) легко определяется и служит управляющим фактором для корректного расширения словаря новыми образами.

$$\frac{\text{Размеры случайных словарей}}{\text{Размеры минимальных словарей}} > n \quad (3)$$

Таким образом, только при совпадении начал фаз образов словаря НСС и процессов ПО (см. рис. 4) и соответствии длин образов словаря и длин процессов получается минимальный словарь образов-процессов, который полностью соответствует исходным причинно-следственным процессам ПО и мы автоматически получаем информационную модель произвольной, априорно неизвестной ПО.

Формирование информационных моделей иерархических ПО. Очевидна справедливость механизма минимизации при формировании информационной модели ИИС и для иерархически сложных процессов ПО, см. рис. 5. В этом случае, сначала формируется словарь образов слов («Коля», «Саша», «ловит», ...), по методам рассмотренным выше. Затем, вместо образов словаря («Коля», «Саша», «ловит», ...) в ТФ ставятся их коды (номера) в словаре 1-го уровня НСС. В результате, мы получаем ТФ индексов и сводим данную задачу к предыдущей.

Проводим процедуру минимизации словаря уже 2-го уровня с кодами-индексами образов 1-го уровня словаря («134256 ...») и получаем двухуровневую структуру словаря («134», «256», ...), топологически изоморфную структуре

исходных процессов ПО. На втором уровне словаря будут сформированы образы псевдо-фраз, эквивалентные фразам-процессам ПО, см. пример на рис. 5.

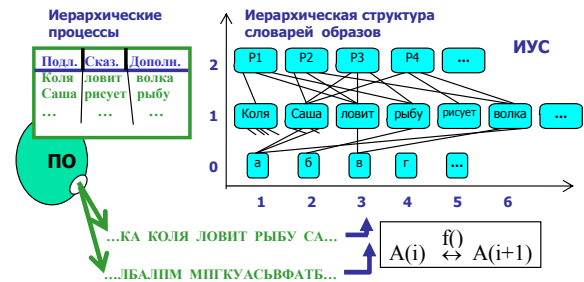


Рис. 5. Формирование иерархических словарей НСС.

На рис. 5 показано, что для взаимоднозначных алфавитов различных ТФ, порожденных одним процессом, информационные модели получаются топологически тождественными. Из этого факта следует, что у различных ИУС может быть сформирована объективная картина мира, независимо от алфавита и даже языка которые они используют.

Процесс формирования структуры словарей ИУС, гомоморфной структуре процессов ПО, мы назвали – автоструктуризацией. Механизм автоструктуризации достаточно много, от простейших вариационных методов до рассмотренного нами частотного метода [Бодякин, 2006, 2008, 2009]. Саму процедуру мы назвали – нейросемантической потому как благодаря автоструктуризации в каждом отдельном нейроразличном N-элементе формируется («стягивается») образ семантики ПО.

1.3. Автоструктуризация

Явление автоструктуризации, как взаимотождественное отображение структуры физических процессов ПО и структуры образов в НСС, первоначально было выявлено при моделировании на ЭВМ. Затем теоретически доказано, что для автоматического формирования в НСС графа, гомоморфного причинно-следственной структуре отображаемых физических процессов задачи любой предметной области, необходимо лишь минимизировать суммарные ресурсные затраты памяти, расходуемые на формирование вершин (N-элементов) и дуг (связей) в НСС, при отображении в ней символического информационного потока из произвольной ПО, см. рис. 4 и рис. 5.

1.4. Эволюционный потенциал ИУС

Известно, что для всех живых организмов энерго-вещественная и прогностическая составляющая их поведения, являются определяющими в характеристике их жизнеспособности. Отсутствие или недостаточность энергии, через некоторое время, приводит к гибели организма. Также практически нежизнеспособен биологический организм, если он

не может предвидеть развитие событий в окружающей его среде. Т.е. энергетическая и информационная составляющие, в своей совокупности, характеризуют потенциальную жизнеспособность организма. Объединение энергетической и информационной составляющих в некоторую комплексную характеристику назовем эволюционным потенциалом. Материальной компонентой эволюционного потенциала является энергия и вещество (Э), измеряемые в джоулях и нематериальная, т.е. информационная компонента (И), измеряемая в битах или образах, см. рис 6.

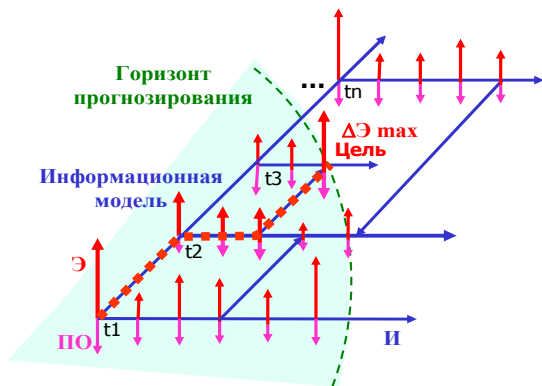


Рис. 6. Задача ИУС – максимизация материальной компоненты (Э) эволюционного потенциала до горизонта прогнозирования (И) информационной модели.

Если первая компонента эволюционного потенциала – это свободная энергия (Э), как необходимый ресурс для преобразования окружающей среды, то вторая компонента – нематериальная (И), содержит в себе информацию или знания о том, как в окружающей среде можно добыть и использовать эту свободную энергию, см. рис. 6.

Эволюционный потенциал характеризует жизнеспособность ИУС, как биологических организмов, так и технических систем. Если в контекстных условиях среды происходит столкновение интересов двух и более ИУС, то победителем выходит ИУС, имеющая больший эволюционный потенциал. Накопление эволюционного потенциала является целевой функцией всех ИУС. Чем больше эволюционный потенциал, тем лучше ИУС может противостоять разрушительным действиям энтропии.

Но эффективность использования этой свободной энергии определяется когнитивными характеристиками ИУС. Соответственно, формально вычисляемая величина эволюционного потенциала ИУС, позволяет делать прогнозы относительно ее будущего.

1.5. Условно-рефлекторная и рекурсивная модели ИУС.

ИУС «генетически» настроена на стремление к накоплению свободной энергии среды, как необходимого ресурса для противодействия разрушающим действиям физической энтропии.

Соответственно, ИУС из всевозможных реакций на текущую ситуацию должна выбрать ту, которая дает ей наибольшее приращение свободной энергии. При варианте «с учителем», целевой функцией ИУС будет стремление к получению положительных оценок «от учителя» за выбор реакций на текущие ситуации ПО. Из такой постановки вытекает условно-рефлекторная модель ИУС.

Если расширить схему условно-рефлекторной ИУС, добавлением в нее внутреннего канала обратной связи, переводящего образы реакции системы на ее вход (своеобразный аналог мышления) и введением дополнительной структуры из иерархических НСС, то мы получим схему рекурсивной ИУС, которая будет способна автоматически формировать абстрактные понятия и алгоритмы-эвристики для обработки процессов ПО, т.е. самообучаться, см. рис 7.

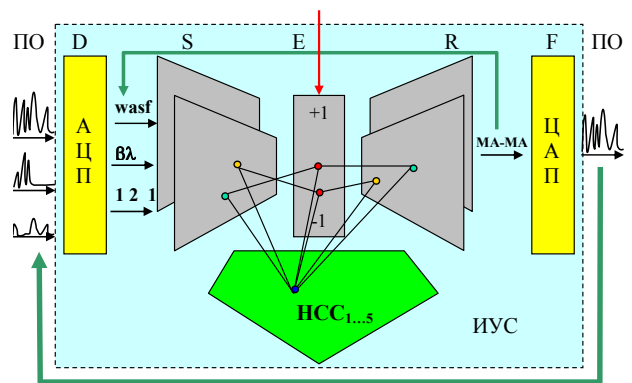


Рис. 7 Архитектура «рекурсивной» ИУС состоит из шести типов блоков: детекторно-рецепторного (D), распознающего (S), двигательного (R), прагматического (E), эффекторного (F) и рекурсивного (НСС_{1...5}).

ЗАКЛЮЧЕНИЕ

Из тезисного изложения материала следует, что на базе нейросемантического подхода можно построить ИУС для работы с неструктурированными и априорно неизвестными информационными потоками. При этом, нейросемантический подход позволяет решить основную проблему современного программирования – автоматизацию формирования информационной (семантической) модели из априорно неизвестных информационных потоков, порожденных из процессов исследуемой предметной области, а также, формирование эвристических правил ИУС не программированием, а посредством ее самообучения. Что позволяет полностью автоматизировать весь цикл создания крупномасштабных ИУС.

Анализ нейросемантического подхода показал большую корреляцию по различным характеристикам (работа с пространственно-временным сигналом, многопараметрическим выходом нейрона, «проращиваемостью»

межнейронных связей в режиме реального времени, иерархичностью нейросетевых структур и др.) с современными нейрофизиологическими данными.

Интересны рассмотренные, вытекающие из нейросемантики естественно-научные определения компонент информационного ресурса:

- Сигнал – произвольный отрезок оцифрованного процесса (отражает характеристику свойств потока текстовой формы);
- Образ – вычлененная информационной системой знаковая последовательность текстовой формы, тождественная целому числу причинно-связанных процессов ПО (семантическая единица ПО);
- Информация – прагматически причинно-связанная последовательность образов, ориентированная на достижение ИУС какого-либо ее целевого образа ПО (прагматическая единица ИУС);
- Знание – причинно-связанное объединение информации о ПО с информацией о состоянии ИУС (семантические закономерности).

А так же, определения степени когнитивности различных классов ИУС:

- Автоматная – с жестко «запаянным» алгоритмом (характерно «зацикливание» при ошибках и необучаемость);
- Адаптивная – возможность самообучения по управлению любым априорно неизвестным объектом за счет перебора всего пространства его состояний (аналог универсальной адаптируемости живого);
- Интеллектуальная – свойство практического преодоления «проклятия размерности» в «гладких» пространствах состояний управляемых объектов за счет набора эвристик;
- Разумная – возможность самоформирования эвристик (отношений).

Анализ приведенных определений указывает на фундаментальность нейросемантического подхода.

Данная работа выполнена при частичной поддержке РФФИ, проект № 11-07-00007-а.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Бодякин, 2006] Бодякин В.И. Определение понятия «информация» с позиций нейросемантики. М.: ИПУ РАН. 2006, 48 с. http://www.mtas.ru/search_results.php?short_view=0&publication_id=3033

[Бодякин, 1998] Бодякин В.И. Куда идешь, человек? (Основы эволюциологии. Информационный подход). - М. СИНТЕГ, 1998, 332с. <http://www.ipu.ru/stran/bod/monograf.htm>

[Бодякин, 2009] Бодякин В.И. Разработка инструментария информационно-управляющих систем на базе нейросемантического подхода для построения крупномасштабных информационно-управляющих систем // XI Всероссийская научно-техническая конференция "Нейроинформатика-2009": М.: МИФИ, 2009. Ч.2. С. 27-38,

[Грекул, 2008] Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем. Интернет-университет информационных технологий ИНТУИТ.ру, 2008 г., 304 стр.

[Бодякин, 2011] Бодякин В.И. «Механизм автоматического формирования информационной модели в информационно-управляющей системе, построенной на базе нейросемантической парадигмы» // Вторая Всероссийская конференция "Нелинейная динамика в когнитивных исследованиях", г. Нижний Новгород, 2011 г. 16-19с. - 260 с.

[Бодякин, 2011] Бодякин В.И. «Анализ формирования когнитивных функций, построенной на базе нейросемантической парадигмы» // Вторая Всероссийская конференция "Нелинейная динамика в когнитивных исследованиях", г. Нижний Новгород, 2011 г. 20-23с. - 260 с.

[Глушков, 1981] Глушков В.М., Валах В.Я. Что такое ОГАС? – М.: Наука, 1981 – 160 с

[Бир, 1993] Бир С. Мозг фирмы: Пер. с англ. – М.: Радио и связь, 1993. 600с

AUTOMATIC CONSTRUCTION OF SEMANTIC MODELS OF ARBITRARY SUBJECT DOMAINS ON BASE OF NEURONSEMANTIC APPROACH

V.I. Bodyakin

V.A. Trapeznokov Institute of Control Sciences, 65 Profsoyuznaya, Moscow 117997, Russia,

body@ipu.ru

The method of automatic exposure of semantic component is in-process investigated from the continuous text stream of the real subject domain. The basic idea of method consists in minimization of reflection of text stream on the specialized neuronsimilar environments. In theory it is well-proven and confirmed by the results of computer experiments on a hypothetical subject domain, that during minimization of reflection an informative model is automatically formed in a neuronsemantic environment, homomorphous to the arbitrary subject domain.

Keywords: neurosimilar structure, neurosemantic, intellectual systems.

CONCLUSION

The computer construction of dictionaries of minimum form showed that both for simple subject domains and for the hierarchical processes of arbitrary complication property of гомоморфности is executed. Thus, the examined mechanism opens practical possibility of the automatic forming of informative (semantic) models of arbitrary subject domains, that will allow to give a new impulse to works on self-training.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89: 004.82: 519.179

О НЕКОТОРЫХ ОБЛАСТЯХ ПРИЛОЖЕНИЙ АЛГЕБРАИЧЕСКОЙ МОДЕЛИ K-ГИПЕРПРОСТРАНСТВА СХ-ГИПЕРТОПОГРАФОВ

Баранович А.Е., Никитин Н.О.

*Российский государственный гуманитарный университет
Институт информационных наук и технологий безопасности,
Центр системного анализа и моделирования мышления
г. Москва, Россия
barae@rambler.ru
Nikita-fin@yandex.ru*

Излагаются результаты краткого аналитического обзора некоторых областей приложений алгебраической модели k -гиперпространства семиотико-хроматических гипертопографов (включая ее сетевую интерпретацию), основанного на материалах ряда опубликованных работ, посвященных использованию исследуемой метамодели в различных задачах социальной деятельности. Приводится ряд замечаний концептуального характера, связанных с интерпретацией понятий «семантической технологии» и «семантического представления знаний» в базе информационно-эволюционного подхода к системному анализу и моделированию интеллектуальных систем.

Ключевые слова: гипертопографы семиотико-хроматические, гипертопосети, моделирование знаний, моделирование информации, моделирование информации атрибутов, моделирование сложных систем

ВВЕДЕНИЕ

Вполне определенную актуальность в последнее время приобрела проблема синтеза универсальной модели представления знаний в интеллектуальных системах (уровня «сознания» [Баранович и др., 2012]), поглощающей известные классы существующих и предназначенных не для решения конкретного, сугубо специализированного класса задач, а для параметрически адаптивной генерации эффективных инструментальных средств, вычислительных архитектур и технологий обработки информации, включая представление актуальных знаний, в интеллектуальных системах (ИС) произвольного генезиса и целевой ориентации. Важным аспектом разрешения выше сформулированной проблемы является исследование механизмов проекции синтезированной модели на классы известных и её прагматическая адаптация к различным областям приложений, в том числе к технологиям «компонентного проектирования интеллектуальных систем» [Голенков и др., 2011].

В настоящей работе в краткой форме излагаются апробированные результаты использования унифицированной алгебраической модели k -гиперпространства семиотико-хроматических (СХ-) гипертопографов G_s (СХ-гипертопосетей для

случая сетевой интерпретации модели) в самых различных областях приложений: от моделирования информационных составляющих (информации) состояний сложных динамических систем (не квантового характера), включая интеллектуальные, до представления знаний и синтеза обобщенной модели автономной эволюционирующей (саморазвивающейся) ИС произвольного генезиса («сильного искусственного интеллекта» в терминологии указанной предметной области).

Аксиоматико-терминологический базис используемой модели в кратком изложении представлен в работе [Баранович, 2011а]. Основные свойства модели исследованы в [Баранович, 2003]. Перечень задач соответствует приведенному перечню опубликованных работ библиографического списка.

1. Моделирование и исчисление семанτικο-прагматических атрибутов информации

Дефиниция и общая характеристика явления информации (И.) и её семанτικο-прагматических атрибутов, а также их моделей, проведена в [Баранович, 2011d]. Предметная область моделирования И. и её характеристических свойств, представления знаний и синтеза обобщенной модели автономной эволюционирующей ИС представлена,

в частности, задачами семантико-аксиологической фильтрации информации.

1.1. Исчисление ценности информации

Сложность решения проблемы количественной оценки ценности информации (ЦИ) в ИС определяется, прежде всего, ее многопараметричностью, когда в качестве параметров обобщенного функционала ЦИ выступают множественные характеристические свойства И., проявляющиеся в процессе взаимодействия воспринимающей ИС с окружающей средой [Баранович, 1997, 2002]. При этом процесс моделирования объективных параметров материальных систем (МС) необходимым образом связан с индивидуальными особенностями взаимодействующих систем (объектов, приобретающих субъективные и рефлексивные свойства).

В основу феноменологии исследуемой проблемы положена система постулатов, характеризующая сущность понятия “ценность И.” с общих телеологических позиций [Баранович, 2002]. В частности, ценность информации определена как индивидуальная прагматическая характеристика результата информационного взаимодействия телеологической ИС (τ-ИС) с материальной системой (МС) - внешней средой, определяющая изменение возможностей достижения системой вполне определенных, имманентных или навязанных целей своего существования при использовании воспринимаемой И. (информационных образов взаимодействующей МС).

Методология синтеза методов и моделей исчисления ЦИ базируется на принципах дедуктивного анализа поведения телеологических τ-ИС произвольного характера с априорно определенным пространством целей, информационные состояния которых моделируются элементами измеримого метрического k -гиперпространства SX -гипертопографов Γ_s уровня топологизации k , $k \leq K \leq \infty$ в метаалгебре

$$A_{\{HTG_V^k x\}} = \langle \{HTG_V^k x\}, \Psi_{\{HTG_V^k x\}} \rangle,$$

а динамика функционирования – целевым $HTG_V^k x$ – метаавтоматом

$$\mathcal{A}_{HTG_V^k x} = \langle X, S, Y, \varphi, \psi \rangle,$$

где $X = \{HTG_V^k x_i^x\}$, $S = \{HTG_V^k x_i^s\}$ и $Y = \{HTG_V^k x_i^y\}$, $\{HTG_V^k x_i^x\}$, $\{HTG_V^k x_i^s\}$, $\{HTG_V^k x_i^y\} \subseteq \{HTG_V^k x\}$ – конечные множества входов, внутренних состояний и выходов метаавтомата, φ и ψ – функции, соответственно, переходов и выхода, реализуемые в сигнатуре $\Psi_{\{HTG_V^k x\}}$ [Баранович, 1997, 2002, 2011с;

Баранович и др., 2011; Шанкин, 2004].

Образцом работоспособного программного продукта (исследовательский прототип), осуществляющего численную оценку ценности

входящей в ИС информации, является интеллектуальная программная среда «АКСИОН», ядро которой может быть использовано как для семантико-аксиологического отбора необходимой для потребителя информации (синтез семантических браузеров и поисковых систем), так и при создании прототипа потокового сетевого аксиологического фильтра [Баранович и др., 2008а, 2008б].

1.2. Семантико-аксиологическая фильтрация информации

Избыточность информации становится одной из основных угроз безопасности информационного общества. Фильтрация информационных массивов – процесс «отсеивания» необходимой информации и ее «переработки» в знания, относится к числу самых актуальных проблем в сфере информационной безопасности [Баранович, 2009]. «Избыточность влечет рост диффузии полезной информации. Информация «растворяется» в «информационном мусоре», объем которого, зачастую, превышает объем необходимой и полезной информации. Проблема фильтрации избыточной (бесполезной, вредной) информации в определенном смысле, двойственна к проблеме информационного поиска полезной информации, ибо выявить необходимую информацию из потенциального множества возможных можно и отсеять всю неподходящую информацию.

По мнению большинства авторитетных источников, наиболее слабым звеном информационно-вычислительных сетей является человек (пользователь), поскольку по большинству своих показателей он уже в настоящее время уступает существующим средствам автоматизации информационных процессов. Вследствие определенных психофизиологических особенностей человек как открытая система с конечными эксплуатационными ресурсами не в состоянии самостоятельно ограничить поток внешней информации и эффективно выделять потоки информации, необходимой для успешного функционирования в социуме [Баранович, 2009]. Специфическими объектами современной информационной инфраструктуры являются и самообучаемые антропогенно-технические интеллектуальные системы, вышеречисленные угрозы в отношении которых реализуются аналогичным образом [Баранович, 2009].

Показателем, характеризующим качество «полезности» поступающей пользователю информации определена ее прагматическая ценность, интерпретируемая в контекстах семантики информации и вполне определенного пространства целей ИС.

В качестве модели-универсума семантической информации (объективная и субъективно-прагматическая интерпретации) используется модель k -гиперпространства SX -гипертопографов произвольного порядка k -топологизации множества-носителя, редуцируемая в последующем в

измеримое метрическое булево k -гиперпространство и позволяющие эффективно интерпретировать известные модели представления декларативных знаний (семантические сети и метасети, системы продукций, фреймы, категориальные модели, концептуальные структуры, онтологии (OWL) и т. п.) [Баранович, 2006].

Процесс оценки ценности входящей в систему семантической информации согласно с п. 1.1 моделируется конечным метаавтоматом, в качестве элементов основных множеств входа и внутренних состояний которого выступают СХ-гипертопографы. Количественная оценка ценности информации осуществляется путем исчисления значений синтезированных метрик на универсальной модели семантической информации, аргументами которых выступают текущие и целевые состояния интеллектуальной системы [Баранович 1997, 2002, 2003].

Участки исследуемого входящего документального контекста (последовательные семиотико-лингвистические модели вербальной коммуникации) преобразуются в семантические образы модели-универсума, которые сравниваются с семантическими образами пространства целевых состояний.

Морфизм СХ-гипертопографов в измеримое метрическое булево k -гиперпространство позволяет перейти к непосредственной алгоритмизации разработанных методик и реализации их на существующих средствах вычислительной техники.

Использование «плавающего» интервала топологизации множества-носителя, наряду с использованием «нечетких» вычислений и формальными практическими ограничениями, налагаемыми на мощности задействованных множеств, обеспечивают «работоспособность» модели на существующих средствах вычислительной техники.

По известным оценкам, автоматический аксиологический фильтр позволяет извлечь из общего потока информации (включая информационный мусор) в десятки раз больше полезной информации (в вербальной форме представления) в единицу времени по сравнению с пользователем-экспертом (что эквивалентно аналогичному сокращению объема избыточности информации в специализированном информационном пространстве).

2. Моделирование и представление знаний

Область применения модели k -гиперпространства СХ-гипертопографов в моделировании знаний представлена задачами концентраций знаний в индивидуальных и коллективных базах знаний ИС и формализации семантической компоненты музыкального текста.

2.1 Концентрация знаний

Сущность предлагаемого подхода заключается в

управляемой эволюции «информационной реальности» социума, характеризуемой использованием механизмов существенного сокращения объема циркулирующей в нем И. в целом (отвечающего динамике полиномиального или, более того, линейного роста И. во времени - в противовес неуправляемому экспоненциальному) [Баранович, 2011b].

В процессе управляемой эволюции (развитии) социальной «информационной реальности» реализуется ряд антропогенно-технических механизмов сокращения производства, воспроизводства (размножения) и рассеивания бесполезной и вредной (ложной) информации, а также механизмов концентрации-аккумуляции семантической информации в ограниченном информационном объеме («концентрация знаний» в «информационном» обществе). С количественной точки зрения, речь идет о сведении экспоненциального темпа роста И. к линейно-полиномиальному, что согласуется с ростом ресурсных возможностей человечества по использованию информации (совокупной интеллектуально-вычислительной и коммуникационной мощностью социума).

В качестве основных механизмов осуществления обозначенной концепции целесообразно определить антропогенно-технические ИС (АГИС), обеспечивающие как пассивную (в локальных информационно-коммуникационных узлах), так и активную (в распределенном сетевом пространстве) фильтрацию-концентрацию информации (знаний) с задействованием среды «агентов-мусорщиков» и «агентов-концентраторов». Использование технологии разработки АГИС, основанной на «архитектуре, обусловленной моделированием», предопределяет выбор в качестве центрального звена проектирования механизмов «фильтрации-концентрации» - универсальной семантической модели информации, позволяющей формализовать и алгоритмически описать вышеупомянутые процедуры в отношении самого широкого круга приложений, а именно модель k -гиперпространства семиотико-хроматических гипертопографов G_s^k .

В процессе реализации процедур «фильтрации / концентрации знаний» АГИС формирует семантический образ входящей информации, проверяет ее на истинность / ложность (в рамках своих «интеллектуальных» возможностей) и выявляет полезную, бесполезную и вредную И.

Выявив полезную информацию, АГИС приступает к процедуре «концентрации знаний», объединяя (путем задействования операции «слияния» на метаалгебре $A_{\{HTG_V^k, X\}}$) полезную входящую информации с индивидуальной (или внешней, коллективной) подсистемой знаний. И далее, осуществляя её оптимизационную реструктуризацию (операция «реструктуризации») с использованием алгоритмов сокращения информационного объема знаний (в частности, путём объединения - поглощения их тождественных экземпляров и сокращения опровергнутых) и

повышения эффективности их использования (в отношении, например, поиска и доступа).

2.2. Моделирование семантики полифонического музыкального текста

В работе [Иглицкая, 2012] предлагается механизм проекции модели k -гиперпространства СХ-гипертопографов на весьма слабо исследованную с естественнонаучных позиций область музыкального творчества. На начальном этапе исследований была исследована модель музыкального текста (МТ) так называемого строгого стиля, представленный моделью дискретных сообщений Дж. фон Неймана нулевого и первого приближений [Иглицкая, 2011]. Данная модель относится к классу последовательных семиотических моделей («семантически тривиальных») и, с одной стороны, требует упрощения МТ до уровня модели конечного алфавита, а с другой стороны, не отражает разнообразия структурных связей в нем, а лишь отношения n последовательных символов (разрешенные и запрещенные n -граммы).

В качестве модели «глубинной семантики» выбрана модель k -гиперпространства СХ-гипертопографов. Предложено использовать модифицированную модель СХ-гипертопографа, с расширенным множеством-носителем, характеризующим множественность различных (посредством хроматических атрибутов) экземпляров односортных элементов [Баранович, 2011с] (дальнейшая топологизация реализуется согласно классической модели СХ-гипертопографа). В качестве элементов множества-носителя берутся все нотные знаки произведения, а различные уровни топологизации отражают иерархию связей между ними.

Вершины первого уровня топологизации представлены одноточечными элементами множества-носителя; все характеристики, относящиеся к отдельной ноте (тембр, громкость, артикуляция, а также координаты в произведении - принадлежность определенному голосу и место в такте), могут быть отражены в хроматических атрибутах вершины.

На следующем уровне топологизации возможно два типа отношений:

вертикальное - объединение одновременно звучащих нот в аккорд;

горизонтальное – объединение подряд идущих нот в музыкальную фразу. Здесь могут возникнуть определенные сложности, связанные со спецификой МТ. Если для вербального текста при аналогичном подходе к формированию множества-носителя дальнейшая топологизация очевидна (буквы объединяются в слова, слова в предложения и так далее), то для МТ понятие «музыкальная фраза» весьма расплывчато, их строение во многом зависит от индивидуальных особенностей стиля конкретного произведения, кроме того, возможно пересечение соседних фраз, когда окончание одной является уже началом следующей. Тем не менее, выделение некоторых структурных единиц в

отдельно взятой мелодической линии всегда возможно, из этого следует потенциальная возможность алгоритмизации данной процедуры.

Дальнейшая топологизация предполагает объединение фраз отношениями. Например, повтора, противопоставления, эха, секвенции, вариации и т.д.

На последнем уровне должны быть отражены отношения между частями музыкальной формы (например, экспозиция - реприза).

Развитие предлагаемого подхода предполагает разработку объективных критериев оценки исполнения музыкального произведения, что является весьма актуальной задачей в связи с большим количеством музыкальных конкурсов, традиционно вызывающих огромное количество споров как в профессиональной, так и в любительской среде. Разрабатываемый подход позволяет также проводить более строгую политику дефиниций (в терминах конструируемой модели) в отношении основных гуманитарных понятий, относящихся к области музыковедения.

3. Моделирование информационных составляющих состояний сложных систем

Область моделирования информационной составляющей состояний сложных динамических систем, включая представление знаний, представлена задачей мониторинга поведения (функционирования) биологических систем, включая антропоморфные.

Общая постановка задачи в рамках разработанного формального аппарата представлена в работах [Баранович и др., 2007, 2008а], где проведен углубленный численный анализ информационных характеристик существующего процесса медицинского обеспечения в Российской Федерации и предложен ряд современных технологических решений по реализации национальной программы постоянного (в модели времени «по наступлению события») мониторинга социума с использованием индивидуального защищенного электронного персонального реестра, обеспечивающего информационное сопровождение полного жизненного цикла субъекта.

Важнейшим принципом реализации процесса мониторинга является представление и использованием информации не формате классических данных, но в формате формализованных знаний. Причем в основу аппарата формализации должны быть положены математические модели представления знаний в соответствующей подсистеме ИС. С точки зрения перспективных тенденций в создании современных автоматизированных информационных систем (АИС), речь идет о целесообразности задействования в системе мониторинга общей для всех участников процесса модели, определяющей основные принципы развития информационной

базы АИС и интеграции различных приложений в систему.

Возможны два варианта использования знаний в системе мониторинга. Первый реализуется в режиме ввода, хранения и использования информации в формате модели представления знаний. В этом случае, мы имеем дело с элементом подсистемы знаний интегрированной ИС, использование и преобразование, которого осуществляется соответствующими интеллектуальными средствами. Второй реализуется в режимах ввода и хранения информации в стандартизованных форматах данных (соответствующих, в частности, форматам существующих баз данных), использование которых в прикладных приложениях (например, с целью диагностики или организации лечения) в данном случае связано с преобразованием накопленных данных в совокупность знаний.

Выбор того или иного механизма накопления и использования знаний определяется емкостными параметрами моделей данных и носителей информации, а также скоростью прямого / обратного преобразования данных в знания и обратно (реализуемых, в частности, лингвистическими процессорами с вполне определенными операционными характеристиками эффективности реализации).

В качестве базового аппарата моделирования знаний используется алгебраическая модель измеримого метрического k -гиперпространства G_s и $HTG_V^k x$ -метаавтомата на базе одноосновной метаалгебры $A_{\{HTG_V^k x\}}$, [Баранович, 2003, 2007], обеспечивающего моделирование динамических процессов функционирования системы. Поступающая в ИС информация преобразуется в формат «входящего» СХ-гипертопографа, сравнивается с текущим состоянием ИС (возможно и с иными целевыми или эталонными состояниями) и интегрируется в базу знаний ИС (k -гиперпространство G_s) с использованием операций «объединения», «пересечения», «развития», «слияния» и «трансформации» СХ-гипертопографов [Баранович и др., 2011]. Элементы множества $HTG_V^k x$ спроецированные на области формирования, хранения, модификации и использования знаний, моделируют как состояния и цели контролируемой системы в период ее жизненного цикла, так и произвольные информационные объекты ИС, в частности, эталоны синдромов заболеваний.

Метрические свойства использованной модели позволяют путем задействования вполне определенного класса нормированных метрик и мер на G_s исчислять вполне определенные количественные отклонения контролируемых текущих состояний от предшествующих или эталонных, тем самым осуществляя интегральную диагностику на множестве заданных синдромов.

4. Ряд замечаний концептуального характера

Предшествующие разделы носили выраженный прагматический характер. Настоящий раздел отражает эволюцию взглядов авторов на содержательные аспекты направления «Универсальные базовые абстрактные семантические модели представления и обработки знаний, обеспечивающие интерпретацию различных моделей решения задач». Последнее касается, прежде всего, строгой содержательной интерпретации используемых определений, терминов и словосочетаний.

Широко распространено заблуждение, характеризующее ошибочную классификацию на независимые (слабосвязанные) лексико-морфологический, синтактико-грамматический и семантический уровни системного анализа и моделирования вербальной информации (текста).

На самом деле, в рамках разработанных и успешно апробированных атрибутивно-ингредиентной концепции информации (АИКИ) и информационно-эволюционного подхода (ИЭП) к системному анализу и моделированию (САМ) объективной реальности (ОР) речь идет о сквозном сильносвязанном семействе моделей информационных прообразов взаимодействующих материальных систем (МС) в процессе семантической коммуникации ИС [Баранович, 2002, 2010, 2011d, 2011e]. В данном контексте, реализуется сквозная экспликация моделей семантики информации – от «поверхностной» (сенсорные информационные образы) до «глубинной» (информационные прообразы МС) в АИКИ.

Соответственно, в отношении вербальной коммуникации (текстовой) ИС, семантика последовательно характеризуется моделями («семантически тривиальными») лексики, морфологии, далее грамматики и синтаксиса.

Модели информационных образов в семантической коммуникации ИС характеризуются уровнем их адекватности (соответствия) информационным прообразам МС ОР. Наиболее грубые модели, лишь частично отвечающие требования адекватности, относятся к классу моделей «поверхностной» («тривиальной») семантики. Более адекватные, точные и детализированные - к классам моделей «срединной» и «глубинной» семантики.

Напомним, что в АИКИ семантика И. объективная характеризует информационные составляющие состояний существования МС ОР, а именно элементы и структуру системы, а также их характеристические атрибуты-значения в «текущем настоящем» в модели времени «по наступлению события».

Семантика же субъективная (прагматическая) интерпретируется в рамках ИЭП как динамический информационный образ объективной семантики (информационного прообраза взаимодействующей МС «внешнего мира») инициализированный в

подсистеме знаний воспринимающей ИС.

Таким образом, все модели И. могут быть охарактеризованы как семантические, того или иного уровня приближения к информационным прообразам моделируемых систем.

Что же тогда есть «семантические технологии проектирования интеллектуальных систем» в контексте выше высказанных соображений. И что есть тогда «не семантические технологии» или просто «технологии проектирования интеллектуальных систем»?

В свою очередь, знания, согласно АИКИ и ИЭП, есть информация, точнее, информационные модели, характеризующиеся рядом вполне определенных свойств [Баранович и др., 2012]. Желая декларировать знания иным образом, приглашаются к конструктивной дискуссии (с изложением собственной дефиниции, и никак иначе).

Чем же тогда отличается «семантическое представление знаний» от своей антиномии «не семантическое представление знаний»? Использованием модели «семантической сети» в качестве базовой модели представления знаний? Весьма неопределенно.

Аналогичные вопросы возникают и в отношении вульгарно-свободной интерпретации понятия «интеллектуальная обработка данных». И чем она отличается от «обработки данных»? Если ничем, то место данному неологизму в корзине «информационного мусора». Если отличия представлены неопределяемыми в лексиконе естественной науки мифологемами – туда же.

Заметим при этом, что попытки свести идентификацию «интеллектуальности» процессов обработки информации к его сравнению с аналогами антропоного интеллекта, (исключая, пока (!), непознанные явления), априори не могут привести к положительным результатам, вследствие, прежде всего, доказанной неэффективности целого ряда основных механизмов оперирования информацией в биологических системах, включая антропные.

В результате:

1. Все что относится к обработке данных (в ЭВМ) декларируется «неинтеллектуальным» процессом. Несмотря на то, что последний является вершиной интеллектуальной эволюции настоящей актуальной реализации ОР и не имеет аналогов в природе. Более того, точные классические методы и модели достаточно эффективно проецируются на их «размытые» образы. Обратное, в общем случае, неверно.

2. То, что декларируется «интеллектуальным» процессом, либо относится к биоинспирированным субоптимальным методам оперирования классами, а не их точными представителями, либо относится к действиям и задачам методы, модели и аксиоматические системы, решения которых на сегодняшний день не синтезированы или не обрели окончательного работоспособного облика.

3. Все (!) работоспособные интеллектуальные методы, включая так называемые «нечеткие» вычисления и нейроинформатику, реализуются в

настоящее время путем интерполяционного моделирования на классической архитектуре вычислений А. Тьюринга - Дж. фон Неймана. Причем символные модели наиболее эффективно реализуются (в соответствии с принципом максимальной простоты сложных систем) в моделях алгебраических систем, обычно, числовых алгебр.

ЗАКЛЮЧЕНИЕ

Вышеприведенный материал вполне обоснованно свидетельствует о перспективности использованного при синтезе и анализе универсальной абстрактной модели семантики И. информационно-эволюционного подхода к системному анализу и моделированию интеллектуальных систем произвольного генезиса [Баранович, 2010, 2011e]. ИЭП САМ характеризуется постнеклассическим междисциплинарным естественнонаучным базисом и существенным образом отличается от превалирующих до настоящего времени «логико-лингвистического», вторичного в отношении аппарата информационно-эволюционного моделирования сложных динамических систем, и «нейробионического», исторически связанного с методологией имитационного анализа «геобиологического» носителя интеллекта. Последнее предопределяет перспективы применения СХ-гипертопографов и СХ-гипертопосетей, включая их модификации, в качестве базиса семейства моделей семантики И. (мышления уровня «сознания») в самых различных приложениях всевозможных предметных областей. В том числе, в областях теории и практики интеллектуальных систем, характеризующихся актуальными проблемами, считающимися неразрешимыми до настоящего времени в рамках господствующих парадигм.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Баранович, 1997] Баранович А.Е. Автоматная модель интеллектуального процесса оценки ценности информации на Х-гиперграфах: в сб. «Универсальный подход к структурному моделированию директивно-целевых информационных процессов» // Сб. статей. М., МО РФ, 1997. - С. 23-47.
- [Баранович, 2002] Баранович А.Е. Структурное метамоделирование телеологических информационных процессов в интеллектуальных системах. М., МО РФ, 2002. - 316 с.: ил.
- [Баранович, 2003] Баранович А.Е. Семиотико-хроматические гипертопографы. Введение в аксиоматическую теорию: информационный аспект. М.: МО РФ, 2003. - 404 с.: ил.
- [Баранович, 2006] Баранович А.Е. К-гиперпространство семиотико-хроматических гипертопографов как универсальная модель представления фактографических знаний // Матер. IX междунар. конф. «Интеллект. сист. и компьют. науки». Т. 1, ч. 1. М., МГУ, 2006. - С. 53-55.
- [Баранович и др., 2007] Баранович А.Е., Баранович А.А., Кузнецова И.А., Мерзликин В.Г. Моделирование процесса информационного сопровождения жизненного цикла биологической системы // Матер. 8-ой междунар. науч.-техн. конф. «Кибернетика и высокие технологии XXI века» С&Т-2007. Воронеж, ВГУ, 2007. - С. 129-143.
- [Баранович и др., 2008a] Баранович А.А., Кузнецова И.А., Лишин Н.А. Мониторинг жизненного цикла биологической системы в ИПС «Аксион». Тр. VIII Междунар. научн.-техн.

конф. «Интеллектуальные системы» (AIS'08). – М., Физматлит, 2008. – С. 238-245.

[Баранович и др., 2008б] Баранович А.А., Лишин Н.А. Исчисление ценности прагматической информации в интеллектуальной программной среде «Аксион» // Тр. XI национ. конф. по ИИ с междунар. участ. (КИИ-08). Т.2. М., ЛЕНАНД, 2008. – С. 364-372.

[Баранович, 2009] Баранович А.Е. Прагматические аспекты информационной безопасности интеллектуальных систем // Вестник РГГУ. Сер. «Информатика. Защита информации. Математика» / Рос. гос. гуманит. ун-т. – Вып. 10 (2009). – М., РГГУ, 2009. – С. 56-70.

[Баранович, 2010] Баранович А.Е. О систематизации аксиоматического аппарата предметной области «Искусственный интеллект» / Интеллектуальные системы. Т. 14. Вып. 1-4. – М., 2010. – С. 5-34.

[Баранович, 2011а] Баранович А.Е. Семиотико-хроматические гипертопосети: унифицированная модель представления знаний / Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2011): мат. Междунар. научн.-техн. конф. – Минск: БГУИР, 2011. – С. 71-86.

[Баранович, 2011б] Баранович А.Е. Семантические аспекты информационной безопасности: концентрация знаний // Вестник РГГУ. Сер. «Информатика. Защита информации. Математика» / Рос. гос. гуманит. ун-т. – Вып. 13/11 (2011). – М., РГГУ, 2011. – С. 38-58.

[Баранович, 2011с] Баранович А.Е. Многоосновные CX-гипертопографы - однообъектная парадигма / Тр. III Междунар. конгресса по интеллект. системам и информ. технол. / XI Междунар. научн.-техн. конф. «Интеллектуальные системы» (AIS'11). – М., Физматлит, 2011. Т. 1. – С. 377-385.

[Баранович, 2011д] Баранович А.Е. Введение в информациологию и ее специальные приложения: дидактические материалы к специальному курсу. М., РГГУ, 2011. – 268 с.: ил.

[Баранович, 2011е] Баранович А.Е. Информационно-эволюционный подход в теории интеллектуальных систем // Матер. X Междунар. конф. «Интеллект. сист. и компьют. науки». – М., МГУ, 2011 (в печ.).

[Баранович и др., 2011] Баранович А.Е., Боровиков Д.В., Лакуша Е.Л. Об алгебраизации модели k -гиперпространства CX-гипертопографов: операции трансформация – развития. // Матер. X Междунар. конф. «Интеллект. сист. и компьют. науки». – М., МГУ, 2011 (в печ.).

[Баранович и др., 2012] Баранович А.А., Ханковский Д.Б. О моделировании взаимодействия подпроцессов мышления уровней «сознание» - «подсознание» // Вестник РГГУ. Сер. «Информатика. Защита информации. Математика» / Рос. гос. гуманит. ун-т. – М., РГГУ, 2012 (в печ.).

[Голенков и др., 2011] Голенков В.В., Гулякина Н.А. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем / 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.

[Иглицкая, 2011] Иглицкая С.М. К вопросу структурно-алгебраического и семантико-прагматического анализа музыкального текста // Вестник РГГУ. Сер. «Информатика. Защита информации. Математика» / Рос. гос. гуманит. ун-т. – Вып. 13/11 (2011). – М., РГГУ, 2011. – С. 59-74.

[Иглицкая, 2012] Иглицкая С.М. Об одном подходе к моделированию семантики полифонического музыкального текста // Вестник РГГУ. Сер. «Информатика. Защита информации. Математика» / Рос. гос. гуманит. ун-т. – М., РГГУ, 2012 (в печ.).

[Шанкин, 2004] Шанкин Г.П. Ценность информации. Вопросы теории и приложений. М.: Филоматис, 2004.

ABOUT SOME APPLICATION AREAS OF THE ALGEBRAIC MODEL FOR k – HYPERSPACE OF SH-HYPERTOPOGRAPHS

Baranovich A.E., Nikitin N.O.

*Institute for Information Sciences
and Security Technologies
Russian State University for the Humanities
System Analysis and Modeling of Thinking
Centre,
Moscow, Russian Federation*

barae@rambler.ru

Nikita-fin@yandex.ru

The results of the brief analytical review for some application areas of the algebraic model for k -hyperspace of semiotic-chromatic (SH-) hypertopographs (including the network interpretation in the format of SH-hyperponets), based on the material of several works devoted to the use of the investigated metamodel in various problems of social activity are presented. A number of conceptual observations connected with the interpretation of the concepts of “semantic technology” and “semantic knowledge representation” in the basis of the information-evolutionary approach to system analysis and modeling of intelligent systems (IS) are stated.

INTRODUCTION

Recently the problem of the synthesis of the universal model for the representation of knowledge in IS (“consciousness” level), absorbing known existing classes and intended not for solving the concrete and highly specialized class of problems, but for parametrically adaptive generation of effective tools, computing architectures and information processing technologies, including representation of actual knowledge, in intelligent systems of any genesis and target orientation has become very actual. The research of the mechanisms of the projection of the synthesized model on known classes and its pragmatic adaptation to various areas of applications, including technologies of “componential designing of intelligent systems» is an important aspect of the formulated problem permission.

MAIN PART

The subject area of modeling information (I.) and its characteristic properties, knowledge representation and synthesis of the generalized model of autonomous evolving IS is presented, in particular, by problems of semantic-axiological filtration of the information.

As model-universe of the semantic information (objective and is subjective-pragmatic interpretations) the algebraic model of k -hyperspace of SH-

hypertopographs $\{I TG_V^k x\}$, of any order k -topologization of the set-carrier, reduced in measurable metric boolean k -hyperspace and allowing effectively to interpret known models of representation of knowledge (semantic nets and meta nets, systems of productions, frames, categorical models, conceptual structures, ontology, etc.) is used. The methodology of synthesis of methods and models of calculation of value of the information is based on results of the deductive analysis of the behavior of teleological IS of any character with a priori certain space of the purposes which information status is modeled by elements of measurable metric k -hyperspace of SH-hypertopographs of level of topologization k , $k \leq K \leq \infty$ in metaalgebra

$$A_{\{HTG_V^k x\}} = \langle \{HTG_V^k x\}, \Psi_{\{HTG_V^k x\}} \rangle,$$

and the dynamics of functioning - the target metaautomat

$$\mathcal{A}_{HTG_V^k x} = \langle X, S, Y, \varphi, \psi \rangle,$$

where $X = \{HTG_V^k x_i^x\}$, $S = \{HTG_V^k x_i^s\}$ and $Y = \{HTG_V^k x_i^y\}$ - final sets of inputs, inwardnesses and metaautomat exits, φ and ψ - functions, accordingly, transitions and an exit, realized in signature $\Psi_{\{HTG_V^k x\}}$.

The field of application of models of k -hyperspace of SH-hypertopographs in modeling of knowledge is presented by problems of concentration of knowledge in individual and collective knowledge bases of IS and formalization semantic components of the musical text. The essence of the mechanism of concentration of knowledge consists in operated evolution of "an information reality» of the society, characterized by the use of mechanisms of essential reduction of the volume of I. circulating in it in whole (responding dynamics polynomial or, moreover, linear growth of I. in time - in a counterbalance uncontrollable exponential). As model of "deep semantics» polyphonic musical text the modified model of k -hyperspace of SH-hypertopographs with the expanded set-carrier characterizing plurality distinguishable (by means of chromatic attributes) copies of monosort elements is chosen. As set-carrier elements all musical signs of the product are taken, and various levels of topologization reflect the hierarchy of communications between them.

The area of modeling of an information component of status of complex dynamic systems, including representation of knowledge, is presented by a problem of monitoring of behavior (functioning) of biological systems, including the anthropomorphous ones. Metric properties of the used model allow to estimate quite certain quantitative deviations of controllable current statuses from previous or reference, thereby carrying out integrated diagnostics on set of defined syndromes by involvement of quite certain class of normalized

metrics and measures.

Within the limits of developed the attributive-ingredient concept of the information and the information-evolutionary approach to the system analysis and modeling of an objective reality the through explication of models of semantics of the information (information prototypes of co-operating material systems) - from "superficial" (touch information images) to "deep" (information prototypes of systems) is actualized and the not transitive terminological system of more strict substantial interpretation of used concepts is offered.

CONCLUSION

The above material quite rightly indicates the perspectives of the used at synthesis and the analysis of universal abstract model of semantics I. information-evolutionary approach to the system analysis and modeling IS of any genesis. The offered approach is characterized by postnonclassical interdisciplinary natural-science basis and essentially differs from prevailing till now "logical-linguistic", secondary concerning the device of information-evolutionary modeling of complex dynamic systems, and from "neurobionical", historically connected with the methodology of the imitating analysis of the "geobiological" intelligence carrier. The last determines the prospects of SH-hypertopographs and SH-hypertoponets, including their modifications, as basis of kind of models of semantics I. (thinking of level of "consciousness") in the most various applications of every possible subject domains, including areas of the theory and practice of the intelligent systems characterized by actual problems, considered unsolvable till now within the limits of dominating paradigms.



УДК 004.8

ПОДХОД К ХРАНЕНИЮ БАЗ НЕЧЕТКИХ ЗНАНИЙ

Глоба Л. С., Терновой М. Ю., Штогрин Е. С.

*Национальный технический университет Украины «Киевский политехнический институт»,
Институт телекоммуникационных систем,*

г. Киев, Украина

lgloba@its.kpi.ua

ternovoy@its.kpi.ua

L_Shtogrina@mail.ru

В работе предложен подход к хранению баз нечетких знаний в реляционной базе данных. Приведена модель базы нечетких знаний как объединение частичных баз. Описаны этапы проектирования схемы базы данных для хранения базы нечетких знаний.

Ключевые слова: база нечетких знаний, лингвистические переменные, отношение, реляционная база данных.

ВВЕДЕНИЕ

Повышение требований к обработке информации в различных областях жизнедеятельности человека требует улучшения существующих и создания новых подходов, позволяющих более эффективно хранить, обрабатывать и представлять информацию в процессе принятия различных решений. Среди существующих подходов к обработке информации следует отдельно выделить те, которые базируются на использовании баз знаний (БЗ) [Субботин, 2008]. Данные подходы позволяют объединить знания экспертов и в последующем их использовать в автоматизированном режиме. При этом необходимо учитывать, что эксперт не всегда оперирует четкой информацией, а также использует в своих рассуждениях не только количественные, но и качественные категории. Для хранения таких знаний используются базы нечетких знаний (БНЗ) [Ротштейн, 1999].

Существует два основных подхода к представлению знаний: процедурный и декларативный [Братко, 2004; Субботин, 2008]. При использовании процедурного подхода знания представляются в виде программного кода, который выполняется. При декларативном – в виде фактов, с которыми работает исполняемая программа. Второй подход является более эффективным для использования в распределенной информационно-телекоммуникационной среде, так как позволяет одновременно использовать базу знаний различным программам.

При декларативном подходе существуют

различные форматы хранения знаний, такие как текстовые файлы, xml документы [Zhang et al. 2011], специальные языки представления знаний [KML].

Представление сложных баз знаний в указанном выше виде затрудняет их одновременное использование. Также недостатком такого хранения является сложность поддержания целостности и непротиворечивости, дублирование знаний, а также отсутствие универсального механизма доступа к знаниям. Выходом из данной ситуации может стать использование реляционных баз данных для хранения БЗ, которые, как раз, и ориентированы на работу в распределенной информационно-телекоммуникационной среде и позволят повысить эффективность работы со знаниями.

1. Модель базы нечетких знаний

В данной работе рассматривается представление знаний в виде нечеткой логической модели. В такой модели выделяются три части: язык, система аксиом и правила вывода. В случае модели сформированной средствами нечеткой логики язык представляет собой лингвистические переменные (ЛП), их терм-множества и структуру правил, к аксиомам относится база нечетких знаний (БНЗ), а правила вывода задаются формулами, с помощью которых определяется значение функции принадлежности (ФП) ЛП и имеют вид:

$$\text{ЕСЛИ}(X_{ml} = a_{ml}) \text{И} \\ (X_{jl} = a_{jl}) \text{И} \dots \text{И}(X_{kl} = a_{kl}) \text{ТО}(Y = d_{nl})'$$

где $X_{ml}, X_{jl}, X_{kl}, Y$ - лингвистические переменные, которые оцениваются качественными терминами $a_{ml}, a_{jl}, a_{kl}, d_{nl}$ соответственно, l обозначает номер правила.

Одним из недостатков данной модели является то, что в общем случае БНЗ неупорядоченная, что приводит к полному перебору правил при логическом выводе. Избежать данного недостатка можно используя подход сведения БНЗ к древовидной структуре, который рассматривается в работе [Глоба и др., 2008]. Также сведение БНЗ к древовидной структуре дает возможность отобразить природную иерархичность корпоративных систем, что позволяет наглядно проследить зависимости и связи между объектами. После применения данного подхода БНЗ будет состоять из частичных БНЗ, которые в совокупности организуют древовидную структуру. Обозначим частичную БНЗ как \bar{R} . Каждая из частичных БНЗ представляется в виде матрицы знаний, and/or графа или правил вида:

$$\begin{aligned} & \text{ЕСЛИ}(X_1 = a_1^{k1}) \text{ И } (X_2 = a_2^{k1}) \text{ И... И } (X_S = a_S^{k1}) \text{ ИЛИ} \\ & (X_1 = a_1^{k2}) \text{ И } (X_2 = a_2^{k2}) \text{ И... И } (X_S = a_S^{k2}) \text{ ИЛИ..} \\ & \text{ИЛИ} (X_1 = a_1^{kq}) \text{ И } (X_2 = a_2^{kq}) \text{ И... И } (X_S = a_S^{kq}) \end{aligned}$$

$$\text{ТО } Y = d_k$$

где a_i^{kr} - нечеткий терм, которым оценивается ЛП X_i ; $k = \overline{1, n}$, где n - количество разных значений, которые используются для оценки выходной переменной Y ; $r = \overline{1, q}$, где q количество элементарных конъюнкций в данном правиле.

2. Схема базы данных для хранения базы нечетких знаний

Необходимость использования баз нечетких знаний различными приложениями в распределенной информационно-телекоммуникационной среде указывает на необходимость создания реляционной базы данных, что позволит эффективно хранить и использовать знания.

Для решения вышеописанной проблемы необходимо решить задачу формирования схемы реляционной БД для хранения представленной модели БНЗ. Эту задачу можно записать в виде:

Дано:

Математическая модель БНЗ:

1. $\{X_i | i = \overline{1, v}\}$ - множество лингвистических переменных (ЛП), которым соответствуют $\{x_i | i = \overline{1, v}\}$, где ЛП - пятерка $\langle X, T, U, G, H \rangle$, в которой:

- 1.1. X - имя ЛП;
 - 1.2. $T = \{t_j | j = \overline{1, z}\}$, где z - терм-множество ЛП X с соответствующими функциями принадлежности (ФП)
 $M = \{\mu_{X_j} | j = \overline{1, z}\}$.
 - 1.3. U - универсальное множество, область определения ЛП;
 - 1.4. $G = \{g_k | k = \overline{1, s}\}$ - синтаксические правила, в виде грамматики, которые порождают названия термов;
 - 1.5. $H = \{h_k | k = \overline{1, s}\}$ - семантические правила, которые задают функции принадлежности нечетких термов, порожденных синтаксическими правилами G .
 2. y - результирующая переменная;
 3. Y - ЛП, которая соответствует y ;
- $BZ = \bigcup \bar{R}$ - древовидная база нечетких знаний

(БНЗ), где \bar{R} - частичная БНЗ [Глоба и др., 2008].

Найти:

Схему реляционной БД $\langle B, L, C \rangle$ для хранения БНЗ, где

1. $B = \{b_q | q = \overline{1, n}\}$ - множество отношений БД, для которых определены атрибуты $A_q = \{a_{qp} | p = \overline{1, m}\}$ и накладываемые на них ограничения $S_p = \{s_{pk} | k = \overline{1, r}\}$;
2. $L = \{l_v | v = \overline{1, h}\}$ - множество связей между отношениями B .
3. C - правила и ограничения, которые обеспечивают целостность БД и поддерживают правильность внесения знаний.

Для решения поставленной задачи опишем концептуальный и логический этапы проектирования БД в соответствии с [Коннолли и др., 2003].

Первым шагом на этапе концептуального проектирования БД является определение сущностей предметной области. Исходя из описанной выше модели БНЗ, можно выделить две сущности, такие как ЛП "LinguisticVariable" и правило вывода "Rule". Связь между этими сущностями будет отображать, что ЛП является частью правила. Также для более детального описания можно ввести третью сущность ФП "MembershipFunction". Рассматривая информацию, которая описывает эти сущности можно определить свойства для этих сущностей. Для описания ЛП необходимо описать каждый ее терм, поэтому свойствами будут "название ЛП", "полное название термина ЛП" и "функция принадлежности". Для правила необходимо описать информацию о его

левой и правой частях, поэтому для сущности “Rule” свойствами будут “условие” и “следствие”. Также необходимо предусмотреть то, что правило содержит не одну составляющую “условие”, а несколько.

На следующем шаге концептуального проектирования необходимо определить ограничения, которые будут обеспечивать целостность данных и соблюдение особенностей информации представления нечетких знаний:

1. ЛП должна быть задана на множестве ее определения.
2. Если ЛП содержится в условии заданного правила, то она не может быть следствием этого правила.
3. Если два правила имеют одинаковые условия, то они не могут иметь различные следствия.

Далее рассмотрим этап логического проектирования. Так как выбран реляционный тип БД, то просто сопоставить сущности с отношениями, а свойства с атрибутами отношений нельзя, так как полученная в результате такого сопоставления схема будет не нормализована, в ней будут присутствовать аномалии и избыточность данных. Также необходимо учесть, что разные ЛП могут иметь одинаковые названия термов и синтаксические правила, которые образуют названия термов. Следовательно, для уменьшения избыточности можно определить классификаторы для таких множеств, которые возможно будет расширять и дополнять. Зададим их с помощью введения отношений “TermName” и “RuleForCreateTermName” соответственно. В результате появляется связь многие-ко-многим (каждой ЛП соответствует несколько термов, и каждый терм может соответствовать нескольким ЛП), которую в реляционной БД необходимо реализовать с помощью третьего отношения, в данном случае “Term”.

Так как любую непрерывную функцию можно аппроксимировать сколь угодно близко кусочно-линейной функцией, то можно допустить, что функции принадлежности задаются не в виде формул, а в виде наборов координат. Такое допущение позволит хранить информацию о ЛП, правилах и ФП в одном месте, и не зависеть от программного обеспечения, которое реализует определение и вычисление функций принадлежности. Еще одним преимуществом такого хранения является возможность введения классификатора для ФП (реализован отношением “MembershipFunction”), что в дальнейшем предоставит возможность определения имеют ли различные ЛП одинаковые ФП. Для хранения координат ФП введем отношение “Func_XY”.

Для хранения правил определим отношения “FuzzyRule” и “KnowledgeMatrix”.

Предложенная схема хранения будет состоять из отношений $B = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\}$, где

- b_1 - отношение “LinguisticVariable” в котором $A_1 = \{a_{11}, a_{12}, a_{13}\}$, где a_{11} - “LinguisticVariableID” уникальный идентификатор ЛП, a_{12} - “Name” название ЛП, a_{13} - “Description” описание ЛП.
- b_2 - отношение “RuleForCreateTermName” в котором $A_2 = \{a_{21}, a_{22}\}$, где a_{21} - “RuleForCreateTermNameID” уникальный идентификатор, a_{22} - “Name” синтаксическое правило для порождения названия термина.
- b_3 - отношение “TermName”, в котором $A_3 = \{a_{31}, a_{32}\}$, где a_{31} - “TermID” уникальный идентификатор, a_{32} - Name название термина.
- b_4 - отношение “MembershipFunction”, в котором $A_4 = \{a_{41}, a_{42}, a_{43}\}$, где a_{41} - “MembershipFunctionID” уникальный идентификатор, a_{42} - “Name” название функции, a_{43} - “Description” описание функции.
- b_5 - отношение “Func_XY”, в котором $A_5 = \{a_{51}, a_{52}, a_{53}\}$, где a_{51} - “MembershipFunctionID” уникальный идентификатор, a_{52} - “X” X координата функции, a_{53} - “Y” Y координата функции.
- b_6 - отношение “Term”, в котором $A_6 = \{a_{61}, a_{62}, a_{63}, a_{64}\}$, где a_{61} - “LinguisticVariableID”, a_{62} - “TermID”, a_{63} - “RuleForCreateTermNameID”, a_{64} - “MembershipFunctionID”. $a_{61}, a_{62}, a_{63}, a_{64}$ - образуют составной ключ. Отношение “Term” является связующим звеном для всех остальных отношений.
- b_7 - отношение “FuzzyRule”, в котором $A_7 = \{a_{71}, a_{72}\}$, где a_{71} - “FuzzyRuleID” уникальный идентификатор, a_{72} - “Description” описание правила.
- b_8 - отношение “KnowledgeMatrix”, в котором $A_8 = \{a_{81}, a_{82}, a_{83}, a_{84}, a_{85}\}$, где a_{81} - “FuzzyRuleID”, a_{82} - “LinguisticVariableID”, a_{83} - “TermID”, a_{84} - “RuleForCreateTermNameID”, a_{85} - “IsResultOfRule” признак того, что это следствие правила. $a_{81}, a_{82}, a_{83}, a_{84}$ - образуют составной ключ.

Для описания связей приведем графическое представление предлагаемой схемы БД (Рис. 1).

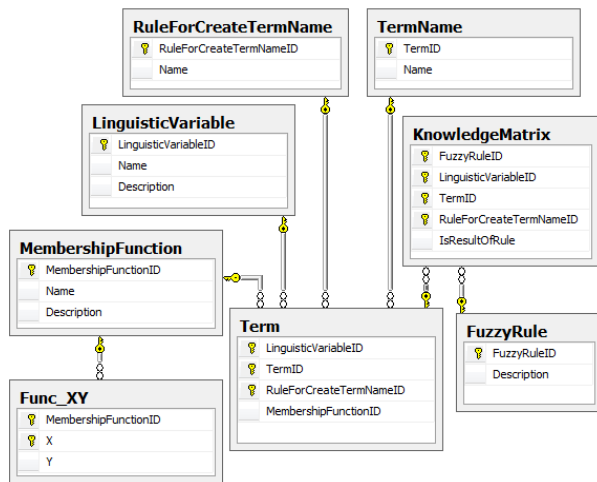


Рисунок 1 - Схема БД для хранения БНЗ

Правила и ограничения, которые обеспечивают целостность БД и описывают особенности представления знаний в БНЗ, определенные на этапе концептуального проектирования и описанные выше, могут быть заданы в виде триггеров в СУБД, которые будут срабатывать при добавлении и изменении информации.

ЗАКЛЮЧЕНИЕ

Предложена схема хранения БНЗ в реляционной БД. Данный подход позволяет использовать все преимущества хранения, и работы с информацией предоставляемые реляционными БД, при этом работать со знаниями которые предоставляются в виде БНЗ. За счет хранения знаний в реляционной модели можно извлекать только необходимые знания для построения логического вывода. Повысить эффективность получения знаний, используя предложенную схему БД возможно введением индексации, что позволит ускорить выполнение запросов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Глоба и др., 2008] Глоба, Л. С. Створення баз нечітких знань для інтелектуальних систем управління / Л. С. Глоба, М. Ю. Терновий, О. С. Штогріна // Комп'ютеринг. - Міжнародний науково-технічний журнал. - том 7, випуск 1. - Тернопіль, «Економічна думка» - 2008. - С.70-79.
- [Ротштейн, 1999] Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети / А. П. Ротштейн // Винница: УНИВЕРСУМ - 1999. - 320 с.
- [Коннолли и др., 2003] Коннолли Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг, А. Страчан // М.: Издательский дом «Вильямс», 3-е изд.: Пер. с англ.: Уч. пос. 2003.- 1440с.
- [Братко, 2004] Братко И. Алгоритмы искусственного интеллекта на языке PROLOG / И. Братко // М.: Издательский дом «Вильямс» - 2004. - 640 с.
- [Субботин, 2008] Субботин С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень / С. О. Субботін // Навчальний посібник. — Запоріжжя: ЗНТУ, 2008. — 341 с.
- [Zhang et al., 2011] Zhang X. A knowledge-based approach for answering fuzzy queries in XML / X. Zhang, X. Meng, X. Wang // Seventh International Conference on Natural Computation (ICNC). - 2011. - pp. 18-22.

[KML] KML (Knowledge Management Tools) [Электронный ресурс]. - Электрон. текстовые данные. - Режим доступа: <http://kml.mipt.ru/A/ru/bin/view/Home/KML2Specification>

APPROACH FOR STORING FUZZY KNOWLEDGE BASES

Globa L.S., Ternovoy M.Y., Shtogrina O.S.

National Technical University of Ukraine "Kyiv Polytechnic Institute", Kiev, Ukraine

lgloba@its.kpi.ua

ternovoy@its.kpi.ua

L_Shtogrina@mail.ru

The paper proposes the approach for fuzzy knowledgebase storing in the relational database. The design process of database scheme is described.

INTRODUCTION

Based on knowledgebase approaches allow to integrate expert knowledge and use them in computer-aided mode. Fuzzy knowledgebases can be highlighted from another knowledgebases as type of knowledgebases which contain fuzzy rules obtained from expert. There are some formats for knowledgebase storing such as text files, xml-based documents etc. The disadvantages of mentioned formats are complexity of consistency maintenance and absence of data access unified mechanism. The solution of the described problem is to use relational databases for knowledgebase storing.

MAIN PART

Fuzzy knowledgebase can be described as union of partial fuzzy knowledgebases each of which contains rules which defined one linguistic variable in the form IF-AND-OR-THEN.

The problem statement can be defined as follows: to obtain relational database scheme if we know the mathematical model of fuzzy knowledgebase.

The design process consists of conceptual and logical steps. At the end of this process relational database scheme is obtained (Fig. 1). There are also defined several constraints which can help to maintain integrity and specific features of fuzzy knowledge storing.

CONCLUSION

Proposed approach allows to increase efficiency of using fuzzy knowledge in information-telecommunication environment because of using unified methods to store and retrieve information which are given by the relational databases. The velocity of knowledge retrieval can be additionally increased by using special type of indexing.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.738.5

ДЕТЕКТИРОВАНИЕ ИЗМЕНЕНИЙ В INTERNET-ДОКУМЕНТАХ

Молчанов Ю. Н., Глоба Л. С., Алексеев Н. А.

*Национальный технический университет Украины «Киевский политехнический институт»,
г. Киев, Украина*

molchanov_y@ukr.net

lgloba@its.kpi.ua

n_alexeyev@hotmail.com

В работе рассмотрена проблематика отслеживания изменений информации в сети Internet, предложен алгоритм детектирования изменений в древовидных структурах данных, обеспечивающий корректное отображение изменений при меньшей временной сложности, по сравнению с существующими методами, а также архитектура системы публикации/подписки, реализующая данный метод.

Ключевые слова: детектирование изменений, задача «хорошего соответствия», системы публикации/подписки, XML.

1. Введение

Одной из особенностей среды Internet в настоящее время является то, что любой человек, имеющий доступ к Internet, может выступать в роли как публикующего, так и подписчика информации. Такой широкий круг публикующих информацию приводит к неконтролируемому постоянному обновлению информации в среде Internet, что вызывает проблему отслеживания изменений по интересующим темам. Традиционный подход, основанный на проверке актуальности данных через определенные промежутки времени приводит к усложнению программно-аппаратных комплексов отслеживания изменений, повышенной нагрузке на ресурсы, предоставляющие информацию, и не всегда приводит к своевременному обнаружению изменений. Наибольшее влияние этих недостатков наблюдается в системах, которые критичны ко времени задержки в обновлении информации.

Данная проблема рассматривается в рамках разработки систем публикации/подписки [Eugster, 2003], которые обеспечивают разделение во времени, пространстве и синхронизации для публикующего и подписчика. Данные системы публикации/подписки разделяются на темо-ориентированные системы [Ramasubramanian et al., 2006], событие-ориентированные [Caroguscio, 2002] и типо-ориентированные. Данные системы имеют ряд недостатков: высокая временная сложность решения задачи детектирования изменений в документах, отсутствие поддержки структурированных документов, которые являются

стандартом представления данных для среды Internet, некорректное или неполное определение изменений, произошедших в документе.

В данной работе предлагается новый алгоритм детектирования изменений в древовидных документах стандарта XML, который обеспечивает корректное отображение изменений при меньшей временной сложности, по сравнению с существующими методами.

2. Детектирование изменений в XML документах

2.1. Проблема детектирования изменений в XML документах

Определение изменений в документах является основной задачей разрабатываемых систем публикации/подписки. Большинство работ по детектированию изменений в XML документах затрагивают вопросы алгоритмической сложности операции вычисления разницы между документами, как для упорядоченных так и для неупорядоченных деревьев [Khanna et al., 2007], оптимизацией последовательности редактирования (также называемых дельты) - количества элементарных операций редактирования (добавление, удаление, обновление, перестановка узла), модели дельт и использование операции детектирования изменений в архитектурах хранения (обычно в базе данных) [Martinez et al., 2002]. В работе [Ronnau et al., 2008] авторы уделяют большое внимание проблеме алгоритмической и скоростной производительности новых diff алгоритмов, используя подход сверху-

вниз (а также предварительно вычисленные хеш коды). Другим подходом, предложенным сообществом разработчиков баз данных является исследование модели детектирования изменений, основанной на так называемых полных дельтах со строгими свойствами, среди которых обратимость и сокращение композиции дельт [Khandagale et al., 2010]. Тем не менее, механизм обозначения полностью зависит от поддержки постоянных идентификаторов каждого узла дерева. Можно сделать вывод, что большинство посвященных данной проблематике работ сосредоточены на производительности, временной и пространственной сложности операции детектирования изменений, а также оптимальности сгенерированных дельт, однако инструментарий и модели, позволяющим использовать все это эффективным образом, уделяется слишком мало внимания. С практической точки зрения не много работ рассматривают необходимость и прикладное значение абстрагирования от дельт и связанных моделей операций.

В данной работе представлен метод определения изменений между структурированными документами стандарта XML, который рассматривает проблему детектирования изменений в условиях неопределенной схемы XML-документов. Данный метод не производит поиск последовательности редактирования, так как его основной задачей является корректное отображение произведенных изменений, а не преобразование исходного документа в измененный. Таким образом, основной задачей предложенного алгоритма детектирования изменений является поиск «хорошего соответствия» (англ. good matching problem) между узлами XML документов.

2.2. Определение соответствия между узлами XML документов

Для эффективного детектирования изменений в определенной части веб-страницы, необходимо вначале найти соответствующую часть в новом документе. Так как в новой версии документа изменения могут вноситься в любой его части, то расположение интересующего подписчика текста может измениться. Поэтому вначале необходимо найти такую часть в новом документе, которая бы наиболее соответствовала тексту в старой версии.

В предыдущих работах [Chawathe et al., 1997], [Chawathe et al., 1998] предусматривалось наличие «хорошего соответствия» между узлами в исследуемом документе, поиск которого является сложной задачей для неупорядоченного XML документа. Здесь под понятием «хорошего соответствия» мы понимаем такое соответствие между частями старой и новой версий документов, когда преобразование старой версии документа в новую требует минимального количества элементарных операций – вставки, удаления, перемещения узлов, обновления содержимого узлов. Еще одним направлением для поиска

соответствующих частей в старой и новой версии документа является нахождения определенной меры похожести между частями документа, которая будет учитывать содержание, расположение и форматирование данной части документа. В работе [Flesca et al., 2009] был предложен набор параметров для определения меры похожести частей HTML документа. На основе идеи, изложенной в [Flesca et al., 2009], в данной работе были введены следующие параметры для поиска соответствия между частями XML документа: параметр соответствия содержимого, параметр соответствия атрибутов, параметр соответствия положения, а также интегральный критерий соответствия. Данный способ позволяет максимально учесть все возможные характеристики отдельного узла XML документа и найти «хорошее соответствие».

Другим важным вопросом является вопрос о преобразовании XML документа в XML дерево с определенными характеристиками. Такое преобразование впервые было представлено в [Barnard et al., 1995]. В предложенном методе также используется данное преобразование. Полученное XML дерево считается не упорядоченным, у которого отсутствуют метки, которые могли бы идентифицировать однозначно каждый узел. Между старой и новой версией документа возможны как структурные изменения, так и изменения содержимого узлов. Кроме того, были поставлены следующие условия при решении задачи детектирования изменений в XML документах:

- XML документ представляется в виде упорядоченного дерева, в котором учитывается порядок узлов слева направо, где узлы дерева – тэги XML, метки узлов дерева, обозначенные как $l(a_i)$ – названия тэгов XML при условии, что a_i - i -й узел дерева, значение узлов $con(a_i)$ - текстовое содержимое i -го узла, связи между узлами в иерархии XML. Если тэг a_i включает в себя тэг a_j , то узел a_i является родительским узлом узла a_j ;
- поиск соответствия производится только для листьев XML документа, значение которых содержит текстовую информацию, наиболее важную для подписчика;
- при сравнении XML тэгов атрибуты считаются совпадающими только при совпадении их названий и их значений;
- при поиске учитывается очередность дочерних узлов для любого родительского узла XML дерева, а также расположение узла в иерархии XML дерева на основании, введенного в данной работе, индексирования XML тэгов.

Данные условия были сформулированы, исходя из требований пользователя при поиске изменений информации в среде Internet.

Для реализации функции учета порядка узлов слева направо было принято решение о введении индексации в документе XML на основании числовых значений. Данное решение позволит учесть порядок узлов слева направо, при этом

сохранится возможность отслеживания местоположения узла в иерархии дерева XML. Это означает, что для каждого тэга в рассматриваемом документе при обработке будет введен атрибут *index*, которому будет присваиваться значение по разработанным правилам.

Далее соответствие между узлами XML деревьев будет определяться на основе числовых параметров. Данное решение было принято для уменьшения вычислительной сложности задачи и упрощения его практической реализации.

Определение 1. Пусть версия документа XML будет представлена деревом $T_1 = T_1(a_i, N, R, AT, con(a_i))$. Тогда дерево $T_1(a_i, N, R, AT, con(a_i))$ характеризуется следующими параметрами:

N – количество узлов в дереве T_1 , что соответствует количеству тэгов в XML документе;

$R = \{r_i | i = 1..m\}$ – совокупность родительских узлов в дереве T_1 , где m – количество родительских узлов, r_i – родительский узел i -го узла.

$AT = \{at_i | i = 1..N\}$ – совокупность атрибутов узлов в дереве T_1 , at_i – атрибут i -го узла.

$con(a_i)$ – содержимое i -го, где a_i – i -й узел дерева T_1 , а $A \in \{a_n\}$ – множество всех узлов заданного дерева.

Определение 2. Для заданного узла a_n документа T обозначим $T(a_n)$ – поддереву дерева T с корнем в узле a_n . При этом $T \in (a_n)$.

Определение 3. Узел документа T может быть представлен в виде множества $a_i(con(a_i), att(a_i), index(a_i))$, где параметр $con(a_i)$ – характеризует содержимое узла a_i ,

$att(a_i)$ – совокупность атрибутов узла a_i ,

$index(a_i)$ – параметр, характеризующий положение узла a_i в структуре дерева T , который определяется при индексировании XML документа

Для узла дерева T , рассматриваемого в данной работе, все родительские узлы будут иметь $con(a_i) = 0$, так как полезную для пользователя информацию несут в себе «листья» дерева.

Определение 4. Введем параметр соответствия содержимого узлов a_1 и a_2 следующим образом:

$$P_{con}(a_1, a_2) = \frac{|con(a_1) \cap con(a_2)|}{|con(a_1) \cup con(a_2)|} \quad (1)$$

Параметр $P_{con}(a_1, a_2)$ показывает отношение количества совпадающих слов в содержимом конечных узлов a_1 и a_2 к общему количеству слов в содержимом узлов a_1 и a_2 .

Определение 5. Параметр соответствия атрибутов можно определить между узлами a_1 и a_2 так:

$$P_{att}(a_1, a_2) = \frac{\sum at_i \in \{at(a_1) \cap at(a_2)\}}{\sum at_i \in \{at(a_1) \cup at(a_2)\}} \quad (2)$$

Параметр $P_{att}(a_1, a_2)$ показывает отношение количества одинаковых атрибутов в двух узлах a_1 и a_2 к максимальному количеству атрибутов, которые содержатся в этих узлах.

Определение 6. Параметр соответствия расположения может быть вычислен по следующей формуле:

$$P_{dist}(a_1, a_2) = \frac{suf(index(a_1), index(a_2))}{\max(index(a_1), index(a_2))} \quad (3)$$

В заданной формуле *suf* показывает длину общего суффикса между атрибутами узлов a_1 и a_2 , которые определяют местоположение узла в иерархии дерева XML – между $index(a_1)$ и $index(a_2)$, а *max* определяет максимальную длину атрибута между $index(a_1)$ и $index(a_2)$.

На основе указанных параметров совпадения значения, атрибутов и положения необходимо вывести интегральный критерий соответствия двух узлов в дереве.

Для учета всех трех параметров необходимо ввести весовые коэффициенты их значений в конечной формуле интегрального критерия соответствия.

Определение 7. Пусть α, β, γ будут весовыми коэффициентами соответственно для $P_{con}(a_1, a_2)$, $P_{att}(a_1, a_2)$, $P_{dist}(a_1, a_2)$.

Тогда $\alpha + \beta + \gamma = 1$, а универсальный критерий соответствия примет вид:

$$CS(a_1, a_2) = -1 + 2 \cdot (\alpha \cdot P_{con}(a_1, a_2) + \beta \cdot P_{att}(a_1, a_2) + \gamma \cdot P_{dist}(a_1, a_2)) \quad (3)$$

Очевидно, что универсальный критерий соответствия может принимать значения $[-1, 1]$, где -1 соответствует минимальной схожести, а 1 соответствует максимальному совпадению.

Пусть A_1, A_2 – текстовые узлы старой версии XML документа (дерево T_1), B_1, B_2, B_3, B_4 – текстовые узлы новой версии XML документа (дерево T_2). На основании заданных условий можно сказать, что из старой версии XML документа были удалены две текстовые части, а оставшиеся части – изменены. Необходимо найти какие части были удалены, а какие изменены и как. Для этого нужно найти, какие из A_1, A_2 соответствуют, B_1, B_2, B_3, B_4 .

Вначале найти интегральные критерии соответствия для каждой пары узлов. Для узлов A_1 и B_1 :

$$CS(A_1, B_1) = -1 + 2 \cdot (\alpha \cdot P_{con}(A_1, B_1) + \beta \cdot P_{att}(A_1, B_1) + \gamma \cdot P_{dist}(A_1, B_1)) \quad (4)$$

Аналогично производится расчет для всех пар узлов. Для данного случая матрица критериев интегрального соответствия будет иметь вид, представленный в табл. 1.

Таблица 1 – Матрица интегральных критериев соответствия

	B_1	B_2	B_3	B_4
A_1	$CS(A_1, B_1)$	$CS(A_1, B_2)$	$CS(A_1, B_3)$	$CS(A_2, B_4)$
A_2	$CS(A_2, B_1)$	$CS(A_2, B_2)$	$CS(A_2, B_3)$	$CS(A_2, B_4)$

2.3. Математическая модель задачи поиска хорошего соответствия

Объектом данной задачи является определение «хорошего соответствия» между узлами документов. Целью задачи является исследование максимально возможной величины соответствия между деревьями, которая равна сумме интегральных критериев соответствия каждой из пар соответствующих узлов.

Для формализации данной задачи введем матрицу связности между узлами деревьев T_1 , что соответствует старой версии документа и T_2 , что соответствует новой версии.

Определение 8. Под параметром x_{ij} будем понимать связность узла A_i и B_j . Если A_i соответствует B_j , то $x_{ij}=1$ Если A_i не соответствует B_j , то $x_{ij}=0$.

Если для каждой пары узлов в старом и новом дереве назначить данный параметр связности, то данные величины могут быть записаны в виде матрицы, представленной в таблице 1.

Таблица 1 – Матрица связности старой и новой версии XML документа

	B_1	B_2	B_3	B_4
A_1	x_{11}	x_{12}	x_{13}	x_{14}
A_2	x_{21}	x_{22}	x_{23}	x_{24}

Введенный параметр связности будет однозначно определять, соответствует ли данная пара узлов друг другу или нет, то есть – одно из возможных состояний исследуемого объекта.

Найдем область допустимых решений задачи. С учетом условий, которые накладываются задачей поиска хорошего соответствия между узлами на решение и критерия оптимальности для задачи поиска «хорошего соответствия» в виде целевой функции, которая основана на матрице интегральных критериев соответствия, получаем следующую задачу булевого линейного программирования:

$$\begin{aligned}
 a_{11}x_{11} + a_{12}x_{12} + a_{13}x_{13} + a_{14}x_{14} &= b_1 \\
 a_{25}x_{21} + a_{26}x_{22} + a_{27}x_{23} + a_{28}x_{24} &= b_2 \\
 a_{31}x_{11} + a_{35}x_{21} &\leq b_3 \\
 a_{42}x_{12} + a_{46}x_{22} &\leq b_4 \\
 a_{53}x_{13} + a_{57}x_{23} &\leq b_5 \\
 a_{64}x_{14} + a_{68}x_{24} &\leq b_6
 \end{aligned} \tag{5}$$

Или в сокращенной записи $Ax_{ij} \leq B$, где матрица условий имеет значение:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

а матрица B имеет вид:

$$B = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{7}$$

Решение задачи состоит в нахождение такого решения (5), которое максимизирует линейную функцию:

$$\langle C, X \rangle = c_{11} \cdot x_{11} + c_{12} \cdot x_{12} + \dots + c_{24} \cdot x_{24} \tag{8}$$

где $c_{11}=CS(A_1, B_1)$, $c_{12}=CS(A_1, B_2)$, $c_{13}=CS(A_1, B_3)$, $c_{14}=CS(A_1, B_4)$, $c_{21}=CS(A_2, B_1)$, $c_{22}=CS(A_2, B_2)$, $c_{23}=CS(A_2, B_3)$, $c_{24}=CS(A_2, B_4)$.

Если рассматривать общий случай задачи и считать, что n - количество тэгов в исходном дереве A , а m -количество тэгов в обновленном дереве B . Не нарушая общности, будем считать, что $n < m$, тогда задача примет вид:

$$\begin{aligned}
 \langle C, X \rangle &= c_{11} \cdot x_{11} + c_{12} \cdot x_{12} + \dots + c_{(n-1)m} \cdot x_{(n-1)m} + \\
 &+ c_{nm} \cdot x_{nm} \\
 a_{11}x_{11} + a_{12}x_{12} + \dots + a_{1m}x_{1m} &= b_1 \\
 a_{n1}x_{n1} + a_{n2}x_{n2} + \dots + a_{nm}x_{nm} &= b_n \\
 a_{(n+1)1}x_{11} + a_{35}x_{21} + \dots + a_{(n+1)(m+1)}x_{n1} &\leq b_n \\
 a_{(n+m)1}x_{1m} + \dots + a_{(n+m)(n+m)}x_{nm} &\leq b_{(n+m)}
 \end{aligned} \tag{9}$$

Максимизация производится по всем булевым значениям величины x , удовлетворяющим системе (9):

$$\max_{x \in B^n: Ax \leq B} \langle C, X \rangle \tag{10}$$

3. Решение задачи поиска «хорошего соответствия» между узлами XML документов

В общем случае, представленная задача поиска «хорошего соответствия» является задачей булевого линейного программирования. Из этого следует, что решение задачи (10) имеет NP-сложность. Таким образом, правомерным является

использование переборных схем для решения такой задачи.

3.1. Неточные методы решения задачи булевого линейного программирования

Так как в задаче булевого линейного программирования, поставленной в рамках алгоритма детектирования изменений, количество листьев в каждом из деревьев может быть произвольным, то точные методы решения приведут к большой временной сложности в решении алгоритма. Для задач с большой размерностью могут быть использованы неточные методы решения, которые дают приближенное (субоптимальное) решение, но имеют меньшую временную сложность.

Более того, мощное направление в развитии приближенных подходов естественным образом возникло внутри точных методов (в основном методов ветвей и границ). При этом неоднократно отмечалось, что для значительного большинства прикладных задач совершенно достаточно вместо точного получить хорошее приближенное решение [Финкельштейн, 1976].

В данной работе предлагается использовать метод локальной оптимизации, описанный в [25]. Идея локальной оптимизации достаточно проста и используется во многих приближенных алгоритмах. Рассматривается задача дискретного программирования:

$$\begin{aligned} f(X) \rightarrow \max \\ X \in D \end{aligned} \quad (11)$$

Предполагается, что для каждого X из D определена окрестность $G(X)$, причем $X \in G(X)$. Тогда метод локальной оптимизации состоит в следующем.

Шаг 1. Находим $X^k \in D$. Переходим к шагу 2.

Шаг 2. Если X^k найдено, то перебором точек X^k находим X^{k+1} из условия $f(X^k) = \max \{f(X) | X \in G(X^k)\}$. Если $f(X^k) \geq f(X^{k+1})$, то $f(X^k)$ — локальный оптимум. Если же $f(X^k) < f(X^{k+1})$, то заменяем X^k на X^{k+1} и переходим к шагу 2.

Так как шаг 2 включает полный перебор точек $G(X^k)$, то понятие окрестности следует вводить таким образом, чтобы для каждого X величина $|G(X)|$ была достаточно малой 3.

Если множество D — конечно, то, очевидно, метод локальной оптимизации конечен.

При практической реализации очень важно разумным способом определить понятие окрестности. Если окрестности будут слишком малы, то процесс будет малоэффективен. Если же

они окажутся слишком велики, то станет труден (или невозможен) сам акт перебора на шаге 2.

3.2. Применение метода локальной оптимизации для решения задачи поиска «хорошего соответствия»

Рассмотрим метод локальной оптимизации применимо к задаче (9) и запишем матрицу C в следующем виде.

$$C = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \dots & \dots & \dots & \dots \\ c_{k1} & c_{k2} & \dots & c_{km} \\ c_{n1} & c_{n2} & \dots & c_{nm} \end{pmatrix}. \quad (12)$$

На шаге 1 найдем допустимое решение для задачи БЛП. Для этого необходимо выполнить следующие действия.

Найти максимальное значение среди элементов матрицы C . Пусть максимальным элементом будет c_{k2} , тогда присвоим $x_{k2}=1$. Из этого следует, что в матрице X , $x_{ki}=0 | i \in [1,2) \cup (2,n]$, $x_{j2}=0 | j \in [1,k) \cup (k,n]$. Так как данные величины в матрице X найдены, то в матрице C необходимо уменьшить область поиска максимальных значений, убрав элементы матрицы C , не находящимися во втором столбце и k -й строчке. Данные действия необходимо повторять, пока не будут найдены все элементы X^k .

На основании указанных действий сформулируем алгоритм для шага 1:

1. Поиск максимального элемента матрицы C - элемент $c_{kl} = \max(C^* | C^* \subset \{c_{11} \dots c_{nm}\})$.

2. Присвоить $x_{kl}=1$. Из этого следует, что $x_{ki}=0 | i \in [1,l) \cup (l,n]$, $x_{j2}=0 | j \in [1,k) \cup (k,n]$. Если не получен X^k , то ограничить зону поиска максимального значения $C^* \subset \{c_{11} \dots c_{nm}\} \cap \{c_{k1} \dots c_{km}\} \cup \{c_{11} \dots c_{n1}\}$ и повторить шаг 1.

В конце работы алгоритма будет получено решение X^k . Данное решение на шаге 2 необходимо локально оптимизировать на окрестности $G(X^k)$.

После проверки, осуществляемой на шаге 2, будет получен оптимальный результат.

Сформулируем алгоритм для шага 2:

1. Для всех столбцов $l|l=2\dots n-m$, получаем значения $X_1^{k+1}\dots X_{l-1}^{k+1}$ с помощью перестановки строчек, содержащих 1 попарно для столбцов $1\dots(l-1)$ и l .

2. Находим значение $f(X^{k+1})$ для всех $X_1^{k+1}\dots X_{l-1}^{k+1}$. Если $f(X^k)\geq f(X^{k+1})$, то $f(X^k)$ – локальный оптимум. Если же $f(X^k)<f(X^{k+1})$, то заменяем X^k на X^{k+1} и переходим к шагу 1.

После работы алгоритма будет получено оптимальное решение задачи БЛП X^* .

3.3. Временная сложность задачи поиска «хорошего соответствия» для узлов XML документов

Оценим число временную сложность решения заданной задачи (10) методом локальной оптимизации.

Ранее было принято без потери общности, что $n<m$. Пусть n – количество строк в матрицах C и X .

В алгоритме первого шага локальной оптимизации необходимо найти максимальное значение в каждой строке матрицы C . При этом на каждом шаге, количество возможных вариантов будет уменьшаться на 1.

Таким образом, сложность задачи первого шага z_1 будет составлять разность двух арифметических прогрессий S_m – суммы всех возможных вариантов перебора от 1 до m и S_{m-n} – суммы неиспользуемых вариантов от 1 до $(m-n)$:

$$R(z_1) = S_m - S_{m-n} = \frac{1+m}{2} \cdot m - \frac{1+(m-n)}{2} \cdot (m-n) = \frac{n^2 + 2nm + n}{2} \quad (13)$$

Временная сложность второго шага обусловлена количеством перестановок и составляет:

$$R(z_2) = S_{m-n-1} = \frac{1+(m-n-1)}{2} \cdot (m-n-1) = \frac{m^2 - 2nm - m + n + n^2}{2} \quad (14)$$

Таким образом, общая временная сложность решения задачи (10) методом линейной оптимизации составляет:

$$R(z) = R(z_1) + R(z_2) = \frac{n^2 + 2nm + n + m^2 - 2nm - m + n + n^2}{2} = \frac{2n^2 + 2n - m + m^2}{2} \quad (15)$$

$$R(z) \propto (n^2 + m^2). \quad (16)$$

Предложенный алгоритм линейной оптимизации относится к неточным методам решения задачи БЛП, но он обеспечит нахождение локального максимума для задачи (10), который в большинстве случаев совпадает с глобальным максимумом, обладая при этом более низкой временной сложностью $O(n^2+m^2)$ по сравнению с существующими методами.

3.4. Сравнение эффективности алгоритмов поиска соответствия между частями XML документа

В данной работе представлен новый подход к решению задачи поиска соответствия между XML документами. Было показано, что данный подход позволяет рассматривать эту задачу как задачу булевого линейного программирования. Для решения этой задачи было предложено использовать комбинаторный метод, сложность которого для данной задачи была определена как $O(n^2+m^2)$, где n и m – количество листьев в старом и новом деревьях.

Для показания целесообразности такого подхода необходимо сравнить временную сложность поставленной задачи в данном подходе с временной сложностью задач существующих алгоритмов.

Для этого сравним полученный алгоритм со всеми типами алгоритмов, представленных в обзоре: алгоритмы, основанные на поиске минимальной последовательности редактирования, семантические алгоритмы и алгоритмы, основанные на хеш подписях.

Алгоритм Чанга и Шаша рассматривает два дерева для нахождения минимальной последовательности редактирования. Данный алгоритм применим для XML документов. Минимальная временная сложность данного алгоритма определяется, как $O(p^2)$ [Caroguscio, 2002], где $p=n+m$ – общее количество узлов в двух деревьях. Тогда сложность алгоритма равна: $O((n+m)^2)$.

Семантические алгоритмы рассматривают XML документ как структуру XML тэгов. При сравнении учитывается иерархия синтаксиса языка и его закономерности. С помощью данного алгоритма можно определить только перемещение узлов, и невозможно определить удаление и вставку узла. Поэтому данный алгоритм не всегда находит корректное соответствие между узлами XML документов. Сложность данного подхода [Seung-Jin et al., 2001]: $O(|X| \cdot |Y| \cdot (|X| \log |X| + |Y| \log |Y|))$, где X и Y – количество ветвей в семантической иерархии каждого из деревьев. Так как количество ветвей иерархии можно принять равной количеством листьев в каждом из деревьев, то сложность данного метода можно представить следующим образом: $O(n \cdot m \cdot (n \log(n) + m \log(m)))$.

В системе детектирования веб-страниц Коури рассматривается схожий данной работой подход. Данный алгоритм рассматривает задачу нахождения соответствия между XML деревьями как задачу венгерского алгоритма. Сложность данной задачи в [Khoury et al., 2007] определена как $O((n+m)^3)$, что гораздо больше, чем сложность предложенного алгоритма.

На Рис. 1. показана сравнительная характеристика рассмотренных методов сравнения XML документов при помощи различных подходов.

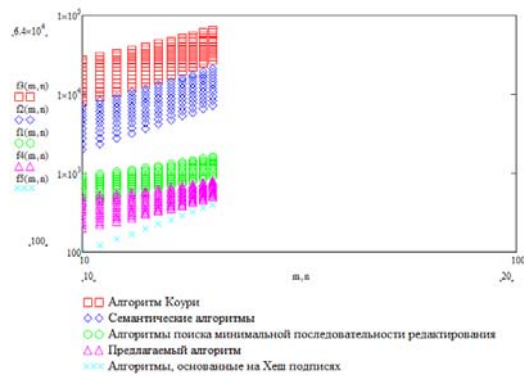


Рисунок 1 Сложность алгоритмов

4. Система публикации/подписки, основанная на методе детектирования изменений в XML документах

4.1. Обобщенная архитектура системы публикации/подписки

Здесь описывается общая архитектура системы публикации/подписки, определяющая основные модули системы и их функции. Система публикации/подписки должна разрабатываться для поддержки многопользовательских синхронных запросов мониторинга. Архитектура наследует клиент-серверную модель – клиентская часть, расположенная на клиентских машинах, обеспечивает интерфейс для общения пользователя с сервером приложения.

Архитектура системы является модульной, таким образом, позволяет изменять функционал или алгоритмы в модулях, не влияя на остальные модули. Модули системы можно условно разделить на 3 группы, что представлено на рис. 2: модули интерфейса, модули алгоритмов и модули хранения.



Рисунок 2 Классификация модулей системы

Далее будут представлены основные модули

системы публикации/подписки, которые реализуют предложенный метод детектирования изменений в XML документах: построитель XML дерева и компаратор.

4.2. Основные функции построителя XML дерева

Построитель дерева отвечает за преобразование XML страницы в общую древовидную структуру данных. Процесс построения дерева состоит из трех этапов, которые реализуются отдельными подмодулями: фильтром, редактором и конструктором дерева. Первые два подмодуля необходимы для адаптации предложенного метода для детектирования изменений в HTML страницах.

Фильтр выполняет удаление неупорядоченных тэгов, скриптов и комментариев из страницы HTML. Фильтр генерирует новую версию документа в отдельном файле, которая может использоваться для детектирования изменений.

Вторым шагом построения дерева является перегруппировка HTML документа для решения проблемы некорректно закрытых тегов с помощью подмодуля редактирования. Выходными данными данного процесса является визуально аналогичный код, который может быть представлен общим деревом. На следующем этапе проводится индексирование тегов по правилам, используемому в предложенном методе детектирования изменений в XML документах.

На последнем шаге генерируется общее дерево из обработанного HTML документа.

4.3. Основные функции модуля компаратора

Компаратор предусматривает выполнение основных шагов предложенного метода детектирования изменений по следующему алгоритму. На вход компаратора подаются два, дерева, сформированных в построителе дерева. Так как в алгоритме было поставлено условие о выборе и сравнении только наиболее значимых для подписчика узлов, то на первом шаге осуществляется поиск таких узлов.

На втором шаге на вход Вычислителя подаются набор наиболее значимых узлов с их содержимым и атрибутами. В Вычислителе для каждой пары узлов из обоих деревьев происходит вычисление их критериев соответствия и формируется матрица интегральных критериев соответствия. Результат работы вычислителя представляет собой таблицу значений интегральных критериев соответствия всех пар узлов сравниваемых деревьев.

На третьем шаге в Блоке решения задачи булевого линейного программирования (БЛП) происходит формирования задачи и ее ограничений, а также производится ее решения с помощью метода локальной оптимизации. На выходе данного шага формируется информация о том, какие узлы исходного дерева совпадают с узлами измененного дерева.

На четвертом шаге в блоке компаратора происходит сравнение содержимого совпадающих

узлов, определенных на третьем шаге. Полученная разность содержимого и определяет изменения в информации, которая была запрошена абонентом.

В формировщике конечного дерева на основании исходного дерева формируется новое дерево T' , в котором будут выделены изменения, определенные между двумя деревьями. Данное дерево будет передано на Построитель Инверсного дерева для дальнейшей обработки и представления информации в удобном виде для конечного пользователя.

ЗАКЛЮЧЕНИЕ

При сравнении предложенного метода решения задачи поиска соответствия между XML документами с другими алгоритмами было показано, что предложенный алгоритм является эффективным с точки зрения временной сложности. Это достигнуто доказательством возможности рассмотрения задачи поиска соответствия между XML документами как задачи булевого линейного программирования и ее эффективного решения с помощью метода локальной оптимизации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Финкельштейн, 1976] Финкельштейн, Ю.Ю., Приближенные методы и прикладные задачи дискретного программирования / Финкельштейн, Ю. // Москва: Наука. – 1976. – 265 с.
- [Barnard et al., 1995] Barnard, D.T. Tree-to-Tree Correction for Document Trees / Barnard, D.T., Larke, G., Duncan, M. // Technical report, Queen's Univ. Computer Science Dept. – 1995. – pp. 23-42.
- [Chawathe, et al., 1997] Chawathe, S. Meaningful change detection in structured data / Chawathe, S., Garcia-Molin: Proceedings of the ACM, SIGMOD International Conference on Management of Data, Tuscon, Arizona – 1997 - pp. 26–37.
- [Chawathe, et al., 1998] Chawathe, S. Representing and querying changes in semistructured Data /Chawathe, S., Abiteboul, S, Widom, // Proceedings of the International Conference on Data Engineering, Orlando, Florida. – 1998. – pp. 4–13.
- [Coruscio et al., 2002] Coruscio, M. Formal Analysis of Clients Mobility in the Siena Publish/Subscribe Middleware / Coruscio, M., Inverardi, P., Pelliccione, P. // Technical Report, Department of Computer Science, University of Colorado – 2002. - pp. 1-18.
- [Eugster et al., 2003] Eugster P. The many faces of Publish/Subscribe / Eugster P., Felber P., Guerraoui R., Kermarrec A.. - ACM Computing Surveys, Vol 35 – 2003.- № 2. – pp. 114-131.
- [Flesca et al., 2007] Flesca, E. Efficient and affective Web change Detection / Flesca, E., Masciari, S. // Proceedings of the International Conference on Data Engineering, Tuscon, Arizona – 2007. – pp. 4–13.
- [Khandagale et al., 2010] Khandagale, H.. A Novel Approach for Web Page Change Detection System / Khandagale, H. P., Halkarnikar, P. P. // International Journal of Computer Theory and Engineering – 2010.
- [Khoury et al., 2007] Khoury, F. An Efficient Web Page Change Detection System Based on an Optimized Hungarian Algorithm / Khoury, F., El-Mawas, R., El-Rawas, O., Mounayar, E., Artail, H. // IEEE Transactions on Knowledge and Data – Vol.19. – 2007. – pp. 48-59.
- [Lim et al., 2001] Lim, S. An Efficient Algorithm to Compute Differences between Structured Documents / Lim, S., Yiu-Kai, N. // Information theory, IEEE. – 2001. – pp. 171-180.
- [Martinez et al., 2002] Martinez, M. A method for the dynamic generation of virtual versions of evolving documents / Martinez, M., Dermame, J.-C., de la Fuente, P. // Proceedings of the 2002 ACM symposium on Applied computing, New York, NY, USA. – 2002. – pp. 476 - 482.
- [Ramasubramanian et al., 2006] Ramasubramanian, V. Corona: A High Performance Publish-Subscribe System for the World Wide

Web. / Ramasubramanian, V., Peterson, R., Sireer, E. // Proc. USENIX Symposium on Networked Systems Design and Implementation. – 2006. – pp. 15-28.

[Ronnau et al., 2008] Ronnau, S. Merging changes in xml documents using reliable context fingerprints / Ronnau, S., Pauli, C., Borgho, M. // Proceeding of the eighth ACM symposium on Document engineering, New York, NY, USA. - 2008. – pp.58-62.

[Wang et al., 2003] Wang, Y. A fast change detection algorithm for XML documents / Wang, Y., DeWitt, D., Cai. X-Di, J. // In International Conference on Data Engineering (ICDE'03). Citeseer. – 2003. – pp. 235-258.

CHANGE DETECTION OF INTERNET DOCUMENTS

Molchanov Y.N., Globa L. S., Alexeyev N. A.

*National Technical University of Ukraine «Kyiv Polytechnic Institute»,
Kiev, Ukraine*

molchanov_y@ukr.net

lgloba@its.kpi.ua

n_alexeyev@hotmail.com

Information change detection problem in Internet is considered in the work, also the change detection algorithm of data tree structures is proposed, which provides correct detected changes presentation with lower time coplexity in comparison with existing methods.

INTRODUCTION

Nowadays, wide range of publishers in Internet environment leads to uncontrolled regular updates of information which causes change detection problem in the interested subjects. Traditional approaches with periodic source check result in complicated hardware and software for changes monitoring, increased information resources load and not always timely and correct change detection.

In this paper new change detection algorithm for structured documents is proposed. The described drawbacks were taken into account during presented change detection algorithm development.

MAIN PART

The paper is structured as follows. In the first section we describe XML document change detection problem and present boolean linear programming task for search of good matching between tree nodes. In the second section we suggest a solution for boolean linear programming task and good matching problem. Change detection system architecture is presented in the third section.

CONCLUSION

By comparing existing change detection algorithms for structured documents, high performance and low time complexity of proposed algorithm was shown.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.82:004.55

WEB-ОРИЕНТИРОВАННАЯ РЕАЛИЗАЦИЯ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Колб Д.Г.

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

kolb@bsuir.by

Рассмотрены принципы и подходы к разработке семантических web-сайтов с использованием средств традиционных технологий построения web-сайтов. В основу предлагаемых подходов положены понятие sc-модели web-сайта и способ псевдоестественного представления таких моделей.

Ключевые слова: семантическая сеть, семантическая модель web-сайта, SC-код, интеллектуальная система.

ВВЕДЕНИЕ

Эффективная интерпретация различных моделей интеллектуальных систем является одной из ключевых проблем разработки интеллектуальных систем. Наиболее остро в рамках данного направления стоят проблемы разработки унифицированных подходов и инструментальных средств интерпретации моделей интеллектуальных систем на различные web-платформы.

Базовым структурным элементом при организации любой информационной системы (ИС) на web-платформе, как правило, является web-страница. Web-страница содержит информацию о некотором фрагменте предметной области, которой посвящен web-сайт или web-портал. При такой организации можно выделить уровни, которые содержит любая ИС: *уровень внешнего представления, уровень языков разметки, уровень серверных скриптов, уровень хранилищ данных*. Рассматривая в таком ракурсе ИС на базе web-платформы, мы приходим к заключению, что степень интеллектуальности ИС будет зависеть от степени внедрения на каждом из выделенных уровней ИС семантических технологий. Обозначим направления работ по интеллектуализации ИС на базе web-платформы в соответствии с выделенными выше уровнями:

- семантическая структуризация информации на уровне внешнего представления, которая необходима для обеспечения семантически-однозначной и понятной для читателя гипертекста навигации по web-пространству сайта;
- семантическая структуризация информации на уровне языков разметки, которая необходима для

обеспечения поисковых машин знаниями, необходимыми для поиска информации на множестве web-ресурсов web-пространства, и которая обеспечивает возможность ввода в интеллектуальную систему на базе web-платформы новых знаний;

- разработка на уровне серверных скриптов операций семантического поиска и навигации по информационному web-пространству сайта для обеспечения пользователей средствами поиска более эффективными, чем обычная навигация, и для обеспечения пользователей средствами решения задач, которые традиционно относят к задачам искусственного интеллекта;
- использование на уровне хранилищ данных хранилищ знаний вместо баз данных, для того, чтобы обеспечить представления информации о предметной области, к которой относится web-сайт, с помощью моделей представления знаний, которые используются в искусственном интеллекте. Такая организация хранилищ знаний позволит эффективно реализовать семантический поиск, а также обеспечит платформу для решения различных задач в рамках предметной области для средств уровня серверных скриптов.

Для обеспечения возможности семантической структуризации в рамках предлагаемого подхода будем рассматривать сущности web-страницы как элементы базы знаний (БЗ). Такой подход позволяет трактовать ИС на базе web-платформы как специализированную интеллектуальную систему, решающую задачу организации диалога человека и предметной интеллектуальной системы, и обеспечивающую решение основных задач предметной интеллектуальной системы. Рассмотрим один из возможных подходов к

решению указанных проблем, который используется в рамках проекта OSTIS.

В качестве формальной основы предлагаемого подхода будем использовать семантические сети с базовой теоретико-множественной интерпретацией. Основным способом кодирования информации для таких сетей является SC-код (Semantic Code) [OSTISa, 2011]. Интеллектуальные системы, построенные с использованием SC-кода, будем называть sc-системами. В основе любой sc-системы лежит понятие некоторой прикладной sc-модели – модели предметной области, закодированной с помощью SC-кода. Таким образом, задача интерпретации некоторой модели интеллектуальной системы на некоторую прикладную (программную или аппаратную) платформу в рамках проекта OSTIS сводится к интерпретации sc-модели этой интеллектуальной системы на данную платформу.

Разработка интеллектуальной системы, при использовании предложенного подхода будет проходить в следующем порядке:

- разработка формального описания предметной области, по которой разрабатывается интеллектуальная система (sc-модели);
- выделение сущностей предметной области, информация о которых будет размещаться на web-страницах;
- разработка операций обработки sc-модели;
- представление информации о выделенных сущностях в формальном виде и погружение этой информации в некоторое хранилище знаний на базе web-платформы;
- тестирование и отладка sc-системы.

Отметим, что приведенные этапы разработки предусматривают реализацию sc-систем с различной функциональностью, что, как правило, обуславливается спецификой конкретной интеллектуальной системы и определяет, один из возможных способов интерпретации этой sc-системы на web-платформу. Например, может быть несколько различных интерпретаций sc-системы, если она принадлежит к классу информационно-справочных систем (к этому классу относится большинство традиционных сайтов). Приведем некоторые из интерпретаций:

- интерпретация sc-системы на базе уровня внешнего представления, которая предполагает ручную семантическую навигацию по web-пространству;
- интерпретация sc-системы на базе уровня внешнего представления и уровня языков разметки, которая предполагает как ручную семантическую навигацию по web-пространству, так навигацию с помощью различных средств семантического поиска;
- интерпретация sc-системы на базе уровня внешнего представления, уровня языков разметки и уровня серверных скриптов, которая предполагает использование функций, доступных на двух предыдущих уровнях и возможность

решения задач в рамках предметной области, по которой разработана интеллектуальная система.

Реализация предлагаемого подхода на практике предполагает решение следующего перечня задач:

- разработка языковых средств представления sc-моделей интеллектуальных систем в рамках некоторой прикладной web-платформы, учитывающих особенности представления мультимедиа данных;
- разработка программных средств, поддерживающих реализацию операций обработки sc-моделей;
- разработка программных средств, обеспечивающих эффективное хранение sc-моделей в рамках некоторого высокопроизводительного хранилища данных.
- разработки средств управления жизненным циклом sc-системы, для эффективной поддержки sc-системы на всех этапах её функционирования.

Рассмотрим решение этих задач подробнее.

1. Языковые средства представления sc-моделей интеллектуальных систем

1.1. Состояние работ в области языковых средств представления моделей интеллектуальных систем.

Уровень внешнего представления – это часть web-ресурса, которая видима пользователю. В настоящее время существует масса подходов к структуризации информации на этом уровне web-ресурса. Наиболее известными из них являются:

- структуризация в виде обычного текста;
- структуризация в виде линейного спискового представления;
- структуризация в виде иерархического спискового представления;
- структуризация в виде табличного представления;
- структуризация за счет выделения элементов различной значимости шрифтом, фоном или оформлением;
- структуризация на основе индуктивного подхода к организации web-ресурса [Microsoft, 2001].

Работ, связанных с семантической структуризацией внешнего представления web-ресурсов, в научной литературе практически нет. Однако, существует масса работ по смежной с данной тематикой, когнитике, в которых затрагиваются проблемы представления любых форм информации в рамках пользовательского интерфейса [Раскин, 2004].

Уровень языков разметки – это, в настоящий момент, наиболее проработанный уровень с научной и практической стороны в смысле структуризации и представления данных в рамках web-ресурса. Основные направления работ здесь связаны, с семантической структуризацией текстового и мультимедиа наполнения web-сайтов.

Первыми результатами в данной области можно назвать реализации для сети Internet формализованной модели гипертекста, основу которой составлял язык HTML [W3C, 2011]. Новой ветвью развития языков представления знаний для сети Internet стало направление Semantic web [W3C, 2011]. Однако, до появления этого направления уже существовали работы, которые были направлены на оптимизацию текущего представления знаний в виде HTML для сети Internet [Гаврилова, 2000].

Semantic web предполагает представление знаний в виде семантической сети с помощью онтологий. Основу технологий предлагаемых Semantic Web составляет семейство стандартов на языки описания, включающее XML, XML Schema, RDF, RDF Schema, OWL, OWL2 [Хорошевский, 2008]. Первые результаты работ в рамках направления Semantic web были использованы разработчиками пользовательских интерфейсов web-ресурсов для семантического размещения типовых сущностей пользовательского интерфейса или его структуры. Основу таких подходов составлял язык XML. В настоящее время разработкой языков описания пользовательского интерфейса и технологиями, поддерживающими такие языки, занимаются такие ведущие разработчики инструментальных средств для разработки программного обеспечения, как Microsoft (XAML, MRML), Adobe (MXML, OpenLaszlo), Oracle (CookSwing, SwiXML, SwixNG, Thinlet, Ultrid, Vexi, XALXAL, XSWT, ZUML), Mozilla (XUL).

Однако только семантическая разметка структуры страниц web-ресурса не давала возможность реализовать эффективные способы поиска информации в рамках web-ресурса. Одним из наиболее доступных по реализации подходов для решения данной проблемы стал подход, основанный на использовании микроформатов (microformats, μF или uF) – способа семантически размечать сведения о разнообразных сущностях на web-страницах, используя стандартные элементы языка HTML (или XHTML) [Вики о микроформатах, 2011].

В настоящее время микроформаты используются как для унифицированного представления однотипных сущностей пользовательского интерфейса web-приложения, так и для настройки и адаптации пользовательского интерфейса web-браузеров к пользователю [Веб-фрагмент, 2011].

Несмотря на простоту использования микроформатного подхода, он не позволял решить все задачи, которые ставило перед собой направление Semantic Web. Поэтому одновременно с микроформатным подходом развивались подходы, в основе которых лежит использование RDF. Ярким примером реализации подхода на базе RDF является проект FOAF ("Friend of a Friend") [FOAF, 2010], который позволяет описывать отношение знакомства с помощью RDF.

С ростом количества web-ресурсов поддерживающих стандарты Semantic web появилась необходимость унифицированного представления знаний для таких web-ресурсов. Решением такой проблемы является использование системы метаданных для представления знаний в web-ресурсах. В настоящее время существует несколько десятков проектов, связанных с разработкой систем метаданных. Одним из наиболее популярных проектов, направленных на решения проблемы унификации представления знаний в виде семантических сетей, стал проект «Дублинское ядро» [DCMI, 2011]. Целью проекта стала разработка стандартов метаданных, которые были бы независимы от платформ и подходили бы для широкого спектра задач. Основными результатами проекта являются словари метаданных общего назначения, стандартизирующие описание ресурсов с помощью различных RDF-форматов.

Несмотря на наличие серьёзных подходов к структуризации на уровне внешнего представления и уровне разметки многие проблемы остались до сих пор не решенными, а именно:

- отсутствуют средства семантической структуризации внешнего представления web-ресурсов;
- отсутствуют единые стандарты для унификации представления знаний (унифицированных языковых средств разметки). Помимо проекта «Дублинское ядро» в сети Internet широко используются ряд других систем метаданных, таких как GILS, MARC, ONIX, LOM, UDDI;
- отсутствуют унифицированные подходы к представлению мультимедиа данных;
- отсутствуют единые подходы к проектированию интеллектуальных систем, в основе которых лежат семантические сети для сети Internet.

1.2. Языковые средства представления sc-моделей интеллектуальных систем на уровне внешнего представления и уровне языков разметки

Основу уровня внешнего представления в предлагаемом подходе составляют *семантически структурированные гипертексты* – гипертексты, информация в которых будет отображаться помощью SCn-кода (способа псевдоестественного кодирования семантических сетей, представленных в SC-коде, Semantic Code natural). SCn-код задается множеством всех sc.n-статей, каждая из которых описывает семантическую окрестность некоторого понятия предметной области. Каждая статья в свою очередь состоит из идентификатора sc-элемента (объекта предметной области, закодированного с помощью SC-кода), описываемого в этой sc.n-статье, и, возможно, одного или нескольких последующих sc.n-полей. При описании sc-элемента в sc.n-статье sc.n-поля описывают как, какими ролями и связками каких отношений, связан описываемый sc-элемент с другими sc-элементами.

Ряд sc.n-полей может содержать мультимедиа или тексты логических утверждений. Мультимедиа может включать любые информационные конструкции, обозначаемые как внешние по отношению к SCn-коду, в том числе и sc.n-тексты.

Основу уровня языков разметки является SCnML (SCn Markup Language) – модель языковых средств разметки семантически структурированных гипертекстов. SCnML – определяет общие правила разметки текстов языка SCn. SCnML содержит следующие классы тегов для разметки sc.n-статей:

- тег описываемого объекта;
- тег связи;
 - тег однокомпонентной связи;
 - тег однокомпонентной связи с тегом ролевого отношения;
 - тег однокомпонентной связи без тега ролевого отношения;
 - тег многокомпонентной связи;
 - тег многокомпонентной связи с вложением;
 - тег многокомпонентной связи без вложения;
 - тег ролевого отношения;
- тег компонента связи;
 - тег компонента связи без вложенного ролевого отношения;
 - тег компонента связи с вложенным ролевым отношением;
- тег scnml-запроса.

Любая прикладная реализации SCnML должна содержать указанные классы тегов. На множестве scnml-тегов заданы следующие типы отношений: “быть корневым тегом текста, соответствующего sc.n-статье”, “родительский тег - дочерний тег”. Приведем свойства SCnML-текстов:

- каждый SCnML-тег связан с другим SCnML-тегом понятием уровня, уровень позволяет задать отношение между родительским и дочерним тегом и определяет, в рамках какого контекста идет описание стоящее ниже по уровню (для какого понятия определены связи и к какой связи относятся компоненты связей);
- на одной web-странице могут располагаться scnml-тексты, соответствующим нескольким sc.n-статьям;
- scnml-текст, соответствующий одной sc.n-статье может входить в состав scnml-текста, соответствующего другой sc.n-статье;
- теги в рамках scnml-текста записываются по следующим правилам:
 - первым следует корневой тег, тег описываемой сущности;
 - за корневым тегом может следовать только тег связи;
 - тегу компонента связи всегда предшествует тег связи.

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. Таким образом, обеспечивается решение двух указанных выше задач семантической структуризации на уровне внешнего представления и уровне языков разметки.

1.3. Языковые средства представления мультимедиа данных в sc-моделях интеллектуальных систем

О том, что разработка средств семантической аннотации мультимедиа данных – это востребованная научная и практическая задача, свидетельствуют работы, которые ведутся в рамках проектов Semantic Web. Группой W3C проведен анализ существующих технологий для применения мультимедиа и выделены следующие классы мультимедиа данных: неподвижные изображения, видео, аудио, текст, данные общего назначения. Также выделены основные классы для категоризации использования мультимедиа по следующим направлениям: по рабочим потокам (для публикации и издания и т. д.), по областям применения (развлечения, новости, спорт и т.д.), для промышленного использования (для вещания, для музыки, для издательства и т.д.).

В результате исследования рассмотрены мультимедиа данные, для которых в настоящие время описаны спецификации. Рассмотренные спецификации включают мультимедиа данные, представляемые как с помощью не связанных с XML-платформой средств, так и с помощью средств XML-платформы, и спецификации мультимедиа данных с использованием онтологий. Основной проблемой работ по спецификации мультимедиа данных является наличие узкоспециализированных направлений по разработке мультимедиа спецификаций, которые не связаны друг с другом, о чем можно судить на основании [W3C, 2007].

Работы, связанные с унифицированной спецификацией мультимедиа данных, только начинают развиваться. В рамках данной статьи предлагается один из подходов к универсальной спецификации мультимедиа данных. Основной целью предлагаемого подхода является не разработка неких универсальных средств спецификации мультимедиа данных, а предложение средств, которые позволят, унифицировано, в едином стиле описывать мультимедиа данные, используя уже существующие специализированные системы метаданных.

В качестве языковых средств унифицированной спецификации произвольных мультимедиа данных предлагается использовать язык гипермедийных структур (ЯГС) [Колб, 2009]. ЯГС позволяет специфицировать любой мультимедиа ресурс с учетом семантики предметной области и связей мультимедиа ресурса с понятиями предметной области. ЯГС является sc-языком (то есть языком, построенным на базе SC-кода), ориентированным на представление структурных моделей предметных

областей, в которых объектами исследования являются отображаемые пользователям файлы интеллектуальной системы, а предметом исследования – семейство понятий, обеспечивающих спецификацию этих файлов и описание различных синтаксических и семантических связей между ними.

Уточним некоторые базовые понятия, которые будут использованы в дальнейшем: понятие sc-файла, понятие формата, понятие способ представления формата, способ визуализации формата и области использования формата.

Основным понятием, с которым работает ЯГС, является понятие файла sc-системы – sc-элемента, который является инородным по отношению к SC-коду объектом произвольной структуры. Будем называть файл sc-системы (или файл, который интегрирован в sc-систему) *sc-файлом*. В рамках данной работы мы рассматриваем, только те типы sc-файлов, которые являются мультимедийной информационной конструкцией. Такие информационные конструкции в SC-коде обозначаются с помощью sc-ссылок или sc-узлов (в случае, если речь идет о файлах, внешних прикладных программах и внешних программных системах, которые в текущий момент до конца не интегрированы в состав sc-системы – т.е. формально отсутствует сама ссылка) [OSTISA, 2011].

Под *форматом* будем понимать спецификацию (однозначное описание) структуры данных, записанных в sc-файле. Формат определяет *способ хранения* информационной конструкции (например: растровый, векторный) и *форму хранения* информационной конструкции (например: используемый алгоритм сжатия).

Под *способом представления* (кодирования) формата будем понимать некоторые языковые средства, позволяющие записать спецификацию структур данных, описываемых некоторым форматом.

Под *способом визуализации* формата будем понимать спецификацию информационной конструкции, позволяющую наглядно (с наибольшим когнитивным эффектом) отобразить данную информационную конструкцию.

Под *областью использования* формата будем понимать некоторую область информационных технологий в рамках, которой данный формат нашел применение. Как правило, область использования определяет набор требований к формату (качество изображения, наличие или отсутствие алгоритма сжатия, поддержка нескольких аудио-поток и т.д.).

С точки зрения семантики понятие формата – это небинарное отношение, которое устанавливает связь sc-файла с его способом хранения, формой хранения, способом представления и множеством областей использования формата. Это отношение,

позволяющее представить знания о том, как обрабатывать sc-файл, находящийся в sc-системе. Каждая связка такого отношения связана отношением *способ визуализации** с некоторой информационной конструкцией, которая определяет способ отображения данного формата с помощью средств визуализации и просмотра, которые присутствуют в sc-системе. То есть понятие формата обеспечивает даталогическую (или синтаксическую) спецификацию мультимедиа ресурса. А использование данных отношений для всех форматов позволяет обеспечить унифицированную семантическую спецификацию для всех возможных представлений мультимедиа данных.

Заметим, что sc-файл вовсе не обязан явно “хранить” некоторую информационную конструкцию в качестве своего содержимого. Эта информационная конструкция может быть неизвестна (не сформирована) и разбита на фрагменты, каждый из которых представлен явно, и, следовательно, нет никакой необходимости явно представлять и хранить всю исходную информационную конструкцию. Данное свойство позволяет говорить о возможности унификации различных подходов для анализа мультимедиа данных [Колб, 2011].

Для инфологической (или семантической) спецификации мультимедиа ресурса, используются понятия и отношения, которые находятся в предметной базе знаний sc-системы. Основу ключевых элементов для этого набора отношений задают ключевые узлы языка гипермедийных структур. Основу языка задают классы различных мультимедийных объектов, приведем основные из них:

- Неподвижное изображение
- Абстрактный текст
 - Линейная конфигурация графем
 - Терм
 - Символьное представление числа
 - Текст
 - Нелинейная конфигурация графем
 - sc.g-конструкция
 - sc.s-конструкция
- Аудиоинформация
- Динамическая графика
 - Видеоинформация
 - Анимация
- Контейнер (составной объект, например архив, который может содержать различные мультимедиа-объекты)
- Элемент программного обеспечения

Каждый объект из приведенного набора классов обладает набором уникальных для класса объекта свойств. Например, с объектами класса *“Неподвижное изображение”* может быть связана

информация об авторе, дате создания объекта, а также информация о том с помощью какого формата представлен в sc-файле объект. С объектами класса *“Текст”* может быть связана информация о кодировке текста, формате представления текстовой информации, авторе, дате создания, дате модификации и т.д.

Приведем некоторые базовые отношения, которые задаются на множестве мультимедийных объектов: *презентация**, *иллюстрация**, *автор**, *комментарий**, *пояснение**, *биография**, *карта**, *рецензент**, *дата создания**, *дата модификации** и др.

2. Интерпретация семантической модели web-сайта на различные платформы

2.1. Состояние работ в области языковых средств обработки и эффективного представления в хранилищах данных моделей интеллектуальных систем

Уровень серверных скриптов, предоставляет разработчикам web-ресурсов широкий выбор инструментов, позволяющих проводить различные операции над разметкой и реализовать требующийся от web-ресурса набор функций. Для современных web-сайтов это инструмент, позволяющий динамически изменять разметку и вносить изменения на внешнем представлении, – другими словами – это основной способ реструктуризации на уровне разметки и внешнего представления. Кроме этого этот уровень на сегодняшний день является звеном, обеспечивающим связь уровня разметки и внешнего представления с уровнем хранилищ данных.

Научный и практический потенциал, достигнутый на этом уровне, позволяет говорить о широких возможностях для реализации алгоритмов искусственного интеллекта. Кроме традиционных технологий, используемых в программировании, в ряде языков, используемых в web-разработках, реализована поддержка следующих парадигм программирования:

- функциональной (PHP, Python, Ruby, Perl, Erlang);
- логической (Mercury [Mercury 2011] через API для Java, C#, Erlang);
- декларативной (PHP, Python, Ruby, Perl);
- метапрограммирования на основе интроспекции и интерпретации произвольного кода (Python, Ruby, PHP и ряд других языков).

Уровень хранилищ данных. Всплеск работ по хранилищам данных и знаний, которые ориентированы на отличное от реляционной модели представление произошел в 2009 году с появлением целого ряда практических результатов в рамках направления NoSQL (Not only SQL) [NoSQL, 2011]. Появление работ было связано с тем, что современные реляционные СУБД не могут эффективно справляться с нагрузками, возникающими при обработке огромных объемов данных в виду наличия ряда проблем в трёх областях:

- горизонтальное масштабирование при больших объемах данных; например, как в случае социальной сети Digg (3 терабайта для зеленых значков, отображаемых, если ваш друг сделал “dugg” на статье) или в случае социальной сети Facebook (50 терабайт для поиска по входящим сообщениям), или в случае системы Internet-аукционов eBay (2 петабайта в целом);
- производительность каждого отдельного сервера;
- не гибкий дизайн логической структуры [Новожилов, 2011].

Как средство, эффективно решающее проблемы реляционных СУБД, появились различные сетевые СУБД, объектно-ориентированные СУБД, а также СУБД “ключ значение”.

Отметим основные недостатки двух нижних уровней:

- высокая степень зависимости существующих технологий разработки web-приложений от платформ, на которых они реализованы, что приводит к существенным изменениям этих технологий при переходе на новые платформы;
- стремление современных подходов к разработке web-ресурсов обеспечить высокий уровень гибкости приложений, без уделения достаточного внимания принципам унификации и интеграции web-ресурсов, построенных на базе разных подходов.

2.2. Иерархическая модель семантического web-сайта

Формальной основой предлагаемого подхода выбран SC-код, поэтому отправной точкой при построении интеллектуальной системы на web-платформе будет являться некоторая прикладная sc-модель (модель предметной области, закодированная с помощью SC-кода).

В качестве одного из возможных подходов к реализации интеллектуальных систем для web-пространства на основе sc-модели предлагается использовать подход на основе иерархической модели семантического web-сайта (где семантическим web-сайтом будем называть сайт, построенный с использованием средств SC-кода). Основным принципом этой модели является интерпретация прикладной sc-модели на различные традиционные прикладные инструментарии (Рис. 1) для разработки web-сайтов согласно ранее выделенным выше уровням.

Основной задачей при таком подходе является четкая трансляция различных представлений прикладной sc-модели на различных уровнях (уровне внешнего представления, уровне языков разметки, уровне серверных скриптов, уровне хранилищ данных.) для обеспечения тесной интеграции между выделенными уровнями. В sc-системах такой класс задач решает компонент транслятор. Основными языковыми средствами, с которыми будет работать разработчик семантического web-сайта, будут SCnML-тексты.

- операции трансляции SCnML-текстов в некоторое представление в хранилище данных и обратная трансляция этого представления в тексты SCnML;
- операции трансляции SCnML-текстов в эквивалентные объекты SCnML-тегам объекты языков прикладного программирования и обратная трансляция.

2.3. Интерпретация прикладной sc-модели на SCn-модель web-сайта

Основной задачей при интерпретации прикладной sc-модели на sc.n-модель web-сайта является выявление наиболее важных сущностей с точки зрения подачи информации пользователю web-сайта. Такие сущности, как правило, соответствуют некоторым web-страницам в традиционных web-сайтах, согласно базовым принципам представления гипертекстов. В рамках данной работы для описания сущностей предметной области мы используем sc.n-статьи. Кроме определения сущностей, которые будут описываться с помощью sc.n-текстов, на плечи разработчика ложится задача разработки

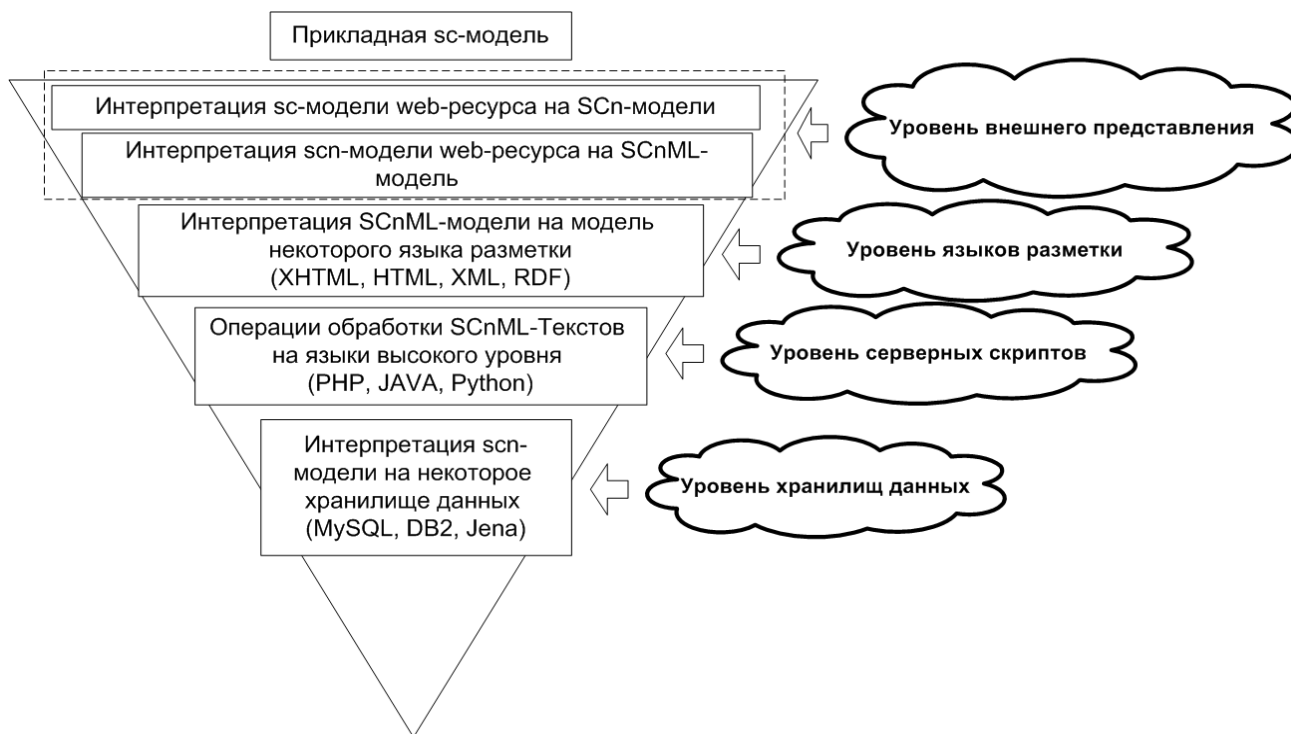


Рисунок 1 – Иерархическая модель семантического web-сайта

Таким образом, можно выделить следующие классы операций трансляции, которые необходимо реализовать при предлагаемом подходе:

- операции трансляции SCnML-текстов в тексты внешнего sc.n-представления;
- операции трансляции SCnML-текстов в тексты языков разметки, используемых в web-ресурсах (HTML, XHTML, XML, RDF и др.);
- операции трансляции SCnML-текстов в эквивалентные объекты SCnML-тегам объекты скриптовых языков уровня разметки (JavaScript) и обратная трансляция;

логической структуры web-страниц, которые будут содержать некоторое множество таких сущностей. В трактовке, предлагаемой в рамках данной работы, web-страница – это методологическая единица, которая определяет порядок и приоритеты пользователя при ознакомлении с материалами, расположенными на странице сайта. Такие web-страницы мы будем называть sc.n-страницами. Таким образом, результатом интерпретации прикладной sc-модели на sc.n-модель web-сайта будет являться множество sc.n-страниц,

соответствующее некоторой методике изложения материала web-сайта.

2.4. Интерпретация SCn-модели web-ресурса на традиционные языки разметки и языки Semantic Web

Для полной интеграции с традиционными инструментариями разработки web-сайтов необходим прозрачный и понятный механизм интеграции, который позволяет мощност традиционных средств разработки сочетать с формальной точностью семантического представления информации. Для этих целей в рамках данной работы предлагается использовать модель языковых средств разметки SCn-кода, SCnML.

Alice

← foaf:Person
⇒ foaf:knows

- Eve
 - ⇒ foaf:name ["Eve"]
 - ⇒ foaf:homepage ["http://example.com/Eve"]
 - ← foaf:Person
- Manu
 - ⇒ foaf:name ["Manu"]
 - ⇒ foaf:homepage ["http://example.com/Manu"]
 - ← foaf:Person

- Bob
 - ⇒ foaf:name ["Bob"]
 - ⇒ foaf:homepage ["http://example.com/Bob"]
 - ← foaf:Person
 - ⇒ dc:creator *Beatufil Sunset*
 - ⇒ dc:title ["Beatufil Sunset"]

Рисунок 2 “Социальная сеть Алисы” в виде sc.n-статьи.

Для интерпретации sc.n-текстов на одну из реализаций scnml-разметки разработчику семантического сайта необходимо выполнить следующие действия: разработать способ представления различных типов связей отношений и разработать способ представления различных типов компонент данных связей.

Рассмотрим такую интерпретацию SCn-текстов (Рис. 2) на примере описания “Социальной сети Алисы”, который является одним из известных примеров применения RDF-модели при разметке

web-документов с помощью языков HTML или XHTML[RDFa primer, 2011]. Пример выбран не случайно и позволяет на наш взгляд показать, как можно использовать технологии на базе SCn-кода и sc-моделей при разработке web-сайтов.

На основании представленного, на рисунке 2 sc.n-текста мы выделяем основной объект описания *Alice* и выделяем следующие типы связей отношений:

- многокомпонетные типы связей: foaf:knows, foaf:homepage, foaf:creator, dc:title;
- однокомпонетные типы связей: foaf:Person.

Также выделим следующие компоненты связей: Eve, [“Eve”], [“http://example.com/Eve”], [“Manu”], [“http://example.com/Manu”], [“Bob”], [“http://example.com/Bob”], *Beatufil Sunset*, [“Beatufil Sunset”]. Все выделенные компоненты связей будут являться компонентами связей без вложенного ролевого отношения.

Введем идентификатор сущности *Alice* “http://example.com/alice#me” этот идентификатор используется в атрибуте RDFa *about* для указания того, что вложенные в HTML тег элементы описывают сущность, указанную в значении свойства *about*. В начале sc.n-статьи у нас указано, что объект *Alice* является элементом множества *foaf:Person* (этот факт sc.n-статье показывает специальный маркер “←”). На основании выше сказанного фрагмент статьи, описывающий первые две строки sc.n-статьи будут выглядеть следующим образом:

```
<div class = "scnarticle" about="http://example.com/alice#me">
  <div class = "concept" about="http://example.com/alice#me">
    <div typeof="foaf:Person" >
      <div property="foaf:name">Alice</div>
    </div>
```

...
</div>

Перейдем к описанию части sc.n-статьи, содержащей бинарное отношение *foaf:knows* (о том, что отношение бинарное в sc.n-статье показывает специальный маркер “⇒”). Ввиду однотипности связанных отношением *foaf:knows* объектов покажем, как представляется нижняя часть sc.n-статьи (рис. 2 выделенный штриховым контуром).

В RDFa информация о связи некоторого web-ресурса или фрагмента web-документа с другим ресурсом или фрагментом web-документа, который описывается в статье, указывается в атрибуте *rel*. Атрибут *property* позволяет указать, каким отношением текстовый фрагмент в web-документе связан с описываемым объектом. Таким образом, получается, что бинарные ориентированные отношения *foaf:name*, *foaf:homepage*, *dc:title* мы будем оформлять с помощью атрибута *property*, так как данные отношения используются для связи объекта *Bob* с некоторыми фактами о нём, представленными в текстовом виде прямо в документе. Отношение *dc:creator* мы будем

оформлять с помощью атрибута *rel*, так как оно связано фрагментом web-документа. С учетом приведенных выше уточнений, sc.n-статья будет выглядеть следующим образом:

```
...
<div class="connective" about="http://example.com/alice#me"
rel="foaf:knows">
...
<div class="component">
  <div typeof="foaf:Person">
    <div property="foaf:name"> Bob </div>
    <a rel="foaf:homepage" href="http://example.com/bob">
Bob</a>
    <div class="in" about="http://example.com/bob">
      <div class="connective" rel="dc:creator"
about="http://example.com/bob">
        <div class="component">
          <div property="dc:title">Beautiful Sunset</div>
        </div>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>
```

Атрибуты *class* со значениями “component” и “connective” в данном фрагменте используются для указания структурных элементов sc.n-статьи. Для уменьшения объёма, где не требуется явного выделения scnml-тегов для реализации верстки sc.n-отображения данной sc.n-статьи (так как, scnml-теги просматриваются, если соотнести sc.n-текст и предложенную в данном примере его html-верстку), структурные элементы sc.n-статьи опущены.

Уточним еще ряд элементов, которые важны при четкой интерпретации. В отличие от RDF модели, в

Кроме этого, необходимо заметить, что, благодаря тому, что sc-модели изначально не привязаны к web-платформе, необходимость в элементах типа *href*, которые позволяют указать ссылку на некоторых внешний ресурс, отпадает.

Отметим также, что для упрощения примера из текста HTML убраны объявления пространств имен для RDF-словарей “Дублинского ядра” (dc) и FOAF (foaf).

Разобранный пример позволяет продемонстрировать один из возможных подходов к представлению sc.n-текстов с помощью средств HTML или XHTML+RDFa. Отметим, что данный пример приведен лишь с целью продемонстрировать прозрачность перехода от sc.n (или sc-представления) к некоторому более традиционному виду web-ресурса. Для практического использования целесообразнее применять другие, более гибкие подходы, связанные с использованием цепочек преобразований в основе, которых лежит единое формальное представление в виде SC-кода.

2.5. Интерпретация sc.n-модели на хранилища данных

Семантическая структуризация на уровнях внешнего представления и уровне языков разметки обеспечивают, как правило, эффективную навигацию и поиск как в ручном режиме, так и с привлечением сторонних поисковых средств. Однако в современных web-сайтах не вся

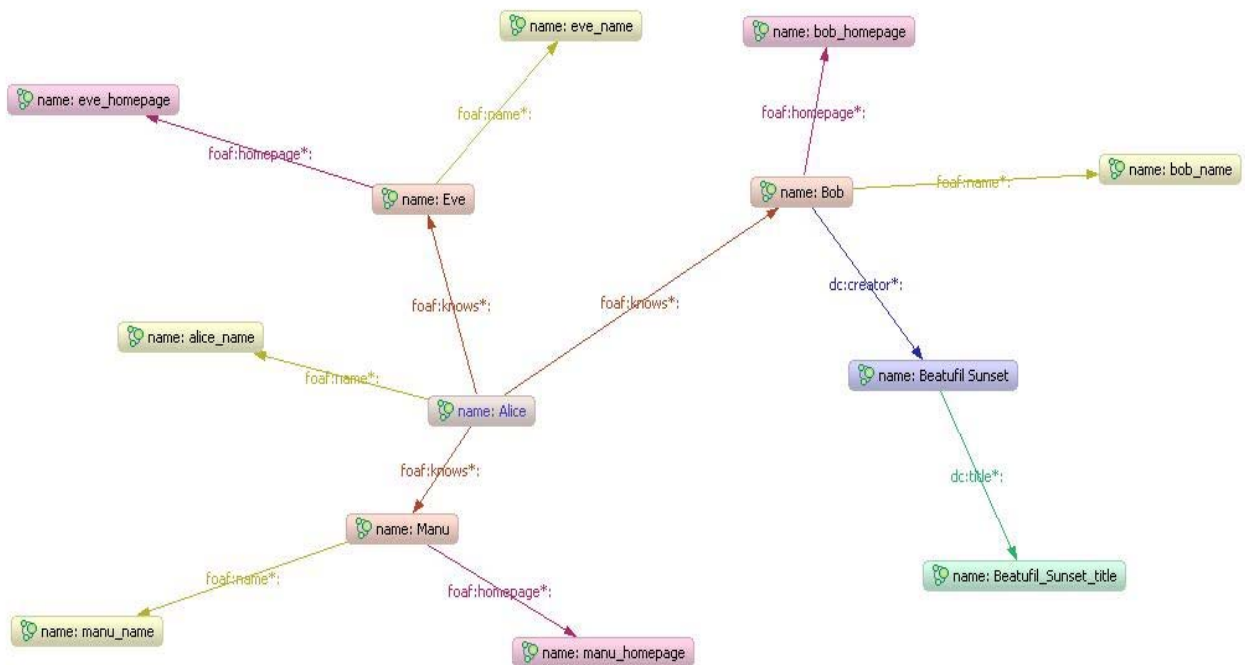


Рисунок 3 Пример sc.n-статьи "Социальная сеть Алисы" представленная в СУБД Neo4j (визуальное представление Neoclipse).

sc-моделях знания об объектах предметной области не привязаны к понятию тройки (субъект предикат объект), поэтому SCn-код не требует в своем представлении элементов подобных *about*, *rel*.

информация представлена с помощью текстов разметки, большая часть информации генерируется динамически, используя ту информацию, которая находится в некотором хранилище данных. При

использовании предложенного подхода эффективность такого архитектурного решения будет зависеть в первую очередь от эффективного представления sc-модели или sc.n-статей в рамках некоего хранилища данных. В качестве примера интерпретации возьмем интерпретации sc.n-модели web-ресурса на графовую СУБД Neo4j [Neo4j, 2011].

СУБД Neo4j поддерживает модель модифицированной семантической сети.

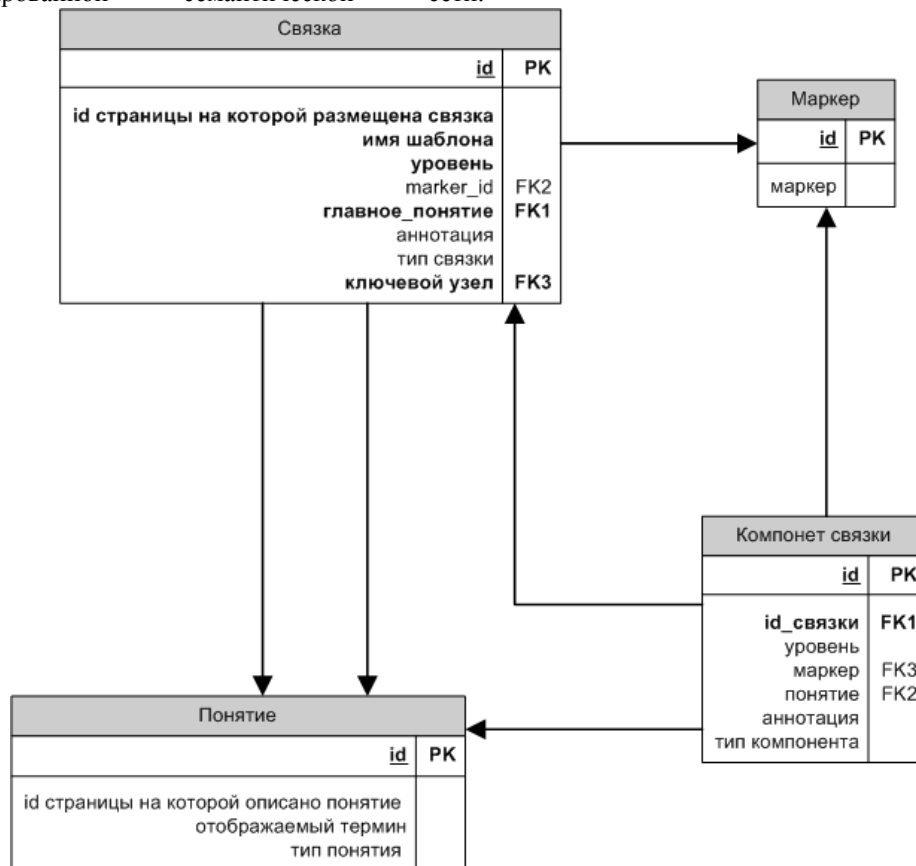


Рисунок 4 Схема экспериментальной базы данных для представления sc.n-статей на сайте conf.ostis.net

Модификация заключается во введении специальных типов узлов и дуг, которые могут быть проиндексированы. За счет механизма индексов в этой СУБД реализованы эффективные средства для реализации различных алгоритмов поиска на графах.

Существенным отличием SC-кода от других средств представления семантических сетей является возможность проведения sc-дуг из sc-узлов sc-дуг в другие sc-дуги. Поэтому проблему представления таких видов конструкций придется решить для СУБД Neo4j. Введем следующие допущения для СУБД Neo4j:

- связка будет представляться в Neo4j дугой в том случае, если она является связкой ориентированной бинарной пары, и в том случае, если не требуется уточнение атрибутивным отношением роли каждого компонента связки в рамках данной связки;

- связка будет представляться в Neo4j узлом в том случае, если она является связкой неориентированного бинарного отношения или связкой небинарного отношения;
- будем добавлять в конце идентификатора связки терм “:” в том случае, если она является связкой атрибутивного отношения;
- будем добавлять в конце идентификатора связки терм “:*” в том случае, если она не является связкой атрибутивного отношения.

Введенных допущений достаточно для представления широкого класса sc.n-текстов. Однако для того, чтобы поддержать все возможности SC-кода необходимо будет ввести специальные обозначения для возможных в SC-коде структурных типов sc-элементов.

Введем еще одно допущение, которое позволит нам моделировать sc.n-страницы и sc.n-статьи в рамках Neo4j. Такое допущение вводится в целях реализации эффективной навигации и поиска в рамках семантического web-сайта при предлагаемом нами подходе. Допущение будет заключаться в группировке узлов и дуг, которые встречаются в конкретной sc.n-статье по sc.n-статьям в рамках представления в Neo4j, и группировки sc.n-статей в рамках sc.n-страниц. Такие группировки мы будем осуществлять, используя механизм индексации узлов и связей Neo4j. Благодаря такому механизму у нас с одной стороны сохранится полноценная семантическая

сеть, а с другой стороны будет обеспечена эффективная навигация между структурными элементами семантического web-сайта – sc.n-статьями, sc.n-страницами и sc-элементами, которые в них расположены. Результат интерпретации sc.n-статьи на СУБД Neo4j представлен на рис.3.

Для поддержки sc-файлов, которые используются в SC-коде для представления мультимедиа данных, в Neo4j предусмотрены “property” у узлов и связей. Эти элементы позволяют хранить различные типы данных и допускают бинарное и текстовое представление этих данных. Ввиду того, что Neo4j позволяет эффективно поддерживать все элементы SC-кода, можно предположить, что эта СУБД будет являться эффективной платформой для программной реализации sc-хранилищ (хранилищ знаний, закодированных с помощью SC-кода).

Рассмотрим решение, которое позволяет осуществить интерпретацию sc.n-модели web-сайта на реляционные СУБД. Наиболее перспективным, на наш взгляд, подходом отображения любых семантических моделей на реляционные СУБД является способ, в основе которого лежит представление не самих элементов семантической сети, а структурных элементов более высокого уровня абстракции – понятий и отношений. Такое представление хотя и обедняет элементную базу семантической сети, однако позволяет эффективно решать практические задачи.

Для решения задачи представления sc.n-модели web-сайта, на наш взгляд, наиболее простым решением является использование в качестве основных отношений базы данных отношения соответствующие типологии связей и компонентов связей. Для более четкой структуризации необходимо ввести отношения, связывающие набор связей и их компонентов с sc.n-статьей и sc.n-страницей.

В качестве примера интерпретации sc.n-модели web-сайта на реляционную СУБД приведем (рис. 4) схему фрагмента базы данных, которая использовалась для поддержки тегов scnml-запросов в рамках сайта конференций OSTIS. Как видно из рисунка, схема БД существенно облегчена, за счет использования в рамках сайта конференции одной sc.n-страницы одной sc.n-статьи. Из примера понятно, что схема реляционной базы данных при интерпретации sc-моделей может быть разработана с учетом типологии используемых в рамках конкретной реализации семантического web-сайта отношений, их связей и компонентов этих связей.

2.6. Операции обработки scnml-текстов на языках высокого уровня

Уровень серверных скриптов – это часть сайта, которая позволяет обеспечить интерактивное общение с пользователем. При реализации подходов к обработке sc.n и scnml-текстов можно выделить следующие направления работ:

- разработка библиотек для эффективной обработки отдельных фрагментов sc.n-страниц на стороне web-клиента с использованием технологии AJAX;
- разработка серверных библиотек для генерации текстов SCnML-разметки на лету или из хранилища данных;
- разработка серверных библиотек обработки текстов SCnML-разметки, представленных в виде текста;
- разработка библиотек общения с хранилищами данных посредством технологий подобных ORM (Object Relation Mapping) для реляционных СУБД или OTM (Object to Triples Mapping) для RDF-хранилищ;
- разработка прикладных интерфейсов общения между различными семантическими web-сайтами на основе SC-кода, и, как следствие, реализация интерфейсов для общения с поисковыми системами. Такой интерфейс должен обеспечить не только эффективное индексирование текстового представления sc.n-страниц, но и позволить использовать средства семантического поиска на основе специального языка вопросов.

Направления практических работ, которые указаны выше, не являются новыми с точки зрения подходов к разработке традиционных сайтов, однако, реализация указанных инструментов предусматривает реализацию ряда алгоритмов, которые непосредственно связаны с задачами обработки семантических сетей.

ЗАКЛЮЧЕНИЕ

Отметим ряд достоинств предлагаемого подхода к разработке интеллектуальных систем на базе web-платформы:

- подход является полностью независимым от программных средств реализации за счет использования единой формальной основы – sc-модели предметной области, использование таких моделей дает возможность реализации в рамках web-сайтов средств семантического поиска;
- при использовании предложенного подхода web-сайты будут совместимы на уровне SCn и SCnML представлений, что позволяет, используя средства интеграции баз знаний, построенных с использованием SC-кода, объединять семантические web-сайты в семантические web-порталы знаний, которые будут построены по единым унифицированным принципам;
- построенные при использовании предложенного подхода web-сайты можно рассматривать как исходные тексты баз знаний;
- предложенный подход позволяет эффективно использовать технологии Semantic web в качестве платформы интерпретации sc-моделей.

Предложенный подход разработан в рамках проекта OSTIS и опробован на следующих сайтах:

- сайт конференций OSTIS (<http://conf.ostis.net>);

- информационно справочная система по геометрии (<http://ostisgeometry.sourceforge.net/>).

Работа выполнена при поддержке гранта БРФФИ Ф10М-085 и гранта БРФФИ-РФФИ Ф10Р-175.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Веб-фрагмент, 2011] Спецификация формата веб-фрагмента – версия 0.9 [Электронный ресурс]. – 2011. - Режим доступа: <http://msdn.microsoft.com/ru-ru/library/cc304073%28VS.85%29.aspx> . – Дата доступа: 30.11.2011

[Вики о микроформатах, 2011] Вики о микроформатах [Электронный ресурс]. – 2011. - Режим доступа: http://microformats.org/wiki/Main_Page-ru. – Дата доступа: 30.11.2011

[Гаврилова, 2000] Гаврилова, Т.А. Базы знаний интеллектуальных систем/ Т.А. Гаврилова, В.Ф. Хорошевский //СПб – Питер, 2000 г., 384 стр.

[Голенков, 2001] Представление и обработка знаний в графодинамических ассоциативных машинах /В. В. Голенков, [и др]; – Мн. : БГУИР, 2001.

[Грибова, 2011] Грибова, В.В. Облачная платформа для разработки и управления интеллектуальными системами //В. В. Грибова, [и др]; // OSTIS-2011/ Открытые семантические технологии проектирования интеллектуальных систем// Материалы международной научно-технической конференции. - 2011. - С. 5-15.

[Колб, 2009] Колб, Д.Г. Средства просмотра баз знаний интеллектуальных систем / Д. Г. Колб // Вестник БрГТУ. - 2009. - № 5. - С.58-62.

[Колб, 2011] Колб, Д.Г. Направления, методы и средства применения семантических сетей в Internet-технологиях/ Д. Г. Колб // OSTIS-2011/ Открытые семантические технологии проектирования интеллектуальных систем// Материалы международной научно-технической конференции. - 2011. - С.443-463.

[Новожилов 2011] Новожилов, А. Обзор NoSql систем. 2011. - Режим доступа: <http://habrahabr.ru/blogs/nosql/77909/> . – Дата доступа: 30.11.2011

[Раскин, 2004] Джеф Раскин Интерфейс: новые направления в развитии компьютерных систем.- Пер. с англ.-СПб: Символ-Плюс, 2004-272 с.

[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.

[DCMI, 2011] The Dublin Core Metadata Initiative [Электронный ресурс]. – 2011. - Режим доступа: <http://dublincore.org/>. – Дата доступа: 30.11.2011

[FOAF, 2011] The Friend of a Friend (FOAF) project [Электронный ресурс]. – 2011. - Режим доступа: <http://www.foaf-project.org>. – Дата доступа: 30.11.2011

[Mercury, 2011] The Mercury Project [Электронный ресурс]. – 2011. - Режим доступа: <http://www.mercury.cs.mu.oz.au/index.html>. – Дата доступа: 30.11.2011

[Microsoft, 2001] Microsoft Inductive User Interface Guidelines 2011 Microsoft Corporation February 9, 2001 [Электронный ресурс]. – 2011. - Режим доступа: <http://msdn.microsoft.com/en-us/library/ms997506>.

[Neo4j, 2011] Neo4j the graph database. [Электронный ресурс]. – 2011. - Режим доступа: <http://neo4j.org>. – Дата доступа: 30.11.2011

[NoSQL, 2011] NoSQL. [Электронный ресурс]. – 2011. - Режим доступа: <http://nosql-database.org>. – Дата доступа: 30.11.2011

[OSTISa, 2011] Проект 10. Унифицированный способ абстрактного кодирования семантических сетей [Электронный ресурс]. – 2011. - Режим доступа: http://www.ostis.net/wiki/Проект_10. – Дата доступа: 30.11.2011

[RDFa primer, 2011] RDFa Primer Bridging the Human and Data Webs W3C Working Group Note 14 October 2008 [Электронный ресурс]. – 2011. - Режим доступа: <http://www.w3.org/TR/2008/NOTE-xhtml-rdfa-primer-20081014/>

[Topaz, 2011] Topaz. [Электронный ресурс]. – 2011. - Режим доступа: <http://www.topazproject.org>. – Дата доступа: 30.11.2011

[W3C, 2007] Multimedia Vocabularies on the Semantic Web W3C Incubator Group Report 24 July 2007 [Электронный ресурс]. – 2011. - Режим доступа: <http://www.w3.org/2005/Incubator/mmssem/XGR-vocabularies/#MusicXML>

[W3C, 2011] World Wide Web Consortium [Электронный ресурс]. – 2011. - Режим доступа: <http://www.w3.org>. – Дата доступа: 30.11.2011

THE WEB-BASED IMPLEMENTATION SEMANTIC MODELS OF INTELLIGENT SYSTEMS

Kolb D.G.

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

kolb@bsuir.by

Principles and approaches to development of semantic web-sites with using facilities traditional technologies are described. The proposed approaches found on SC-model's concept and the way of the pseudonaturally representation these models.

INTRODUCTION

The most sharpest problems for developers intelligent systems is problems of the development of unified ways and tools advancement systems of artificial intelligent in Internet. The main way for organization traditional informational systems (IS) in the Internet is organization it's in like of web-portal or web-site. For that organization IS can be extract following levels on which can be exercise conversion traditional web-resource into intellectual web-resource. There are levels of external representation, markup languages, server's script and stores

MAIN PART

As the formal basis of the proposed approach will use the semantic network with a base set-theoretic interpretation. The main method of information coding for these networks is the SC (Semantic Code)-code. Intelligent systems built using SC-code is called sc-systems. Entities web-pages will be represent the elements of a knowledge base (KB). This approach allows us to consider a web-site as a specialized intelligent system that solves the problem of dialogue and substantive rights of intellectual systems and providing solutions to key subject of intellectual systems.

Looking at in this perspective, a web-site, we conclude that the degree of intelligence web-site will depend on the extent to which each of the extracted web-site levels of semantic technologies.

CONCLUSION

The main advantage of the introduced approach is the ability to use it in developing intelligent systems for various web-platforms, including Semantic Web

platform.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.38

СОВРЕМЕННОЕ СОСТОЯНИЕ И ТЕНДЕНЦИИ РАЗВИТИЯ ТЕХНОЛОГИЙ РЕКОНФИГУРИРУЕМЫХ ЦИФРОВЫХ УСТРОЙСТВ

Иванюк А.А.

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

ivaniuk@bsuir.by

В докладе приводится исторический обзор эволюции реконфигурируемых цифровых устройств. Приводятся примеры архитектур современных программируемых логических интегральных схем и области их применения. Рассматриваются вопросы реализации вычислительных устройств со статической и динамической конфигурацией. Делается обзор возможных направлений развития технологий реконфигурируемых цифровых устройств.
Ключевые слова: реконфигурируемые вычисления, программируемые логические интегральные схемы.

ВВЕДЕНИЕ

Целью данного доклада является определение возможностей современных технологий для проектирования и создания цифровых вычислительных устройств с нетрадиционными архитектурами, в том числе и динамически реконфигурируемых вычислительных устройств. Широкое распространение программируемых логических интегральных схем (ПЛИС) совместно с высокоуровневыми HDL-языками дает практически каждому проектировщику неограниченные возможности по схемотехническому проектированию и прототипированию заказных цифровых устройств (ASIC, от англ. – Application Specific Integrated Circuit). С другой стороны, относительная дешевизна и неуклонный рост степени сложности ПЛИС делают их прямыми конкурентами ASIC, позволяя, например, реализовывать не только модели коммерческих микропроцессоров (soft-processors), но и вычислительных устройств с уникальными нетрадиционными архитектурами, представляющие собой объекты интеллектуальной собственности (IP-cores).

1. От ПЗУ до программируемых систем на кристалле

За всю историю развития цифровой вычислительной техники проектировщики и разработчики пытались находить различные подходы, позволяющие быстро и эффективно воплощать вычислительные алгоритмы в цифровой аппаратуре. Цифровую схему произвольного вычислительного устройства можно представить в виде совокупности комбина-

ционных и последовательностных схем (схем с памятью), первые из которых являются результатом реализации переключательных функций [Magio, 1986]. Произвольную комбинационную схему можно построить на основе базовых логических элементов (например И, ИЛИ, НЕ). Соответствующие методики декомпозиции представления комбинационных схем позволяли их реализовывать на интегральных схемах малой и средней степени интеграции. Незначительное изменение функционала реализуемых переключательных функций либо структуры комбинационной схемы неизбежно приводило к полному перепроектированию и повторной реализации соответствующего устройства. Однако произвольная переключательная функция может быть представлена в совершенной дизъюнктивной нормальной форме (СДНФ) или иными словами – таблицей истинности. Аппаратная реализация таблицы истинности возможна посредством использования запоминающих устройств с произвольным доступом (ПЗУ и/или ОЗУ). Например, для реализации произвольной переключательной функции трех аргументов необходимо и достаточно бит-ориентированное ПЗУ с организацией 8×1 . При этом ПЗУ программируется таким образом, чтобы значения адресов ячеек, в которые будет записано значение '1', совпадали со значениями единичных термов переключательной функции. Во все остальные ячейки необходимо записать значение '0'. На адресные входы ПЗУ подаются значения соответствующих аргументов переключательной функции, а по окончании операции чтения на выходной линии данных ПЗУ фиксируется значение реализованной функции. Перепрограммирование ПЗУ иным информационным контентом ведет к реализации другой переключатель-

ной функции. Так, ПЗУ с организацией 8x1 способно реализовать одну из 256 возможных функций от трех аргументов, либо иными словами – 256 различных комбинационных схем. Имея в составе вычислительного устройства подобное ПЗУ возможно изменении его комбинационной части не прибегая к его перепроектированию и повторной реализации.

Подобная идея нашла широкое применение у проектировщиков и разработчиков цифровой аппаратуры и стала неуклонно совершенствоваться. Например, для увеличения числа аргументов переключательной функции стали использовать двумерное декодирование, заключающееся в расширении функционирования ПЗУ при помощи мультиплексов. На информационные входы мультиплексора подаются фиксированные значения '0' и '1' в соответствии с таблицей истинности функции, а на селективные входы мультиплексов – значение аргументов функции. На выходе мультиплексора формируется значение заданной функции. Подобная методика до сих пор применяется для реализации комбинационной логики в реконфигурируемых цифровых устройствах.

Следующим эволюционным шагом для реализации произвольных переключательных функций стало появление технологии программируемых логических матриц (ПЛМ), в состав которых входили две матрицы И или ИЛИ с программируемыми элементами, которые позволяли получать необходимые термы и дизъюнкции соответствующих термов [Угрюмов, 2004]. Однако, как показала практика матрица ИЛИ в большинстве случаев реализации имела излишний неиспользованный ресурс. Замена программируемой матрицы дизъюнкторов на фиксированную и введение дополнительных выходных макроячеек с трехстабильными буферами привели к появлению программируемых матриц логики (ПМЛ) [Угрюмов, 2004]. Прорывным технологическим новшеством было наличие именно выходных макроячеек, позволявшие управлять выходным значением реализованной функции, переключать выходной порт в режим входного порта, буферизировать вырабатываемое значение при помощи триггера и т.п.

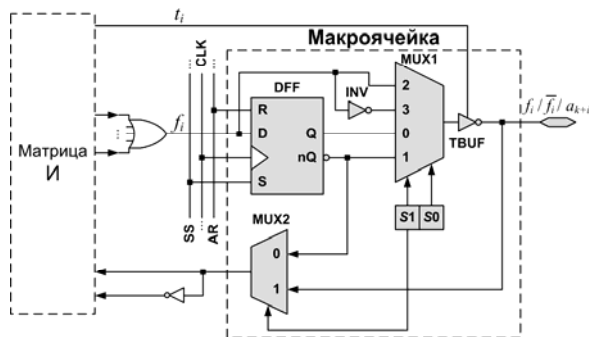


Рисунок 1 – Структура макроячейки AMD 22V10

На рисунке 1 представлена структура макроячейки ПМЛ AMD 22V10, использующая два про-

граммируемых мультиплексора, позволяющие изменять ее аппаратную конфигурацию [PAL, 1986].

В 1985 году корпорация Lattice Semiconductor усовершенствовала изготовление ПМЛ, сделав матрицу И и макроячейки многократно программируемыми, совместив на одном кристалле технологию КМОП и EEPROM. Это нововведение позволило разработчикам использовать ПМЛ для отладки прототипов будущих цифровых устройств, а новая технология ПЛУ получила название GAL (Generic Array Logic). Помимо этого в состав структуры GAL были включены аппаратные блоки внутрисистемного программирования (ISP, In-System Programming), а наличие дополнительных портов граничного сканирования JTAG, соответствующего кабеля и программного обеспечения дало возможность перепрограммировать устройство не извлекая его из состава цифровой системы [ISP].

Появление на рынке интегральных схем GAL фирмы Lattice Semiconductor можно считать точкой отсчета современных реконфигурируемых цифровых устройств. В начале 90-х годов прошлого века совершенствование технологий изготовления полупроводниковых интегральных схем позволило разместить на одном кристалле множество ПМЛ, объединенных программируемыми связями. Такие микросхемы получили название «сложные ПЛУ» или CPLD (Complex Programmable Logic Devices). Структура CPLD состоит из двумерной матрицы функциональных блоков, построенные по принципу перепрограммируемой ПМЛ, состоящей из настраиваемой матрицы И и программируемых макроячеек. В свою очередь каждый функциональный блок соединен с матрицей программируемых соединений и с настраиваемыми блоками ввода-вывода (БВВ). Контроллер внутреннего доступа предназначен для проведения процедур внутрисистемного программирования CPLD и для обеспечения контролепригодного доступа к внутренним ресурсам с целью проведения тестирования, диагностики и отладки программируемого устройства. Практически все внутренние блоки CPLD являются программируемыми, при этом настраиваемость ресурсов обеспечивается энергонезависимой памятью выполненной, как правило, по Flash-технологии. Современные CPLD представляют собой относительно дешевые и многофункциональные энергонезависимые ПЛИС и применяются в основном для реализации цифровых устройств, не требующих частой замены и реконфигурации. Основными областями применения CPLD являются интерфейсные контроллеры, программируемые матрицы коммутаций цифровых линий, устройства управления, арбитры системных и локальных шин микропроцессорных систем, контроллеры периферийных устройств и т.п. Современные CPLD являются достаточно производительными устройствами, позволяющие реализовывать сложные цифровые системы управления и даже пользовательские микроконтроллерные системы. Например, корпорация Xilinx предлагает готовое решение в виде настраиваемого 8-разрядного микроконтроллера PicoBlaze для реализации на CPLD серии

CoolRunner [PicoBlaze]. В свою очередь CPLD CoolRunner-II XC2C512 имеет следующие характеристики: 512 макроячеек, 270 настраиваемых портов ввода/вывода, системная тактовая частота 180 МГц, кристалл изготовлен по 0,18 микронному КМОП техпроцессу, память конфигурации поддерживает порядка 1000 циклов стирания/перепрограммирования, продолжительность хранения конфигурации — около 20 лет.

Одновременно с развитием архитектур ПЛУ трансформировались и совершенствовались архитектуры полужаказных ИС вентиляльных матриц с масочным программированием. Например, базовые матричные кристаллы (БМК), имея регулярную структуру, программировались непосредственно на этапе производства. СБИС БМК представляли собой набор базовых цифровых блоков, при этом настройка связей между ними осуществлялась на этапе трассировки сигнальных линий кристалла. Реализация полужаказной логики подобного типа облегчает процесс их производства, однако не всегда удовлетворяет проектировщиков с точки зрения проектной и эксплуатационной гибкости разрабатываемых цифровых устройств. В 1985 году фирма Xilinx Inc. спроектировала и начала серийное производство СБИС вентиляльных матриц, неоднократное программирование которых осуществлялось пользователями, а не на этапе производства. Этот вид ПЛУ получил название FPGA (Field Programmable Gate Array), что в переводе означает «программируемая пользователем вентиляльная матрица». Появление и дальнейшее развитие технологии FPGA обеспечило большой прорыв в архитектурных решениях современных ПЛУ и привело к возникновению таких терминов, как программируемые системы на кристалле (PSoC, от англ. Programmable System on a Chip) и реконфигурируемые вычисления (Reconfigurable Computing).

2. Современные реконфигурируемые цифровые устройства

Структуры современных ПЛИС типа FPGA являются настолько разнообразными по типам внутренних блоков и обладают настолько высокой степенью интеграции, что их принято относить к классу интегральных схем типа «система на кристалле». На мировом рынке ПЛИС присутствуют два ярко выраженных производителя: Xilinx Inc. (доля мирового рынка 50%) и Altera Inc. (доля мирового рынка 36%) [Тишер, 2009], среди которых лидером, безусловно, является Xilinx Inc. Так, в 2011 году Xilinx Inc. выпустила самую «мощную» ПЛИС типа FPGA Virtex-7 2000T, выполненную по технологии 2,5D SSI (Stacked Silicon Interconnect) с технологическими нормами 28нм и имеющую в своем составе 2 миллиона логических блоков, 6,8 миллиарда транзисторов [Virtex-7]. Число конфигурируемых блоков ввода/вывода равно 1200, максимальный объем встроенных блоков памяти - 85 Мбайт (для серии Virtex-7 XT), пиковая пропускная способность встроенных приемопередатчиков с последователь-

ным интерфейсом составляет порядка 28 Гбит/с, 3600 встроенных блоков цифровой обработки сигналов (DSP Slices) с суммарной пиковой производительностью 5,335 GMAC/s. Каждый конфигурационный логический блок CLB состоит из двух частей Slice, которые в свою очередь состоят из одного 6-входного LUT-блока (Look-Up Table), используемого в качестве генератора переключательных функций шести аргументов, либо в качестве запоминающего устройства с емкостью 64x1, либо в качестве сдвигового регистра. Помимо этого в каждом Slice-блоке имеется восемь конфигурируемых триггеров, набор настраиваемых мультиплексоров и логика ускоренного переноса, используемая в случаях реализации схем быстрых сумматоров и умножителей. Каждый блок встроенной памяти Block RAM представляет собой двухпортовое статическое ОЗУ с информационной емкостью 36Кбит, с возможностью конфигурации размерности шин данных до 72 разрядов. Каждый блок памяти снабжен встроенным средством обнаружения и исправления ошибок и программируемой логикой, позволяющей реализовывать схемы памяти с последовательным доступом (например FIFO).

Такое большое количество разнообразных и конфигурируемых аппаратных блоков позволяет эффективно использовать FPGA в различных приложениях, например: прототипирование ASIC, аэрокосмическая и военная электроника, системы мультимедиа, автомобильная электроника, высокопроизводительные вычисления, медицинская электроника, цифровые системы безопасности и защиты данных, системы проводной и беспроводной связи и т.д. [ХАРР].

В последнее время, помимо самого рынка ПЛИС, стремительно развивается рынок систем быстрого прототипирования (Fast Prototyping Boards). Среди всего многообразия фирм, занимающихся выпуском подобного оборудования, следует отметить фирму Digilent Inc., которая среди прочих услуг предлагает недорогие решения для университетов [Digilent]. В общем прототипная плата представляет собой законченную систему, центральным ядром которой является ПЛИС. Помимо этого система включает в себя набор различных периферийных интерфейсов и контроллеров, модули ОЗУ и энергонезависимой памяти, множество дополнительных контактов предназначенных для подключения других устройств. Кроме того, прототипные системы имеют разнообразные механизмы внутреннего доступа для возможности программирования ПЛИС, проведения диагностики состояний элементов системы, пошаговой отладки системы и т.п.

Наличие прототипных систем и бесплатных систем автоматизированного проектирования цифровых устройств [ISE], позволяющие использовать высокоуровневые HDL-языки, делают современные ПЛИС основной и широкомасштабной технологической платформой для цифровой схемотехники.

3. Реконфигурируемые вычисления

Ресурсов современных ПЛИС типа FPGA более чем достаточно для реализации микропроцессорных ядер. Высокоуровневые HDL-языки проектирования позволяют довольно легко описать структуру и функционирование сложных микропроцессорных систем. Такие HDL-описания носят название *soft-processor*, а их реализация на ПЛИС является микропроцессорным IP-ядром. Одним из первых *soft-processor* был микропроцессор MicroBlaze, представляющий собой 32-разрядное RISC-ядро, построенное по гарвардской архитектуре [MicroBlaze]. Трехступенчатая система обработки потока инструкций совместно с тридцатью двумя 32-разрядными регистрами общего назначения определяли MicroBlaze как достаточно эффективное IP-ядро для встроенных систем. Реализованная компонента MicroBlaze для различных типов FPGA имеет различные аппаратные затраты и показатели производительности. Так, для FPGA серии Spartan, функционирующих на частоте 100 МГц, MicroBlaze достигает пиковой производительности в 92MIPS, а для FPGA серии Virtex-4 производительность оценивается как 166MIPS для частоты 200МГц.

Самое важное, что при наличии исходного описания *soft-processor* проектировщик способен внести различные изменения как в структуру микропроцессорного ядра, так и изменять его функциональность, например путем удаления неиспользуемых или добавления новых инструкций, регистров и т.п. Подобные модификации на стадии проектирования IP-ядер получили название *статической конфигурации*. Статическая конфигурация используется в случаях эффективного применения существующей микропроцессорной системы для решения конкретной задачи либо задач, принадлежащих одному классу. Статическая конфигурация может применяться для расширения функционирования промышленных *soft-процессоров*, например путем увеличения разрядности системных шин, регистров и блоков памяти. А большой объем внутренних ресурсов FPGA может быть использован для реализации многопроцессорных и многоядерных вычислительных систем. Изменение функционирования микропроцессорных IP-ядер со статической конфигурацией возможно путем изменения контента памяти команд либо путем изменения исходного HDL-описания с последующим перепроектированием.

Традиционные типы ПЛИС имеют одноконтекстное ЗУ конфигурации, которое может быть только целиком загружено извне по последовательному интерфейсу. Для изменения хотя бы одного бита конфигурации подобное ЗУ должно быть полностью перезаписано. Для реализации динамической реконфигурации одноконтекстных ПЛИС используют ЗУ с произвольным доступом. Такой подход называется *частичной реконфигурацией* (*partial reconfiguration*) [Partial, a], [Partial, b] и не требует остановки функционирования всей ПЛИС. Примерами подобных ПЛИС могут служить FPGA Atmel 40K и Xilinx 62xx. Однако ПЛИС с одноконтекст-

ным ЗУ не позволяют быстро и эффективно изменять полностью свою конфигурацию. Другими примерами динамически реконфигурируемых систем являются многоконтекстные FPGA (*multicontext FPGA*), для которых каждый программируемый элемент конфигурации представлен несколькими битами. При этом каждый бит принадлежит одному контекстному уровню, что позволяет быстро, без загрузки извне и больших временных затрат, изменять аппаратную настройку FPGA. Так, данная технология нашла свое применение в реконфигурируемой системе Chameleon [Chameleon], СБИС CS2000 RCD которой имеет четыре контекстных уровня. Используя подобные системы, появляется возможность дополнительно загрузить новые контексты извне, не прерывая функционирования ПЛИС, а время переключения контекстов оценивается наносекундами. Наличие подобных технологических решений открывает возможности для исследования и реализации *динамически реконфигурируемых микропроцессорных устройств*.

Для реализации высокопроизводительных систем на кристалле производителя ПЛИС используют технологию «*embedded processor*», заключающуюся в реализации истинного микропроцессорного ядра на одном кристалле с ПЛИС. При этом проектировщик может использовать как ресурсы встроенного процессора, так и реализовывать свои собственные *soft-процессоры* и дополнительные цифровые блоки на свободных ресурсах ПЛИС. Примером технологии «*embedded processor*» могут служить FPGA серии Virtex-4, которые имеют в своем составе до четырех аппаратных ядер микропроцессора PowerPC-405 [Virtex-4].

В 2010 году корпорация Intel пошла схожим путем, выпустив совместно с Altera Inc. первый реконфигурируемый микропроцессор Atom [Intel]. Само микропроцессорное ядро не является реконфигурируемым, но расположено в реконфигурируемой среде, позволяющее сторонним разработчикам самим реализовывать дополнительные аппаратные модули, например интерфейсных контроллеров.

4. Тенденции развития

В настоящее время наблюдается устойчивая тенденция использования ПЛИС не только в качестве технологической платформы для прототипирования ASIC и не только для реализации аппаратных ускорителей, но их использование в качестве самостоятельной технологии реализации программируемых систем на кристалле. Свойства статической и динамической реконфигурации позволяют реализовывать высоконадежные PSoC, проектируемые с учетом наличия аппаратуры оперативного контроля (Concurrent Checking), неразрушающего самотестирования (Transparent Self Testing), неразрушающей самодиагностики (Transparent Self Diagnostics) и саморемонта (Self Repair) [Akoglu, 2009]. Например, технологические особенности современных ПЛИС типа FPGA уже имеют все необходимые аппаратные ресурсы для реализации вышеперечисленных

свойств. Так, наличие встроенных блоков двухпортового ОЗУ позволяют использовать методы самоконтроля и неразрушающего самотестирования состояния памяти данных и инструкций не используя дополнительные аппаратные и временные ресурсы. Гетерогенная структура FPGA и возможность динамической конфигурации коммуникаций и функционала внутренних ресурсов делают возможным реализацию средств саморемонта, при котором осуществляется локализация отказавшего элемента (блока), его изоляция и включение новой копии элемента из числа резервных.

Перспективным направлением использования PSoC является реализация концепции динамически реконфигурируемых вычислительных ядер, которая подразумевает адаптацию вычислительного ядра к потоку исполняемых инструкций с общей целью повышения быстродействия [Karthigaikumar, 2006]. Постепенный переход к данной концепции наблюдается на примерах реализации реконфигурируемых блоков АЛУ микропроцессоров общего назначения для повышения быстродействия при выполнении алгоритмов, относящихся к единому классу задач. Такая методика нацелена на детальное исследование свойств общности подобных алгоритмов, экстракцию наиболее встречаемых общих фрагментов из алгоритмических последовательностей, последующих их синтез и получение соответствующих функциональных блоков, которые затем «внедряются» в реконфигурируемое поле межсоединений АЛУ. Это дает возможность динамически управлять аппаратной конфигурацией АЛУ, при этом управление задается непосредственно из потока инструкций, несущего информацию не только для декодера инструкций, но и информацию о конфигурации АЛУ. Подобные модификации никоим образом не отражаются на производительности микропроцессорного ядра общего назначения, но способны увеличивать его производительность при выполнении определенных задач. Например, подобная возможность динамической конфигурации может быть основой для разработки и исследования новых подходов аппаратно-программной реализации графо-динамических моделей семантических сетей и иных аспектов теории искусственного интеллекта [Голенков, 2001].

Наличие технологии PSoC также открывает новые горизонты для разработчиков супер-ЭВМ. Если FPGA уже применяются при проектировании и реализации динамически коммутируемых межсоединений в кластерных супер-ЭВМ [Awad, 2009], вычислительными ядрами которых по-прежнему являются микропроцессорные системы общего назначения, то следующим шагом может быть использование ПЛИС в качестве динамически реконфигурируемых вычислительных ядер. Это даст возможность не только динамически управлять потоками обрабатываемых данных, но и исполнять соответствующие подзадачи, адаптируя свою аппаратуру для их эффективного решения.

Еще одним перспективным направлением применения ПЛИС можно считать аппаратную (физическую) криптографию. Так, появление в 2001 году теории физически неклонированных функций (PUF, от англ. – Physically Unclonable Functions) [Ruhmair, 2009] способствовало их детальному исследованию благодаря наличию ПЛИС типа FPGA. При этом FPGA является не только экспериментальным оборудованием, посредством которого было исследовано и разработано множество новых видов PUF, но и аппаратной платформой для реализации коммерческих криптостойких систем защиты информации. Также PUF являются перспективными для идентификации интегральных схем, внедрения аппаратных водяных знаков и защиты интегральных схем от несанкционированного клонирования.

ЗАКЛЮЧЕНИЕ

В данном докладе была рассмотрена эволюция реконфигурируемых цифровых устройств от ПЗУ до программируемых систем на кристалле. Показана динамика расширяющегося мирового рынка реконфигурируемых устройств и заинтересованность мировых производителей интегральных схем в масштабном изменении технологий реализации вычислительных устройств. Показано, что статическая и динамическая аппаратная конфигурация является технологической основой для реконфигурируемых вычислений и поиска новых нетрадиционных подходов в реализации вычислительных архитектур.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Mario, 1986] Marino, L.R. Principles of Computer Design / L.R. Marino. — Rockville: Computer Science Press, Inc., 1986. — 578 p.
- [Угрюмов, 2004] Угрюмов, Е. П. Цифровая схемотехника / Е.П. Угрюмов. - БХВ-Петербург, 2004.
- [PAL, 1986] High-Performance IMPACT Programmable Array Logic Circuits (TIBPAL22V10C, TIBPAL22V10AC, TIBPAL22V10AM): technical documentation, Texas Instruments, SRPS02AA — October 1986 – Revised April 2010. — Dallas: Texas Instruments, 2011. — 21 p.
- [ISP] In-System Programming Design Guidelines for ispJTAG Devices: technical documentation, Lattice Semiconductor Corporation [Electronic resource]. — Mode of access : <http://www.latticesemi.com/lit/docs/ispbs/ispguide.pdf>. — Date of access : 18.08.2011.
- [PicoBlaze] PicoBlaze 8-bit Microcontroller / [Electronic resource]. — Mode of access: <http://www.xilinx.com/products/intellectual-property/picoblaze.htm>. — Date of access : 09.01.2012.
- [Тишлер, 2009] Тишлер, М. Россия – очень важный рынок для компании Altera / М. Тишлер // Компоненты и технологии. – 2009. – №9. – С. 6–7.
- [Virtex-7] Virtex-7 FPGA Family / [Electronic resource]. — Mode of access: <http://www.xilinx.com/products/silicon-devices/fpga/virtex-7/index.htm>. — Date of access : 09.01.2012.
- [XAPP] Xilinx FPGA Applications / [Electronic resource]. — Mode of access: <http://www.xilinx.com/applications/index.htm>. — Date of access : 09.01.2012.
- [Digent] Digilent Inc. [Electronic resource]. — Digilent Inc. — 2008. — Mode of access: <http://digilentinc.com>. — Date of access: 25.03.2011.
- [ISE] ISE WebPACK Design Software / [Electronic resource]. — Mode of access: <http://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.htm>. — Date of access : 04.01.2012.

[**MicroBlaze**] MicroBlaze Soft Processor Core / [Electronic resource]. — Mode of access: <http://www.xilinx.com/tools/microblaze.htm>. — Date of access : 09.01.2012.

[**Partial, a**] Dynamic partial reconfiguration on FPGA / [Electronic resource]. — Mode of access: <http://www.vlsi-world.com/content/view/48/47/>. — Date of access : 09.01.2012.

[**Partial, b**] Partial reconfiguration in FPGA rapid prototyping tools / [Electronic resource]. — Mode of access: <http://www.design-reuse.com/articles/24368/partial-reconfiguration-fpga-rapid-prototyping.html>. — Date of access : 09.01.2012.

[**Chameleon**] Chameleon Chips. Full Report / [Electronic resource]. — Mode of access: <http://www.scribd.com/doc/67986721/Chameleon-Chips-Full-Report>. — Date of access : 09.01.2012.

[**Virtex-4**] Virtex-4 Family Overview / [Electronic resource]. — Mode of access: http://www.xilinx.com/support/documentation/data_sheets/ds112.pdf. — Date of access : 09.01.2012.

[**Intel**] Intel Expands Customer Choice with First Configurable Intel Atom-based Processor / K. Temple // Intel Newsroom [Electronic resource]. — 22 Nov., 2010. — Mode of access : http://newsroom.intel.com/community/intel_newsroom/blog/2010/11/22/intel-expands-customer-choice-with-first-configurable-intel-atom-based-processor. — Date of access : 13.06.2011.

[**Akoglu, 2009**] Akoglu, A. FPGA based distributed self healing architecture for reusable systems / A. Akoglu, A. Sreeramareddy, J.G. Josiah // Cluster Computing. — 2009. — Vol. 12, Num. 3. — P. 269 — 284.

[**Karthigaikumar, 2006**] Karthigaikumar, P. A Novel FPGA Architecture for a Reconfigurable ALU / P. Karthigaikumar // Academic Open Internet Journal. — 2006. — Vol. 19. — Mode of access : <http://www.acadjournal.com/2006/v19/part6/p6/>. — Date of access : 09.01.2012.

[**Голенков, 2001**] Программирование в ассоциативных машинах / В.В. Голенков [и др.]; - Минск : БГУИР, 2001.

[**Awad, 2009**] Awad, M. FPGA supercomputing platforms: A survey / M. Awad // Proc. of Int. Conf. of Field Programmable Logic (FPL 2009): 31 Aug., 2009, Prague. — P. 564 – 568.

[**Ruhrmair, 2009**] Ruhrmair, U. On the Foundations of Physical Unclonable Functions / U. Ruhrmair, J. Solter, F. Sehn // Cryptology ePrint Archive [Electronic resource]. — Mode of access : <http://eprint.iacr.org/2009/277.pdf>. — Date of access : 01.09.2011.

CURRENT STATE AND DEVELOPMENT TRENDS OF TECHNOLOGIES OF RECONFIGURABLE DIGITAL DEVICES

Ivaniuk A.A.

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

ivaniuk@bsuir.by

In this article the historical review of evolution of reconfigurable digital devices is presented. Examples of architecture of modern programmed logic integrated schemes and area of their application are described. Questions of realization of computers with a static and dynamic configuration are considered. The review of possible trends of development of reconfigurable digital devices is carried out.

INTRODUCTION

The purpose of the given article is definition of possibilities of modern technologies for designing and creation of digital computers with nonconventional architecture, including dynamically reconfigurable

computers. The wide circulation of programmable logic devices (PLD) together with higher-level HDL-languages gives to almost each designer unlimited possibilities on designing and prototyping of application specific integrated circuits (ASIC). On the other hand, relative cheapness and steady growth of degree of complexity do PLDs by their direct competitors ASIC, allowing to realize, for example, not only models of commercial microprocessors (soft-processors), but also computers with the unique nonconventional architecture, representing objects of intellectual property (IP-cores).

MAIN PART

Fast evolution of reconfigurable digital devices is caused by desire of developers to change functionality of devices without using their repeated designing.

Modern reconfigurable devices are so difficult integrated that it is accepted to name them programmable systems on a chip and they become direct competitors to ASICs.

Modern PLDs are applied to realization of microprocessor IP-cores, and interest of large corporations in use of such technology speaks about their perspective.

One of perspective directions reconfigurable technologies development is working out of superreliable self-repairable digital devices and dynamically reconfigurable microprocessors.

CONCLUSION

In the given article the evolution of reconfigurable digital devices from ROM to programmed systems on a chip has been considered. The dynamics of the extending world market of reconfigurable devices and interest of world manufacturers of integrated circuits in scale change the technologies of realization of computers. It is shown that the static and dynamic hardware configuration is a technological basis for reconfigurable computing and for research in the field of new nonconventional approaches in realization of computing architectures.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

РЕКОНФИГУРИРУЕМЫЕ ВЫЧИСЛЕНИЯ КАК СРЕДСТВО ОПТИМИЗАЦИИ РЕШЕНИЙ ПОДКЛАССА ЗАДАЧ

Абрамов Н.В. *, Иванюк А.А. *

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

nickolaib2004@gmail.com

ivaniuk@bsuir.by

Решение любой задачи можно представить в виде алгоритма. Алгоритм задает последовательность шагов. В некотором смысле алгоритм можно рассматривать как правила, которым нужно следовать для достижения конкретной цели. Однако правила сами по себе совершенно бесполезны при отсутствии некоторого объекта, который мог бы выполнять эти правила и тем самым получать некоторый результат.

Ключевые слова: реконфигурируемые вычисления, помехоустойчивое кодирование, обобщение подкласса задач.

Введение

Алгоритм можно представлять различным образом и с различной детализацией, при чем, уровень детализации должен быть достаточным для того чтобы исполнитель мог выполнить абсолютно все действия предписанные алгоритмом. Если в описании алгоритма будет иметь место хотя бы одно непонятное действие, алгоритм корректным являться не будет. Другими словами алгоритм и исполнитель алгоритма должны оперировать общими, понятными друг другу терминами. Применительно к вычислительной технике можно сказать, что например, невозможно выполнить некоторый алгоритм, представленный в виде блок-схем, или в виде текстового описания. Микропроцессоры не умеют оперировать такими сложными объектами. Поэтому в конечном итоге любое описание в любом виде, проецируется на набор инструкций, поддерживаемый конкретным микропроцессором. Однако не все задачи могут быть эффективно представлены в виде программы. В этом случае использование реконфигурируемых вычислений добавляет существенно больше возможностей для отображения алгоритмов на вычислительные устройства. Эффективное отображение прикладной области на реконфигурируемое устройство можно выполнить при помощи онтологий, другими словами необходимо провести формализацию предметной области и представить ее в виде семантической сети.

1. Микропроцессор общего назначения

В качестве примера микропроцессора общего назначения рассмотрим виртуальный MIPS процессор WinMIPS64. Имитатор данного процессора находится в открытом доступе, более подробную информацию о нем можно найти в [Hennesy, 2006]. WinMIPS64 – 64 битный RISC процессор, с конвейерной обработкой команд. Структура конвейера данного микроконтроллера представлена на рисунке 1. Конвейер состоит из пяти основных шагов.

IF (Instruction Fetch) шаг на котором происходит извлечение инструкции и памяти программы в регистр инструкции. При этом счетчик команд увеличивается на единицу.

ID (Instruction Decode) шаг, на котором имеет место декодирование инструкции извлечение двух операндов, которые подаются на вход АЛУ. Кроме того, на данном шаге выполняются операции ветвления.

EX (Execution) шаг предназначен для выполнения некоторых действий над операндами, которые были поданы на шаге ID в АЛУ. В зависимости от кода инструкции на данном шаге может иметь место сложение, вычитание, умножение, сдвиг, работа с вещественными числами и т.д.

MEM (MEMoryAccess) шаг, на котором выполняются инструкции по работе с памятью, такие как ld, sd, lb, sb и т.д. С полным набором поддерживаемых инструкций можно ознакомиться в [Scott, 2003].

WB (Write Back) шаг, на котором данные полученные из памяти или как результат вычислений в АЛУ, записываются в регистр назначения. Запись данных на этой стадии происходит в течение первой половины такта, таким образом, на второй половине такта на стадии ID можно прочитать новое значение регистра.

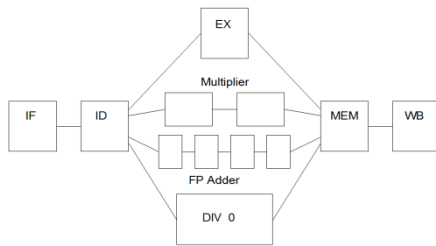


Рисунок 1 - Структура конвейера микропроцессора WinMIPS64

Данные пять ступеней конвейера, и заданный набор инструкций позволяют реализовать практически любую задачу.

2. Применение микропроцессора общего назначения для решения задач

Рассмотрим три алгоритма из области помехоустойчивого кодирования: использование бита четности, код Хэмминга, циклический код.

Данные алгоритмы основаны на внесении избыточных проверочных бит в сообщение, что позволяет в случае ошибок в системе связывосстанавливать потерянные данные или, по крайней мере, сообщать об имеющейся в них ошибке [Блейхут, 1986].

Первый рассматриваемый алгоритм – использование бита четности для определения наличия ошибки нечетной кратности.

Имея информационное сообщение $m = \{i_3 i_2 i_1 i_0\}$ можно вычислить бит четности по следующей формуле:

$$p = i_3 \oplus i_2 \oplus i_1 \oplus i_0$$

Совмещая информационное сообщение с битом четности, передаваемое сообщение можно представить в виде $c' = \{i_3 i_2 i_1 i_0 p\}$. Расширяя сообщение до одного байта, по средствам добавления нулей в старшие разряды получаем окончательный код $c = \{000 i_3 i_2 i_1 i_0 p\}$. Восемь битный код подходит для хранения в регистрах большинства микропроцессоров.

Программа для микропроцессора MIPS64, формирующая кодовое сообщение c , представлена в листинге 1.

```

1: ;Листинг 1 - Бит четности
2: .data
3: Message: .word 7
4: Count: .word 8
5: Result: .word 0
6: Mask: .word 1

```

```

7: MaskBit1: .word 1
8: MaskBit2: .word 2
9: MaskBit3: .word 4
10: MaskBit4: .word 8
11: .text
12: main:
13: ;r4 - исходное сообщение
14: ld r4, Message(r0)
15: ;r5 - счетчик
16: ld r5, Count(r0)
17: ;r6 - маска
18: ld r6, Mask(r0)
19: ;r7 - результат
20: ld r7, Result(r0)
21: ;определяем информационные биты
22: ld r6, MaskBit1(r0)
23: and r8, r4, r6
24: beqz r8, skip_i1
25: ld r9, Mask(r0)
26: skip_i1:
27: ld r6, MaskBit2(r0)
28: and r8, r4, r6
29: beqz r8, skip_i2
30: ld r10, Mask(r0)
31: skip_i2:
32: ld r6, MaskBit3(r0)
33: and r8, r4, r6
34: beqz r8, skip_i3
35: ld r11, Mask(r0)
36: skip_i3:
37: ld r6, MaskBit4(r0)
38: and r8, r4, r6
39: beqz r8, skip_i4
40: ld r12, Mask(r0)
41: skip_i4:
42: xor r13, r9, r10
43: xor r13, r13, r11
44: xor r13, r13, r12
45: dsll r4, r4, 1
46: xor r6, r4, r13
47: halt

```

Очевидно, можно привести множество других реализаций, однако был выбран именно такой способ, для демонстрации сходства реализаций алгоритмов одной предметной области.

Программа выполняет следующие действия. Исходное сообщение помещается в регистр r4. Затем в регистры r9, r10, r11, r12 помещаются биты данного сообщения. Имея отдельные биты сообщения нетрудно вычислить их сумму по модулю два используя инструкцию xor. Определив бит четности, формируется окончательное сообщение по средствам сдвига исходного и добавления бита четности в младший разряд при помощи инструкции xor.

Данная микропрограмма состоит из 25 инструкций. Так как обработка команд происходит конвейерным образом, это приводит к тому, что одновременно в микропроцессоре выполняется до пяти команд, находящихся на различных этапах конвейера. Такая псевдопараллельная работа приводит к тому, что 25 инструкций, с учетом того, что на одну команду требуется 5 тактов, выполняются за 38 тактов, а не за 125, если бы имела места последовательная обработка команд. Таким образом, на одну команду в данной микропрограмме было затрачено в среднем 1.52 такта.

Второй алгоритм – код Хэмминга (7,4). Данный код позволяет исправлять ошибку единичной кратности. Общая длина кода составляет семь бит.

Среди этих бит четыре бит информационные и три бита проверочные.

Имея сообщение m , проверочные биты можно вычислить по следующей формуле:

$$\begin{aligned} r_0 &= i_0 \oplus i_1 \oplus i_3; \\ r_1 &= i_0 \oplus i_2 \oplus i_3; \\ r_2 &= i_1 \oplus i_2 \oplus i_3. \end{aligned}$$

Определив проверочные биты можно сформировать кодовое слово $c' = \{i_3 i_2 i_1 i_0 r_2 r_1 r_0\}$, добавляя 0 в старший разряд получаем представление кода размером в один байт $c = \{0 i_3 i_2 i_1 i_0 r_2 r_1 r_0\}$. Программа для микропроцессора MIPS64, формирующая кодовое сообщения c используя код Хэмминга (7,4), представлена в листинге 2.

```

1: ;Листинг 2 - Код Хэмминга
2: .data
3: Message: .word 13
4: Count: .word 8
5: Result: .word 0
6: Mask: .word 1
7: MaskBit1: .word 1
8: MaskBit2: .word 2
9: MaskBit3: .word 4
10: MaskBit4: .word 8
11: .text
12: main:
13: ld r4, Message(r0)
14: ld r5, Count(r0)
15: ld r6, Mask(r0)
16: ld r7, Result(r0)
17: ld r8, Result(r0)
18: ld r9, Result(r0)
19: ld r10, Result(r0)
20: ld r11, Result(r0)
21: ld r12, Result(r0)
22: ld r6, MaskBit1(r0)
23: and r8, r4, r6
24: beqz r8, skip_i1
25: ld r9, Mask(r0)
26: skip_i1:
27: ld r6, MaskBit2(r0)
28: and r8, r4, r6
29: beqz r8, skip_i2
30: ld r10, Mask(r0)
31: skip_i2:
32: ld r6, MaskBit3(r0)
33: and r8, r4, r6
34: beqz r8, skip_i3
35: ld r11, Mask(r0)
36: skip_i3:
37: ld r6, MaskBit4(r0)
38: and r8, r4, r6
39: beqz r8, skip_i4
40: ld r12, Mask(r0)
41: skip_i4:
42: ;определяем проверочные биты
43: xor r13, r9, r10
44: xor r13, r13, r12
45: xor r14, r9, r11
46: xor r14, r14, r12
47: xor r15, r10, r11
48: xor r15, r15, r12
49: ;формируем закодированное сообщение
50: dsll r6, r4, 3
51: dsll r14, r14, 1
52: dsll r15, r15, 2
53: xor r8, r13, r14
54: xor r8, r8, r15
55: xor r6, r6, r8
56: halt

```

Анализируя листинг программы 2, можно выявить некоторые ее сходства с листингом программы 1. Различие заключается в вычислении проверочных бит и формировании кодового слова. Участки кода по извлечению информационных бит из регистра идентичны.

Данная микропрограмма состоит из 37 инструкций. Время выполнения программы составило 50 тактов. Таким образом, на одну команду пришлось 1.351 такта.

Третий алгоритм – циклический код [Уильямс, 1993]. Для вычисления циклического кода необходимо задать порождающий простой полином. В данной работе, для того, чтобы можно было воспользоваться тем же самым исходным сообщением, как и в предыдущих двух алгоритмах, используется полином четвертой степени $p(x) = x^4 + x + 1$. Имея простой полином можно построить порождающую матрицу, при помощи которой можно рассчитать код для заданного сообщения. Порождающая матрица для данного полинома имеет следующий вид:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Используя порождающую матрицу можно вычислить код следующим образом:

$$c = m \cdot G$$

Обратив внимание на то, что в левой части матрицы находится единичная матрица, можно упростить задачу вычисления кода, рассчитывая только проверочные биты по следующей формуле:

$$\begin{aligned} r_0 &= i_3 \oplus i_0; \\ r_1 &= i_3 \oplus i_2 \oplus i_0; \\ r_2 &= i_2 \oplus i_1; \\ r_3 &= i_1 \oplus i_0. \end{aligned}$$

Имея проверочные биты можно сформировать кодовое слово $c = \{i_3 i_2 i_1 i_0 r_3 r_2 r_1 r_0\}$.

Микропрограмма, формирующая кодовое сообщение c , используя алгоритм формирования циклического кода, представлена в листинге 3.

```

1: ;Листинг 3 - Циклический код
2: .data
3: Message: .word 13
4: Count: .word 8
5: Result: .word 0
6: MaskBit1: .word 1
7: MaskBit2: .word 2
8: MaskBit3: .word 4
9: MaskBit4: .word 8
10: Mask: .word 1
11: .text
12: main:
13: ld r4, Message(r0)
14: ld r5, Count(r0)
15: ld r6, Mask(r0)
16: ld r7, Result(r0)
17: ld r6, MaskBit1(r0)
18: and r8, r4, r6

```

```

19: beqz      r8, skip_i1
20: ld        r9, Mask(r0)
21: skip_i1:
22: ld        r6, MaskBit2(r0)
23: and       r8, r4, r6
24: beqz      r8, skip_i2
25: ld        r10, Mask(r0)
26: skip_i2:
27: ld        r6, MaskBit3(r0)
28: and       r8, r4, r6
29: beqz      r8, skip_i3
30: ld        r11, Mask(r0)
31: skip_i3:
32: ld        r6, MaskBit4(r0)
33: and       r8, r4, r6
34: beqz      r8, skip_i4
35: ld        r12, Mask(r0)
36: skip_i4:
37: xor       r13, r9, r12
38: xor       r14, r12, r10
39: xor       r14, r14, r9
40: xor       r15, r11, r10
41: xor       r16, r12, r11
42: dsll     r6, r4, 4
43: dsll     r14, r14, 1
44: dsll     r15, r15, 2
45: dsll     r16, r16, 3
46: xor       r8, r14, r15
47: xor       r8, r8, r16
48: xor       r8, r8, r13
49: xor       r6, r8, r6
50: halt

```

Микропрограмма, приведенная в листинге 3, состоит из 33 инструкций, которые выполняются на микропроцессоре MIPS64 за 47 тактов. Таким образом, на выполнение одной инструкции в среднем требуется 1.424 такта.

Имея три реализации можно заметить явные схожие черты у них. Можно оценить среднее время выполнения команды по трем реализациям – 1.432 такта. Кроме того можно легко заметить недостаток микропроцессора MIPS64 для выполнения некоторых действий. В частности извлечение конкретного бита из регистра общего назначения требует четыре инструкции, одна из которых условный переход, который приводит к сбросу конвейера:

```

1: ld        r6, MaskBit1(r0)
2: and       r8, r4, r6
3: beqz      r8, skip_i1
4: ld        r9, Mask(r0)
5: skip_i1:

```

Реализовав данную операцию в виде одной команды можно как минимум увеличить скорость работы данного участка кода на 4.296 такта.

Однако возникает закономерный вопрос, каким образом можно добавить желаемую функциональность, наибольшим образом подходящую для реализуемого алгоритма?

3. Реконфигурируемые вычисления

Внедрение реконфигурируемых вычислений позволяет добиться желаемого решения. Приведем листинг программы расчета бита четности в том виде, в котором нам бы хотелось его видеть.

```

1: ;Листинг 4 - Бит четности
   (реконфигурация)
2: .data
3: Message:      .word 7
4: Count:        .word 8
5: Result:       .word 0
6: Mask:         .word 1
7: .text
8: main:
9: ld            r4, Message(r0)
10: ld           r5, Count(r0)
11: ld           r6, Mask(r0)
12: ld           r7, Result(r0)
13: exbt        r9, r4, 0
14: exbt        r10, r4, 1
15: exbt        r11, r4, 2
16: exbt        r12, r4, 3
17: xor         r13, r9, r10
18: xor         r13, r13, r11
19: xor         r13, r13, r12
20: dsll        r4, r4, 1
21: xor         r6, r4, r13
22: halt

```

В листинге 4 содержится инструкция `exbt` (**extract bit**), которая по формату полностью соответствует формату команд принятому в данном микропроцессоре, то есть первый операнд задает приемник, второй и третий операнды задают операнды, подаваемые на входы АЛУ. Результатом работы данной инструкции, является занесение указанного в третьем операнде бита, второго операнда в младший разряд первого. Другими словами имеет место извлечение указанного бита из указанного регистра.

Реконфигурируемые вычисления позволяют изменять архитектуру устройства в процессе выполнения за счет перепрограммирования внутренних конфигурационных ячеек. Конфигурационные ячейки управляют связями между базовыми элементами чипа, такими как вентили «и-не», «или», «и» и т.д., или такими как LUT-таблицы, которые могут функционировать как память или логическая функция. Другими словами реконфигурируемые устройства поддерживают определенные средства для модификации своей архитектуры в процессе работы [Compton, 2002], [Bobda, 2007].

Одним из достоинств реконфигурируемых устройств является их способность реализовать практически любую требуемую архитектуру. Можно без изменений взять описание микропроцессора MIPS64 и отобразить его на реконфигурируемое устройство, и оно станет функционировать согласно той логике, которая предусмотрена данной архитектурой. Очевидно, что изначальное описание не предусматривает наличие таких инструкций как `exbt` и какую-либо реконфигурацию.

Однако, имея описание микропроцессора на высокоуровневом языке описания аппаратуры, таком как VHDL, можно модифицировать некоторые элементы микропроцессора, таким образом, чтобы он мог воспользоваться возможностями, предоставляемыми реконфигурируемым устройством.

При использовании динамической реконфигурации кроме некоторого набора команд возникает гораздо более широкое понятие – конфигурация. При чем, различные конфигурации могут предоставлять различные наборы команд, которые в свою очередь могут быть использованы в одной и той же программе. Принцип работы схож с принципом работы с памятью, когда при работе с памятью используются одни и те же адреса, но в различных банках памяти.

Для достижения данного эффекта команды, которыми оперирует программист можно разбить на два класса: команды реконфигурации, обычные команды. При выполнении такой программы, вычислительное устройство при обработке команды реконфигурации выполняет перепрограммирование реконфигурируемой части устройства, а при обработке обычных команд – передает управление текущей конфигурации.

Очевидно, что переключение конфигурации достаточно длительный процесс. При реконфигурации необходимо дождаться завершения инструкций находящихся в конвейере, так как реконфигурация фактически изменяет элементы конвейера. Кроме того необходимо время на собственное изменение внутренних конфигурационных ячеек устройства. Таким образом, в целом процесс может затянуться. На сегодняшний день существует несколько реконфигурируемых архитектур, поддерживающие достаточно быструю реконфигурацию, например адаптивная архитектура описанная в [Watson, 2002].

Принимая во внимание вышенаписанное, и, имея реконфигурируемый процессор с двумя конфигурациями, листинг 4, может быть переписан следующим образом:

```

23: ;Листинг 5 - Бит четности
    (реконфигурация)
24: .data
25: Message:      .word 7
26: Count:       .word 8
27: Result:      .word 0
28: Mask:       .word 1
29: .text
30: main:
31: ld          r4, Message(r0)
32: ld          r5, Count(r0)
33: ld          r6, Mask(r0)
34: ld          r7, Result(r0)
35: conf      1
36: exbt       r9, r4, 0
37: exbt       r10, r4, 1
38: exbt       r11, r4, 2
39: exbt       r12, r4, 3
40: conf      0
41: xor        r13, r9, r10
42: xor        r13, r13, r11
43: xor        r13, r13, r12
44: dsll      r4, r4, 1
45: xor        r6, r4, r13
46: halt

```

В листинге 4, инструкция `conf`, относится к категории инструкций конфигурации, и приводит к переключению текущей конфигурации на указанную в единственном операнде. При этом инструкция `exbt` может иметь такой же код как,

например, и инструкции `hog`, что не приведет ни к каким конфликтам, так как мнемоника инструкции интересна только программисту, а функциональность при таком подходе определяется как конфигурацией, так и инструкцией.

Упрощенная логическая структура такого конвейера представлена на рисунке 2.

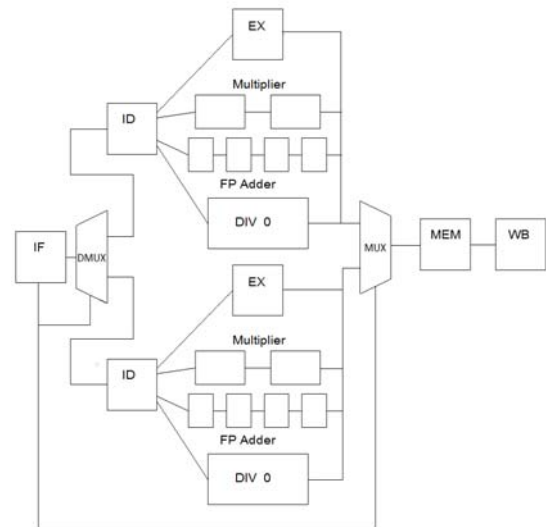


Рисунок 2 - Реконфигурируемый конвейер

В данной архитектуре на этапе извлечения команды определяется необходимая конфигурация, которая определяет декодер и АЛУ используемые для дальнейшей обработки инструкций.

На рисунке 3 приведены размеры программ с использованием и без использования реконфигурации.

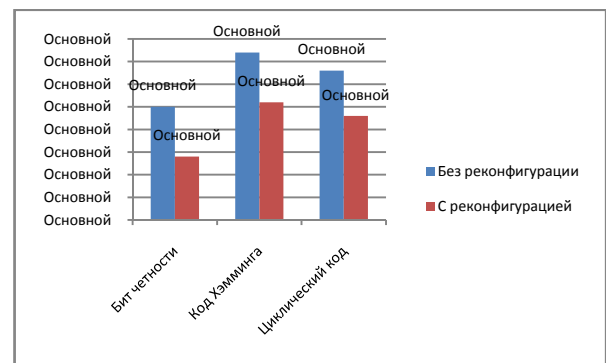


Рисунок 3 - Сравнение размеров программ без и с использованием реконфигурации

На рисунке 4 приведено сравнение времени выполнения в тактах приведенных выше программ с и без использования реконфигурации.

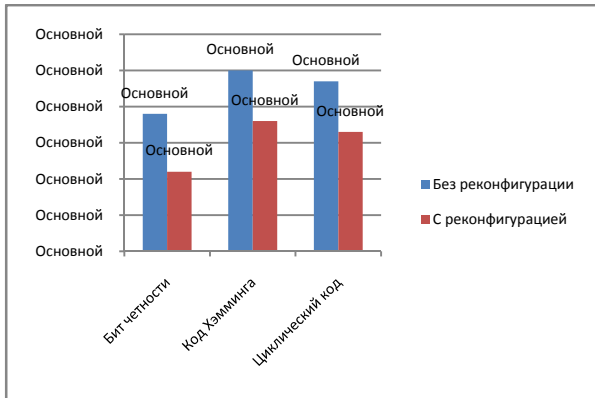


Рисунок 4 - Время выполнения программы без и с использованием реконфигурации

ЗАКЛЮЧЕНИЕ

Внедрение реконфигурируемых вычислений вносит существенную гибкость в систему, позволяя для различных подзадач приложения выделять специализированные аппаратные элементы, которые, за счет динамической реконфигурации могут использовать одни и те же ресурсы.

Для конечного пользователя такого устройства, добавляется необходимость определять требуемую конфигурацию и следить, за тем, какие инструкции используются в выбранной конфигурации. Однако та гибкость, и возможность оптимизировать специфические участки предметной области, делают вполне приемлемыми новые требования к написанию программ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Hennessy, 2006] Computer Architecture: A Quantitative Approach, 4th Edition / Hennessy – Morgan Cauffman, 2006 – 704 p.
- [Scott, 2003] Using WinMIPS64 Simulator – Электронный ресурс: Режим доступа: <ftp://ftp.computing.dcu.ie/pub/resources/crypto/winmipstut.doc>
- [Блейхут, 1986] Теория и практика кодов, контролирующих ошибки / Р. Блейхут. – М.: Мир, 1986. 576 с.
- [Уильямс, 1993] Элементарное руководство по CRC-алгоритмам обнаружения ошибок / Р. Уильямс. – 1993. 36 с.
- [Compton, 2002] Reconfigurable Computing: A Survey of Systems and Software, / ACM Computing Surveys, June 2002, pp. 171-210.
- [Bobda, 2007] Introduction to Reconfigurable Computing. Architectures, Algorithms and Applications / C. Bobda – Dordrecht: Springer, – 2007 – 375 p.
- [Watson, 2002] Adaptive Computing IC Technology Enables SDR and Multi-functionality in next-generation wireless devices / J. Watson // SDR 02 Technical Conference and Product Exposition, – 2002.

RECONFIGURABLE COMPUTING AS MEANS OF SOLUTIONS' OPTIMIZATION OF TASKS OF COMMON FIELD

Abramov N. V.*, Ivaniuk A.A.*

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

nickolaib2004@gmail.com, ivaniuk@bsuir.by

The solution to any problem can be represented in the form of the algorithm. The algorithm defines a sequence of steps. The algorithm can be regarded as rules to follow to achieve a particular goal. However, the rules themselves are quite useless in the absence of an object that could meet these rules and thereby obtain a result.

Introduction

The algorithm can be represented in different ways and with varying detail, the level of detail is determined by executor, because it might be able to do absolutely all actions prescribed by the algorithm. If there is at least one incomprehensible act in the description of the algorithm, it will not be considered correct. In other words, the algorithm and its executor must operate on common, understandable to each other terms. With the respect to computer technology we can say that for example, it is impossible to perform a certain algorithm, presented in the form of block diagrams, or as a text description. Microprocessors are not able to handle such complex objects. Therefore, the final outcome of any description in any form is projected onto the set of instructions supported by a specific microprocessor. However, not all problems can be effectively represented in the form of the program. In this case, the use of reconfigurable computing adds significantly more opportunities to map algorithms on computing devices. Efficient mapping of applied problems on reconfigurable device can be performed using ontology, in other words it is necessary to formalize the domain and present it as a semantic network.

Main Part

This paper discusses the use of reconfigurable computing to optimize the solutions belonging to the same domain. Examples are taken from the field of antinoise coding: parity bit check, Hamming code (7,4), CRC code.

For each selected algorithm source code for microprocessor simulator MIPS64 is listed.

Using statistics provided by algorithm realization common parts of each realization is determined, and using this information an optimization for each of them is suggested. The solution is based on the use of reconfigurable computing. Reconfiguration is introduced in the source pipeline of the processor to fit better the algorithm.

Conclusion

Reconfigurable computing introduces significant flexibility into the system, providing specialized hardware elements for specific subtasks of the program that, due to the dynamic reconfiguration can even share same resources.

The use of the reconfigurable device for the end user is added up to the selection of the appropriate configuration.



УДК 004.93, 004.272

ПАРАЛЛЕЛЬНЫЕ ПРОЦЕССОРЫ ДЛЯ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

С.А.Байрак*, Д.Н.Одинец*, М.М.Татур*, Ф.Филипов**, Марио Мунос***

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

tatur@i-proc.com

dima_odin@i-proc.com

info@i-proc.com

** *Национальный Институт Геофизики, Геодезии, Географии-Болгарская Академия Наук, г. София, Болгария*

philip.philipoff@gmail.com

Мадридский Университет имени Карлоса III, г. Мадрид, Испания

munozm@it.uc3m.es

В настоящей работе изложены результаты научного исследования, практическим итогом которого явилось создание прототипа параллельного процессора, ориентированного на решение задач классификации. Опыт разработки обобщен в виде методологии проектирования проблемно-ориентированных процессоров.

Ключевые слова: параллельный процессор, задача классификации, распознавание образов.

Введение

В настоящей работе под «интеллектуальными вычислениями», будем понимать распознавание образов в различных областях, например, распознавание изображений, звуковых либо электромагнитных сигналов, динамических характеристик (графиков) и т.п. Образы, как правило, представлены в виде слабоструктурированных данных информативных признаков, состав которых определяется уникально для каждой прикладной системы. Образ – это абстрактный вектор, который характеризуется лишь типом данных (количественные, логические, порядковые, номинальные), форматом представления данных и естественно – размером n .

Интеллектуальным ядром системы распознавания является классификатор, принимающий решения – к какому классу объектов принадлежит входной образ. Обычно, схему формального вычислительного процесса классификации изображают в виде полносвязного двудольного графа, рис.1., где n – число обрабатываемых признаков x_i , а k – число классов Z_j . Способы решения задач классификации различаются применяемым алгоритмом или

математическим методом (моделью) - III, среди которых наиболее известными являются классификация по минимуму расстояния между образами, нейронные сети, нечеткий вывод и др. [Кохонен, 2008].

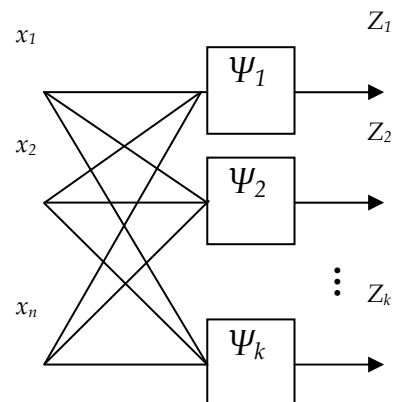


Рисунок 1 – Граф вычислений для задач классификации

Очевидно, что при построении «интеллектуальных» информационных систем с применением обычных РС с последовательной архитектурой существует проблема обеспечения

режима реального времени в тех случаях, когда n и k значительно возрастают, а необходимое время обработки - ограничено. Возможные пути повышения производительности - это применение унифицированных параллельных компьютеров (графических процессоров, суперкомпьютеров и т.п.) либо создание специализированных компьютеров. В первом случае ограничения на модель III не накладываются, но тогда сложность и стоимость процессорного элемента - велики. Во втором случае - наоборот, за счет ограничений на функционал III оказывается возможным существенно сократить сложность процессорного элемента и реализовать десятки и сотни процессорных элементов в одном чипе. Наиболее часто исследователи в качестве функционала III используют модель какой-либо нейронной сети. Поэтому рынок электроники откликнулся созданием целого ряда нейрочипов и нейрокомпьютеров на их основе [Аляутдинов и др., 2008]. Но эффективность нейросетевого hardware оказалась не столь высокой как ожидалось, и поэтому при всей известности термина «нейрокомпьютер» его редко используют в прикладных системах, особенно в системах медицинского и специального применения.

В числе основных сдерживающих факторов массового применения нейрокомпьютеров в «интеллектуальных» вычислениях можно назвать следующие:

- нейросетевые классификаторы по сути являются «черными ящиками», в которых не решена проблема «определения глобального минимума», а входы настроек не обладают физическим смыслом, т.е. являются не интерпретируемыми [Том и др., 2008];

- нейросетевые классификаторы, как правило, имеют многоуровневую, нерегулярную и не стандартизованную топологию, как следствие существует проблема непосредственной либо виртуальной коммутации процессорных элементов (формальных нейронов) в произвольную сеть [Кирсанов, 2004];

- разработчику сложно отобразить (запрограммировать) реализуемую нейронную сеть на заданной архитектуре нейрокомпьютера, получить экспресс-оценки характеристик системы по точности классификации и производительности для используемой аппаратной платформы.

Как результат, известные нейрокомпьютеры не позволяют существенно сократить время и стоимость разработки интеллектуальной системы в целом, скорее отдельные нейрокомпьютеры создаются под конкретную прикладную задачу, тогда экономический фактор, естественно не является преобладающим. С учетом сделанных замечаний, представляется актуальной проблема создания специализированных параллельных компьютеров для решения задач интеллектуальных задач классификации образов, альтернативных классическим нейрокомпьютерам. При этом, выдвигаются следующие основные показатели эффективности:

- унификация в решении различных задач

классификации;

- интерпретируемость процесса обучения и результатов классификации;

- возможность наращивания и гибкого изменения числа информативных признаков - n и числа классов - k ;

- возможность увеличения производительности за счет каскадного включения процессорных элементов;

- низкая стоимость и малое время разработки завершённой интеллектуальной системы на базе параллельного процессора.

Очевидно, что первый и второй показатели будут определяться математической моделью классификации, а третий и четвертый - архитектурными особенностями процессора. Пятый показатель является интегральным следствием от первых четырех.

В работах [Tatur et al., 2010], [Татур и др., 2010a] предложена и исследована модель классификации (Z -модель), которая обладает необходимыми свойствами, чтобы быть положенной в основу таких параллельных процессоров, а в [Татур и др., 2010b] определено место данной модели среди известных методов классификации.

В настоящей работе излагаются результаты построения и испытаний экспериментального образца параллельного процессора-классификатора, основанного на указанной модели. Рассматриваются способы его применения в различных областях. При этом необходимо принять во внимание, что по сути реализованного математического метода рассматриваемая архитектура не рассчитана на отображение какой-либо нейросети, поэтому термины «нейрочип-нейропроцессор-нейрокомпьютер» будут неточно соответствовать сути предлагаемых технических решений. Скорее речь пойдет о нейроподобном (нейронечетком) процессоре, основанном на оригинальной математической идентификации (условно, Z -модели), поэтому в дальнейшем в работе будем называть его Z -процессором.

1. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ИДЕНТИФИКАЦИИ (Z -МОДЕЛЬ)

В одной из своих работ Т.Кохонен обращал внимание на возможность значительного улучшения классификации за счет использования всех доступных априорных знаний: - «Во многих случаях ... на основе физических или других свойств наблюдаемых величин, их распределение имеет ту или иную определенную форму. В результате становится возможным получать математические выражения для дискриминантных (разделяющих) поверхностей, оптимальных со статистической точки зрения». Авторы полностью разделяют эту точку зрения и в своих работах [Tatur et al., 2010], [Татур и др., 2010a], [Татур и др., 2010c] предложили объединить линейную дискриминантную функцию с элементами нечеткого вывода, тем самым обеспечив возможность записи априорных данных о задаче в обобщенной модели идентификации.

Ниже приведен аннотированный вариант описания модели. Назовем:

$$Z = F(X, Y), \quad (1)$$

идеальным классификатором, где X – вектор входных информативных признаков; Y – вектор констант – настроек классификатора для решения конкретной задачи, причем функционал F и настройки Y – априори известны.

Тогда разработка алгоритма будет сведена к настройке либо уточнению параметров Y некоторой модели идеального классификатора (т.е. к обучению). В качестве обобщенной математической модели идентификации предложен функционал (2):

$$Z = Q \cdot L, \quad (2)$$

где

$$Q = \begin{cases} 1, & S(X) > p_2 \\ \frac{S(X) - p_1}{p_2 - p_1}, & p_1 \leq S(X) \leq p_2, \\ 0, & S(X) < p_1 \end{cases}$$

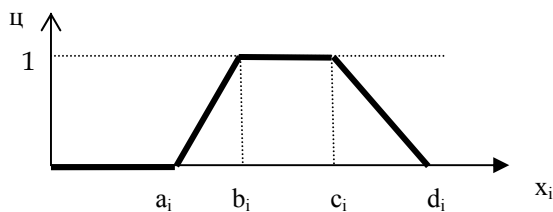
$$S(X) = \sum_{i=1}^n w_i \varphi(x_i, a_i, b_i, c_i, d_i),$$

$$\varphi(x_i, a_i, b_i, c_i, d_i) = \begin{cases} 0, & a_i > x_i, x_i > d_i \\ \frac{d_i - x_i}{d_i - c_i}, & c_i \leq x_i \leq d_i \\ 1, & b_i < x_i < c_i \\ \frac{x_i - a_i}{b_i - a_i}, & a_i \leq x_i \leq b_i \end{cases}$$

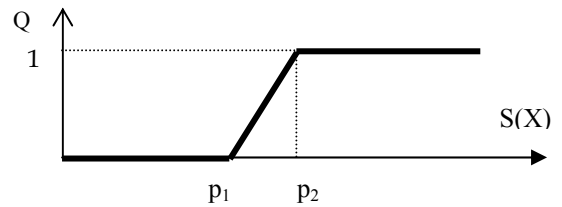
$$L = \min_{i=1}^n \varphi'(x_i, a_i, b_i, c_i, d_i)$$

$$\varphi'(x_i, a_i, b_i, c_i, d_i) = \begin{cases} \varphi(x_i, a_i, b_i, c_i, d_i), & i \in N' \\ 1, & i \notin N' \end{cases}$$

где a_i, b_i, c_i, d_i – параметры линейно-аппроксимированной функции принадлежности ($i=1, \dots, n$):



w_i – вес i -го признака в принятии решения;
 N' – подмножество ключевых признаков;
 p_1, p_2 – параметры линейно-аппроксимированного нечеткого порога:



Графически модель идентификации может быть представлена, как показано на рис.2.

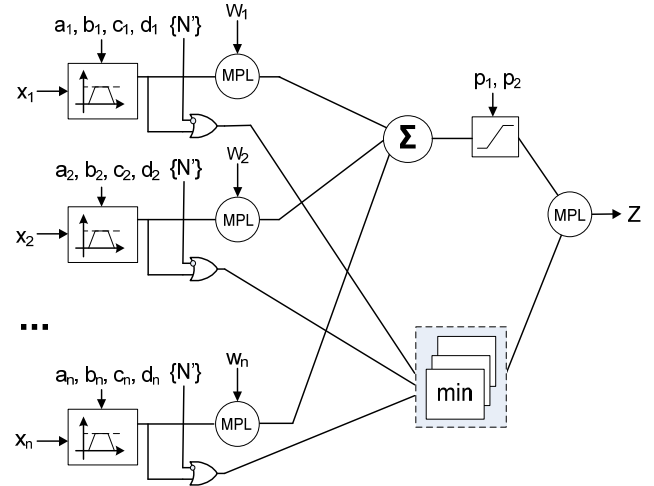


Рисунок 2 – Граф-схема обобщенной модели идентификации

Все настройки модели $W, A, B, C, D, N', p_1, p_2$, в соответствии с принятым обозначением являются подмножеством вектора $Y(Y')$. Синтез алгоритмов классификации заключается в составлении перечня априори известных сведений (зависимостей, ограничений) о прикладной задаче. Настройки модели обладают физическим смыслом (являются интерпретируемыми), поэтому наряду с формальным обучением, классификаторы на основе Z -модели могут настраиваться экспертом. А за счет нечеткой параметризации информативных признаков достигается нелинейность формируемых разделяющих поверхностей. Таким образом, предложенная модель обладает с одной стороны необходимыми функциональными свойствами, а с другой стороны регулярным вычислительным графом, что позволяет положить ее в основу унифицированной архитектуры параллельного процессора.

2.АРХИТЕКТУРА Z-ПРОЦЕССОРА

Архитектура процессора разрабатывалась с учетом решения целого ряда оптимизационных задач и представлена на рис.3. [Татур и др., 2010с]. В целом, архитектура относится к SIMD-типу (Single Instruction Multiply Data) и характеризуется следующими основными возможностями.

В архитектуре реализован способ распараллеливания, согласно которого: информативные признаки обрабатываются последовательно, классы – параллельно, либо параллельно-последовательно.

Процессорный элемент (ПЭ) является базовым вычислительным модулем, аппаратно реализующим одну Z-модель идентификации на n -входов. Процессорные элементы расположены в линейку с обеспечением возможности параллельно-последовательного режима работы;

Наращивание числа обрабатываемых признаков и числа классов осуществляется за счет увеличения емкости памяти коэффициентов. Наращивание производительности процессора осуществляется за счет каскадного включения дополнительных ПЭ. Архитектура с группой из N_k ПЭ рассчитана на параллельное решение задач классификации если $k \leq N_k$, и параллельно-последовательно (за m циклов), если $k > N_k$:

$$m = \lceil k / N_k \rceil$$

Память коэффициентов содержит все настройки Y , которые заносятся перед началом работы. Память разделена на фреймы по числу ПЭ с емкостью $n \cdot m$ каждый. В ходе вычислений коэффициенты последовательно извлекаются и участвуют в обработке соответствующего информативного признака.

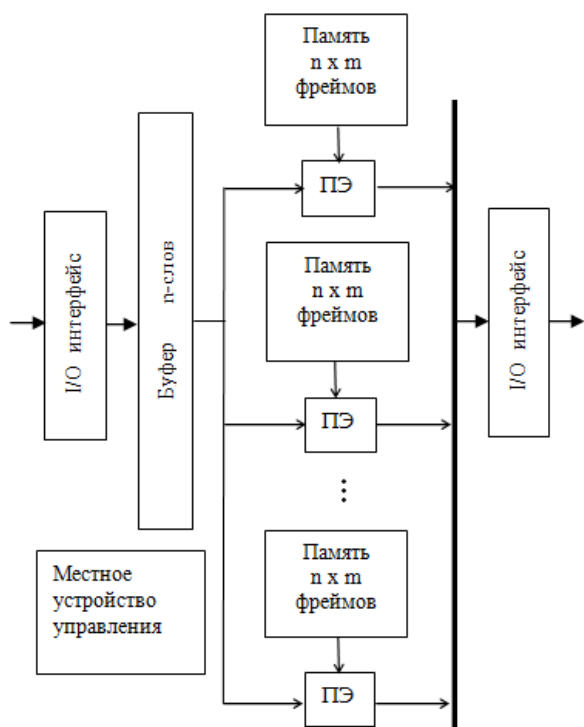


Рисунок 3 – Архитектура параллельного Z-процессора

Автоматическая перестройка временных диаграмм управления процессором в зависимости от параметров n и k осуществляется местным управляющим автоматом.

3. ЭКСПЕРИМЕНТАЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА НА БАЗЕ Z-ПРОЦЕССОРА

Современные FPGA, содержащие встроенные аппаратные умножители и блочную память, представляют собой эффективное средство для быстрого создания аппаратных прототипов. Так, разработанная архитектура была реализована на базе FPGA Xilinx Spartan 6 со следующими выходными характеристиками: 8 ПЭ с объемом фрейма памяти - 2 кБ и временем вычисления по одному признаку - 50 нс. Общее время классификации (T), непосредственно характеризующее производительность процессора, определяется из выражения (3).

$$T \approx (50nk) / N_k \quad (3)$$

Таким образом, процессор представляет собой унифицированное программно-настраиваемое вычислительное устройство, позволяющее решать различные задачи классификации в диапазоне параметров n и k , ограниченном емкостью фрейма памяти, и представленного соотношением $n \cdot k = 100$. При необходимости расширить функциональные возможности процессора достаточно либо использовать более мощную (ресурсоемкую) FPGA, либо разместить проект в нескольких FPGA, при минимальных издержках на архитектурное и структурное перепроектирование.

Очевидно, что для демонстрации функциональных возможностей процессора и проведения натурных испытаний необходимо формировать поток входных данных, принимать результаты промежуточных вычислений, управлять критериями принятия решений и выполнять ряд других сервисных функций. С этой целью была создана экспериментальная вычислительная система, в рамках которой функции ввода-вывода данных, обучения и конечного принятия решений возлагаются на универсальный компьютер, а массовые функции идентификации – на проблемно-ориентированный сопроцессор. На рис.4 представлена обобщенная схема вычислительного комплекса. Host-компьютер и сопроцессор обмениваются командами и данными по стандартным интерфейсам (USB). Такое архитектурное решение позволяет сконцентрировать усилия на исследовании программно-аппаратной реализации классификаторов, при этом реализована возможность удобно обеспечивать общее управление, пользовательский интерфейс и при необходимости предобработку и постобработку обычными программными средствами. Таким образом, вычислительный комплекс позволяет создавать и исследовать широкий класс систем распознавания. Классификатор, разработанный и апробированный в рамках данных инструментальных средств может быть быстро и с минимальными затратами адаптирован к конкретной прикладной системе, к конкретным

конструктивным особенностям (в виде автономного блока или сопроцессора).

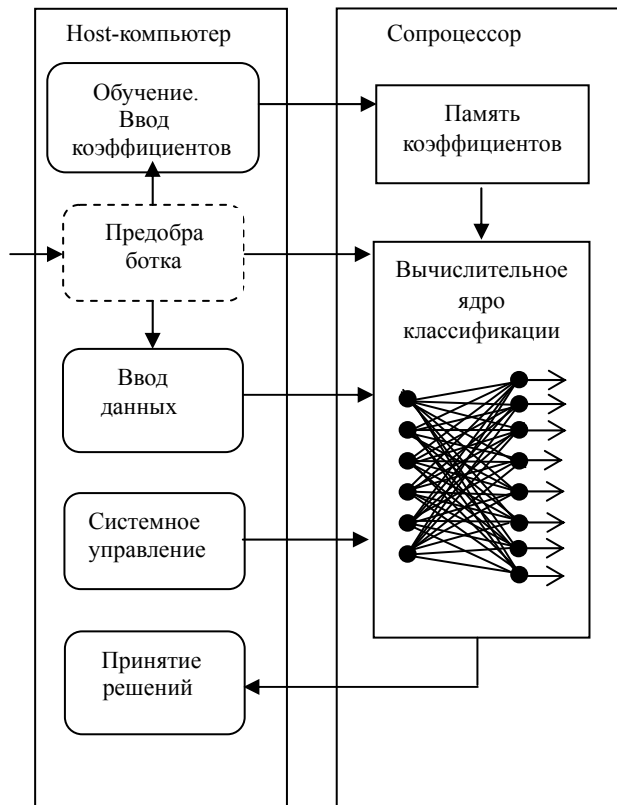


Рисунок 4 – Блок схема вычислительного комплекса

Заключение

В настоящей работе изложены результаты научного исследования, практическим итогом которого явилось создание прототипа параллельного процессора, ориентированного на решение задач классификации. Данный процессор обладает рядом отличительных и в то же время полезных свойств.

1. Процессорный элемент рассчитан на реализацию строго определенной модели классификации. По сути, процессорный элемент – это спецпроцессор, что позволяет обеспечить максимальное быстродействие и упростить его структуру. При этом выбранная математическая модель обеспечивает необходимую унификацию на алгоритмическом уровне и выгодно отличает представленный процессор от нейрокомпьютеров и устройств логического/нечеткого вывода.

2. Архитектура процессора позволяет с минимальными настройками, без программирования, решать задачи классификации в широком диапазоне количественных параметров: входов (информативных признаков) и выходов (классов).

3. Архитектура процессора позволяет легко наращивать производительность за счет каскадного включения дополнительных процессорных элементов.

Следуя классическим путем познания от частного к общему, постараемся обобщить полученный опыт и сформулировать его в виде следующих тезисов по методологии проектирования проблемно-ориентированных процессоров.

Прежде чем приступить к разработке оригинальной аппаратной платформы под какой-либо класс задач следует принять во внимание следующие аргументы:

-существует ли необходимость повышения производительности за счет распараллеливания вычислительного процесса?

существуют ли причины отказа от универсальных параллельных процессоров на базе многоядерных CPU, GPU, DSP в пользу разработки оригинальной архитектуры?

Разработка оригинальной архитектуры процессора потребует значительных временных, трудовых, а следовательно и материальных ресурсов. С целью обеспечения рентабельности проекта в целом необходимо стремиться обеспечить унификацию процессора в «некоторых» рамках. Формально установить эти рамки не представляется возможным, и поэтому жизнеспособность проекта будет зависеть от того, насколько удачно эти рамки будут определены и насколько удачно архитектура будет унифицирована в этих рамках.

Универсальность, как свойство процессора может и должна быть заложена на самом верхнем – идеологическом уровне, на среднем – алгоритмическом и на нижнем – архитектурно-аппаратном уровне. При этом, достигаемые технические характеристики и издержки на обеспечение универсальности оригинальной архитектуры должны быть конкурентными по сравнению со спецпроцессорами и серийными параллельными процессорами (многоядерными CPU, GPU, DSP).

При разработке спецпроцессоров, равно как и универсальных параллельных вычислительных систем присутствует сложившееся разделение труда: математики и алгоритмисты решают задачу на своем – верхнем уровне, разработчики «железа» - на своем. При создании проблемно-ориентированных процессоров с конкурентными технико-экономическими показателями разработчикам необходимо глубоко и скрупулезно владеть предметом как в области математических методов решения задач, так и в области аппаратного проектирования. И в этом состоит одна из ключевых проблем.

По мнению авторов сделанные выводы будут актуальны при создании новых оригинальных архитектур проблемно-ориентированных процессоров, таких как семантические процессоры, ассоциативные процессоры [Голенков и др., 2001], видеопроцессоры, самоорганизующиеся клеточные процессоры, процессоры цифровой обработки сигналов [Philipoff et al., 2010] и др., применимые для построения интеллектуальных систем.

Библиографический список

- [Кохонен, 2008] Самоорганизующиеся карты / Т. Кохонен; под ред. Ю.В.Тюменцева// М.: БИНОМ, . – 655 с.
- [Аляутдинов и др., 2008] Нейрокомпьютеры: от программной к аппаратной реализации/ Аляутдинов М.А. [и др.]; – М.: Горячая линия – Телеком, 2008. – 152 с.
- [Том и др., 2008] Методы интеллектуального анализа многомерных данных для решения задач классификации/ Том И.Э. [и др.]; – Минск, ОИПИ НАН Беларуси, 2011. – 233 с.
- [Кирсанов, 2004] Нейрокомпьютеры с параллельной архитектурой / Э.Ю.Кирсанов. // М.: Радиотехника, 2004. – 221 с.
- [Tatur et al., 2010] Synthesis and Analysis of Classifiers Based on Generalized Model of Identification/ M.Tatur, D. Adzinets, M.Lukashevich, S.Bairak // Advances in intelligent and soft computing. – 2010. - Vol. 71. – P.529-536.
- [Татур и др., 2010а] Классификаторы на основе Z-модели идентификации/ М.М.Татур, Д.Н.Одинец, В.В.Островский, Д.А.Лавникович // Доклады БГУИР. – 2010. -№ 5 (51). – С. 76-81.
- [Татур и др., 2010б] О систематизации методов классификации данных и знаний/ М.М.Татур, Д.Н.Одинец // Информатика. – 2010. - № 3(27). – С.103-113.
- [Татур и др., 2010с] Классификаторы в системах распознавания: прикладные аспекты синтеза и анализа/ М.М.Татур, Д.Н. Одинец. - Минск, Бестпринт, 2008, - 164 с.
- [Голенков и др., 2001] Программирование в ассоциативных машинах/ В.В.Голенков, Г.С.Осипов, Н.А.Гулякина и др. – Мн.: БГУИР, 2001. – 276 с.
- [Philipoff et al., 2010] Spectral Analysis of Real Symmetric Function//P. Philipoff, A.Ivanov, M. Mucoz, G.Raikova, M.Tatur , P.Michaylov// Proc. Of 10-th International Conference VSU'2011, Sofia.

PARALLEL PROCESSORS FOR INTELLIGENT SYSTEMS DEVELOPMENT

Bairak S^{*}, Adzinets D^{*}, Tatur M.^{*},
Philipoff P.^{**}, Munoz M.^{***}

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

tatur@i-proc.com
dima_odin@i-proc.com
info@i-proc.com

*** National Institute for Geophysics, Geodesy,
Geography-Bulgarian Academy of Sciences, Sofia,
Bulgaria*

philip.philipoff@gmail.com

**** Universidad Carlos III de Madrid, Madrid,
España*

munozm@it.uc3m.es

Introduction

Classifier makes decision on what class the pattern belong to and it is intelligent core of any recognition systems. Usually the computing diagram of classification process is shown as graph on fig.1, where n - is number of processed features x_i , and k – is number of classes Z_j . There is the problem to provide the real time processing in intelligent IT-systems with sequence computer architecture for that cases when n and k increase significantly and needed processing time was restricted.

Main part

Authors have proposed the original identification model (1,2), which combines the properties of linear discriminant function with aspects of fuzzy conclusion. It provides the possibilities for a priory data recording in model. Graphical diagram of proposed identification model is shown in fig.2. So, this model possess the number essential properties (including nonlinear discrimination) and has regular computing graph that to put it into basis of parallel processor's architecture unification.

The original architecture belongs to SIMD-kind (Single Instruction and Multiply Data) was developed (fig.3) and it is characterized with the following properties. The increasing of processed features number is provided by means of growing up of the internal memory volume for coefficients solving. Increasing of computer performance is executed by cascade connecting of additional processor elements.

Proposed architecture has been realized with FPGA Xilinx Spartan 6 basis with following output parameters: 8 processor elements with 2 KB memory for each frame; process time for one feature is 50 ns. The full time of classification (T) which determines processor's performance is calculated by (3). To demonstrate the processor's functional possibilities the experimental computer system has been created (fig.4). The control function, learning, preprocessing and post processing are realized by universal computer and mass identification functions – by developed coprocessor.

Conclusion

In given paper the results of scientific researches with practical output on creating of parallel processor prototype oriented to solving of classification problem are presented. The development experience has been generalized as methodology of design for problem oriented processors.

СЕКЦИЯ № 2

СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ, ПРОГРАММ И ПАКЕТОВ ПРОГРАММ, ОРИЕНТИРОВАННЫХ НА ОБРАБОТКУ ЗНАНИЙ

SECTION № 2

SEMANTIC TECHNOLOGIES OF DESIGNING OF KNOWLEDGE BASES, PROGRAMS AND THE SOFTWARE PACKAGES ORIENTED ON KNOWLEDGE PROCESSING

Секция 2 посвящена семантическим моделям баз знаний и семантическим технологиям их разработки. При этом особое внимание уделено семантическим моделям и технологиям для такого важного вида знаний, как программы, представляющие собой знания о способах решения различных задач.

В основе проектирования баз знаний интеллектуальных систем лежит семантическая типология знаний, входящих в состав проектируемой базы знаний, и соответствующее семейство семантически совместимых языков, каждый из которых ориентируется на представление определенного вида знаний. Результатом интеграции таких специализированных языков является общий **язык представления знаний**, который, очевидно, должен быть открытым, поскольку в любой момент времени может возникнуть необходимость в использовании ранее неизвестного вида знаний и соответствующего ему языка.

Интеллектуальная система, в отличие от традиционной компьютерной системы должна **знать** (осознавать, понимать) не только то, **как** она обрабатывает информацию (такие знания о способах обработки информации есть ни что иное, как хранимые программы), но и то, **что** она обрабатывает (т.е. смысл самой обрабатываемой информации), а также то, **зачем** (с какой целью) она осуществляет эту обработку. Образно говоря, интеллектуальная система должна "ведать", и то **как** она творит, и то, **что** она творит, и то **зачем** она творит.

Таким образом, рассмотренный признак семантической типологии знаний, определяемый характером их использования, позволяет выделить такие виды знаний, как **программы** (процедурные, декларативные, параллельные, последовательные, информационные, поведенческие) и **цели**, к числу которых относятся вопросы (информационные цели) и поведенческие цели. Но кроме программ и целей,

интеллектуальные системы оперируют и такими видами знаний как:

- фактографические знания (факты);
- высказывания, описывающие различные свойства и закономерности;
- формальные теории, соответствующие различным логическим исчислениям;
- онтологии;
- терминологические знания;
- описания действий и субъектно-объектных связей;
- описания ситуаций, событий и связей между ними;
- знания, описывающие текущее состояние хранимой базы знаний, а также события, направленные на изменение этого состояния;
- и многие другие виды знаний.

ВОПРОСЫ, ВЫНОСИМЫЕ НА ОБСУЖДЕНИЕ КРУГЛОГО СТОЛА СЕКЦИИ № 2:

- *Чем принципиально отличается семантическое представление знаний от несемантического*
- *О какой семантической типологии знаний можно говорить*
- *Что такое качество знаний (степень уверенности, не-факторы)*
- *В чем заключаются проблемы интеграции баз знаний.*
- *Как обеспечить семантическую совместимость баз знаний.*
- *Проблема согласования онтологий баз знаний.*
- *Что может быть ip-компонентами баз знаний.*
- *Зачем нужна структуризация баз знаний и каковы виды такой структуризации.*
- *Что такое метаинформация и ее типология.*
- *Требуется ли интеллектуальной системе использование нескольких языков программирования.*
- *В чем специфика языков программирования, ориентированных на обработку баз знаний.*
- *В чем суть семантической совместимости языков программирования и программ, написанных на одном или на разных языках программирования.*
- *Семантические модели программ и языков программирования.*
- *Семантическая типология языков программирования.*
- *В чем специфика языков программирования, ориентированных на обработку семантических сетей.*
- *Проблемы создания параллельных языков программирования различного уровня, ориентированных на обработку семантических сетей в реконфигурируемой памяти на основе ассоциативного доступа по произвольному образцу.*
- *Как могут быть использованы семантические сети для развития CASE-технологий.*



УДК 004.822:514

СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ: ОЖИДАНИЯ И ТРЕНДЫ

Хорошевский В.Ф.

*Федеральное государственное бюджетное учреждение науки,
Вычислительный центр им. А.А. Дородницына РАН
г. Москва, Россия*

khor@ccas.ru

В статье обсуждается состояние исследований и разработок в области нового направления «Семантические технологии». Основное внимание уделяется анализу новых технологических трендов в данной области и экспертной оценке их значимости и сроков появления реальных прикладных продуктов на основе использования методов и средств семантической обработки информации.

Ключевые слова: научно-технический тренд, прикладная интеллектуальная система, семантические технологии, технологический триггер.

ВВЕДЕНИЕ

Исследования мировых лидеров в прогнозных исследованиях [IDC, 2012; Tofler, 2006; GARTNER, 2012] показывают, что «...до 2020 года количество информации и потребности в ней будут расти экспоненциально... Без умения создавать и обрабатывать такие объемы информации ЛПР будущего будут введены в состояние, которое можно назвать "аналитический паралич"...». Таким образом, одной из самых больших проблем современного общества является информационное переполнение. И в настоящее время уже осознано направление главного «удара» в борьбе с информационным взрывом – переход от хранения и обработки данных к накоплению и обработке знаний, что, в свою очередь, формирует новую семантическую волну (Рис. 1), которая, по оценке руководителя проекта Project 10X Миллса Дэвиса [Mills, 2006], существенным образом изменит характер работы с информацией.

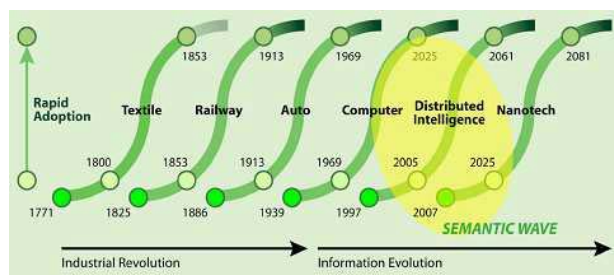


Рисунок 1 – Семантическая волна

Понятно, что в рамках такой семантической волны формируются и развиваются семантические

информационные технологии представления и обработки знаний, анализ зарождения и развития которых представляет значительный интерес.

Учитывая вышесказанное, целью настоящей статьи является обсуждение основных трендов в области семантических технологий и экспертная оценка тех ожиданий, которые связываются с их внедрением в практику. Организовано изложение следующим образом: сначала кратко представлены базисные семантические технологии, без которых, по нашему мнению, невозможно появление практически значимых семантических приложений, а затем, более подробно, обсуждаются некоторые из основных трендов в области семантических технологий.

1. Базовые тренды в тематической области «Семантические технологии»

Как представляется, базовые тренды в области семантических технологий в значительной мере связаны с концепцией Semantic Web (SW), которую в 2000 г. выдвинул Тим Бернерс-Ли (Tim Berners-Lee) – один из основоположников WWW и нынешний председатель консорциума W3C [Berners-Lee, 2003].

С момента появления этой концепции прошло уже более 10 лет, но пока SW-эра, в отличие от эпохи Интернет, еще только приближается и на этом пути существует значительное число научных, технических, технологических и человеческих проблем, основными из которых являются [Benjamins et al., 2002] доступность семантического контента, доступность онтологий и средств их

разработки, а также эволюция онтологий, масштабируемость, мультязыковость, визуализация и стабильность.

Доступность семантического контента является основной проблемой на пути формирования и использования пространств знаний, так как сейчас основная масса информации не представлена в «семантических» форматах и нет надежды, что эта работа может быть выполнена вручную. Онтологии, по мнению практически всех специалистов, являются ключевым компонентом в решении проблемы семантизации контента. В связи с этим особое значение приобретают проблемы онтологического инжиниринга, а также доступность уже существующих онтологий. Значительные усилия должны быть предприняты для хранения, обработки и поиска семантического контента, причем решения в этой области должны обеспечивать эффективную работу с огромными объемами знаний. Проблема мультязыковости контента существует давно, но для пространств знаний, где, по сути, должен поддерживаться эффективный доступ к информации независимо от того, на каком языке она представлена изначально, данная проблема является одной из основных. Представление информации для пользователей (визуализация контента) также должно претерпеть существенные изменения и обеспечить свободную ориентацию в огромном количестве фактов, которые отвечают его потребностям. Последняя по счету, но не по важности, проблема связана с обеспечением стабильности пространств знаний, а это, в свою очередь, предполагает, что существенные усилия должны быть предприняты в области стандартизации, обеспечивающей создание технологий, необходимых для формирования и сопровождения пространств знаний.

Понятно, что подробное рассмотрение и обсуждение даже только перечисленных выше базовых проблем и трендов семантических технологий в рамках одной статьи является практически безнадежным делом. Поэтому ниже мы сосредоточимся на нескольких трендах прикладных семантических технологий, а читателей, интересующихся базисными семантическими технологиями отсылаем к аналитическим обзорам, представленным в работах [Хорошевский, 2008; Хорошевский, 2009; Хорошевский, 2012].

2. Прикладные тренды семантических технологий

2.1. Предварительные замечания

Как известно, «тренд» (от англ. Trend) это долговременная общая тенденция изменения исследуемого временного ряда. В данном разделе обсуждаются технологические тренды – активно развивающиеся в последние 5 лет технологические направления в области семантических технологий, которые, как ожидается, продолжат свое развитие в ближайшие 10 лет. Обсуждение основано на

результатах анализа новых технологических трендов в области семантических технологий в рамках Программы фундаментальных исследований НИУ ВШЭ в 2011 году, в которых принимал участие автор настоящей работы. В процессе этих исследований был проведен опрос экспертов, в результате чего для более детального анализа был сформирован перечень основных технологических трендов в области «Семантические технологии», куда вошли:

- Открытые связанные данные (Linked Open Data);
- Социальный семантический Вэб (Social Semantic Web);
- Бизнес-анализ на базе семантических технологий (Semantic Business Intelligence);
- Семантические цифровые библиотеки (Semantic Digital Libraries) и
- Семантические технологии в приложениях (Semantic-based e-Apps).

Учитывая ограничения на объем настоящей работы, ниже обсуждаются подробнее три из них – открытые связанные данные, социальный семантический Вэб и семантические цифровые библиотеки.

2.2. Открытые связанные данные

Следует сразу отметить, что термин Linked Open Data является не вполне удачной альтернативой для термина Linked Data, поскольку последний представляет действительно технологию, а первый – название проекта, который задал направление исследований и разработок в области связывания данных в Web-пространстве. Вместе с тем, в настоящее время термин Linked Open Data часто используется как альтернативный и/или даже синонимичный к термину Linked Data.

2.2.1. Описание тренда

Термин Linked Data соответствует одной из центральных идей Семантического Вэба, в основе которой лежит то, что первичными объектами здесь являются описания сущностей с явным указанием их семантики и семантики ссылок (отношений) между ними. Технологически это обеспечивается представлением данных в виде триплетов «субъект-предикат-объект» на языке RDF, идентификацией данных с помощью URI, механизмом доступа по протоколу HTTP, а также спецификацией контролируемых словарей на языках RDFS и OWL. Таким образом, понятие Linked Data относится к установлению наилучших способов публикации и связывания данных в Web-пространстве [Bizer et al., 2009], а характерной чертой связанных данных является их ориентация на Вэб и Семантический Вэб, позволяющая производить разделение данных на глобальном уровне между множественными разнородными источниками.

В представленной выше трактовке Linked Open Data является технологическим трендом появление которого достаточно четко связывается со статьей

Тима Бернерса-Ли [Berners-Lee, 2006], где были сформулированы следующие правила (Linked Data principles) публикации данных на Вэбе:

- Использование URI как имен сущностей.
- Использование HTTP URI, чтобы люди могли «видеть» и использовать эти имена.
- Использование стандартов RDF и SPARQL для получения полезной информации через посредство URI.
- Включение в публикации связей с другими URI и, таким образом, обеспечение возможностей отражения новых связанных сущностей.

Сформулированные Тимом Бернерсом-Ли принципы технологии Linked Data, по существу, определили подходы и методы их реализации, которые в значительной мере использовали результаты предшествующих исследований и разработок в области объединения и совместного использования распределенных баз данных, а также в области представления и обработки знаний в среде Интернет.

Вместе с тем, становление и развитие LD-технологии поставило перед исследователями и разработчиками, работающими в данной области, новые научно-технические проблемы. К основным из них, которые не решены полностью до настоящего времени, относятся:

- Проблема функционально полных и то же время эффективных методов и средств спецификации моделей пространств знаний (онтологическое моделирование предметных областей).
- Проблема объединения и выравнивания (Merging & Alignment) онтологических моделей, разработанных разными коллективами и используемых в разных приложениях, а также валидация и верификация онтологических моделей.
- Проблема оценивания связываемых данных и формализации методов определения доверия к используемым данным.

Отметим, что решение первой из указанных проблем в настоящее время связывается, в первую очередь, с использованием уже разработанных стандартов W3C, среди которых основную роль играют языки спецификации онтологий семейства OWL [OWL, 2009] и язык запросов к RDF-хранилищам SPARQL [SPARQL, 2008].

Одновременно с научно-техническими проблемами развитие LD-технологии потребовало решения ряда технологических задач, среди которых, в первую очередь необходимо выделить

- эффективную реализацию RDF-хранилищ,
- масштабирование связанных данных и
- эффективность доступа к связанным данным в условиях экспоненциально увеличивающихся объемов числа запросов на данные.

Решение этих проблем в настоящее время идет по нескольким направлениям. С одной стороны, разрабатываются новые методы и средства

эффективной реализации RDF-хранилищ и эффективного хранения данных в RDF-хранилищах [Лапшин, 2010], а с другой – предпринимаются усилия по реализации RDF-хранилищ в среде облачных вычислений.

В целом же можно констатировать, что в настоящее время технология связанных данных активно развивается в проектах открытых связанных данных (Linked Open Data), формируя тем самым новый тренд – «Открытые связанные данные».

При этом интересно отметить, что научно-технические проблемы LD-технологии практически непонятны ни бизнес-сообществу, участвующему в финансировании работ в данной области, ни потребителям полученных здесь решений. И более того, потребители результатов LD-технологии и LOD-проектов могут даже не подозревать, что в основе конечных продуктов и сервисов, которыми они пользуются, лежат инновационные решения из области семантических технологий. Указанная ситуация связана с тем, что LD-технология является базовой и обеспечивает разработку новых информационных пространств, но не имеет конечных потребительских свойств, очевидных для пользователей.

2.2.2. Проекты, продукты и сервисы тренда

С учетом вышесказанного, в основе данного технологического тренда в настоящее время лежат не столько продукты, сколько проекты в области открытых связанных данных (LOD-проекты) и те сервисы, которые постепенно формируются на их основе.

Как известно, первым проектом нового поколения в области связанных данных был проект LOD [LOD, 2012], выполняемый в рамках SWEO (Semantic Web Education and Outreach) Interest Group, миссия которой в расширении взаимодействия в Вэб-сообществе и в обучении необходимым для этого решениям и технологиям. Целью общего движения за открытые данные было и остается желание сделать данные свободно доступными для всех заинтересованных организаций и частных лиц. При этом уже до начала собственно проекта W3C SWEO существовали разные интересные наборы открытых данных, доступные на Вэбе, среди которых наиболее известными были Wikipedia, Wikibooks, Geonames, MusicBrainz, WordNet, the DBLP bibliography и другие, опубликованные под лицензиями Creative Commons или Talis.

Учитывая вышесказанное, целью проекта LOD было расширение Вэба различными уже опубликованными открытыми данными за счет использования RDF-представлений и установления RDF-связей между «атомами» данных из различных источников данных, что должно обеспечить навигацию от атомарных данных одних источников данных к соответствующим атомарным данным

других источников с использованием браузеров Semantic Web и, кроме того, индексацию RDF-связей краулерами семантических поисковиков. Такой подход, по мнению авторов проекта, должен обеспечить интеллектуальный поиск и возможности запросов на данные, результатами которых были бы не просто ссылки на HTML-страницы, а структурированные данные, которые бы могли использоваться другими приложениями.

В настоящее время (сентябрь 2011 г.) в рамках данного проекта уже опубликовано 295 наборов данных, содержащих более 31 блн. RDF-триплов, которые «перевязаны» около 504 млн. RDF-связей.

При этом аргументы «ЗА» и «ПРОТИВ» открытых данных хорошо известны и сводятся, в частности, к следующему:

- Данные принадлежат всему человечеству.
- Для выполнения работ использованы деньги общества и потому данные должны быть в свободном доступе.
- Данные создавались правительственными организациями (обычно это US National Laboratories и другие государственные агентства).
- Факты не могут быть копирайтными легально.
- Спонсоры исследований не получают полной «отдачи» пока результирующие данные находятся в свободном доступе.
- и т.д.

С учетом вышесказанного, новые технологии открытых связанных данных могут стать препятствием для их внедрения в повседневную практику использования, хотя полезность и целесообразность их несомненны.

Кроме рассмотренного выше проекта LOD наиболее известными в данной области в настоящее время являются:

- Проект европейского сообщества Linked open data around the clock (LATIC) [LATIC, 2012], который выполняется в рамках 7-й Рамочной Программы и направлен на поддержку публикации и использования связанных открытых данных, на совершенствование инфраструктуры для мониторинга использования и повышение качества связанных открытых данных; снижение барьеров доступа к данным для издателей и потребителей; разработку библиотеки инструментов обработки открытых источников данных; эксперименты по обработке связанных данных в комбинации с данными Европейского Союза и поддержку сообщества учебной литературой и лучшими примерами (best practices).
- Проект европейского сообщества PlanetData [PlanetData, 2012], целью которого является формирование сети достижений, где были бы объединены европейские исследователи в области управления данными большой размерности, в которой бы были представлены

RDF-данные, опубликованные в соответствии с принципами Linked Data principles. Проект Planet Data уникален тем, что открыт для новых партнеров в процессе его выполнения.

- Европейский проект Linking Open Data 2 [LOD2, 2012] – часть 7-й Рамочной Программы Европейского Сообщества с бюджетом в 6.5 млн. Евро (2010-2014 г.г.), который продолжает работы по проекту LOD и ставит своей целью создание пространств знаний через взаимосвязанные данные путем разработки промышленных инструментов и методологий для экспозиции и управления очень большими объемами структурированной информации из Вэба данных (Data Web); испытательного стенда и расширяемой сети высококачественных мультипредметных и многоязыковых онтологий из таких источников, как Wikipedia и OpenStreetMap; алгоритмов на основе машинного обучения для автоматического связывания и объединения данных, представленных на Вэбе; стандартов и методов для надежного отслеживания «происхождения», обеспечения приватности и безопасности данных, а также достижения высокого качества информации; адаптивных средств поиска, навигации и авторства связанных данных.

Приведенные аннотации LOD-проектов еще раз подтверждают представленное выше утверждение о том, что внедрение технологии L(O)D только предстоит, а пока идет работа по ее реализации, поскольку, по определению, технология это совокупность методов и средств, направленных на достижение определенных целей.

Из продуктов и сервисов в области охвата технологического тренда «Открытые связанные данные» в настоящее время можно отметить браузеры и навигаторы связанных данных, визуализаторы связанных данных и редакторы запросов.

2.2.3. Новые потребительские свойства и ожидаемые эффекты тренда

Новые потребительские свойства, присущие технологическому обсуждаемому тренду определяются тем, что результаты развития L(O)D-технологии несут в себе новые возможности в формировании пространств взаимосвязанных данных в сети Интернет, приближая тем самым переход от Вэба данных к Вэбу знаний. Такой переход будет характеризоваться принципиально новыми возможностями поиска и навигации по данным в сети Интернет, с одной стороны, и переходом от информационных систем к информационно-аналитическим системам нового поколения практически во всех предметных областях.

При этом наиболее востребованными, на наш взгляд, зонтичными областями, где использование L(O)D-технологий может дать максимальный

эффект являются

- информационные системы органов государственной власти всех уровней;
- социальные сети;
- информационно-аналитические сети научных и профессиональных сообществ;
- системы дистанционного обучения и
- системы экспертных консультаций.

Среди важнейших ожидаемых эффектов технологического тренда «Открытые связанные данные», в первую очередь, целесообразно выделить: *социальный* (существенное увеличение открытости данных и информационной прозрачности органов власти, вовлечение новых слоев гражданского общества в использование открытых данных и расширение социальных сетей); *экономический* (сокращение сроков поиска нужной информации, облегчение поиска бизнес-партнеров и необходимых для производства материальных, производственных и людских ресурсов, сокращение сроков и повышение качества маркетинга); *политический* (повышение доверия граждан к власти и демократизация общества за счет вовлечения в процессы принятия решений значительных слоев населения).

возможные области применения тренда

Как показывает анализ, основные работы в рамках технологического тренда «Открытые связанные данные» ведутся на стадиях фундаментальных и прикладных исследований, результаты которых начинают использоваться при создании опытных образцов.

Учитывая вышесказанное, вероятный срок появления на рынке важнейших продуктов и услуг в рамках технологического тренда «Открытые связанные данные» определяется сроками завершения основных фундаментальных и прикладных исследований после чего должно пройти еще 1-3 года. Таким образом, если считать, что основные фундаментальные и прикладные исследования дадут необходимые для практики результаты высокого качества в течение 5-ти ближайших лет, то реального появления на рынке важнейших продуктов/услуг в рамках обсуждаемого технологического тренда можно ожидать не ранее 2017-2020 г.г.

Следует также отметить, что прогнозы сообщества R&D значительно отличаются от прогнозов пользователей и даже от прогнозов бизнес-сообщества (Рис. 2).

2.2.4. Оценка сроков внедрения приложений и

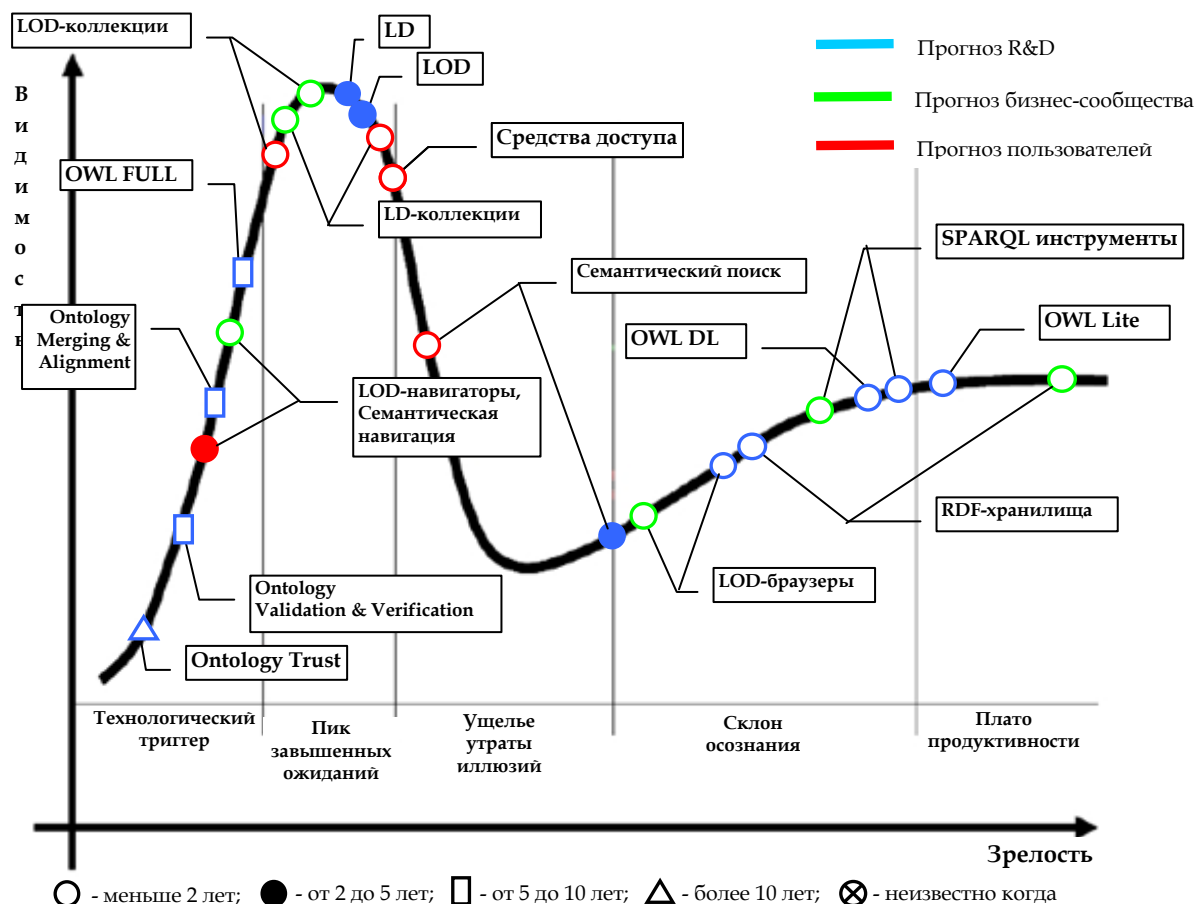


Рисунок 2 – Прогнозная кривая для технологического тренда «Открытые Связанные Данные»

Как отмечалось выше, технологический тренд «Открытые связанные данные» является базисным для всех областей применения, где уже имеются представительные множества коллекций востребованных данных. По нашему мнению, в первую очередь, это

- области фундаментальных и прикладных исследований, для которых необходимо формирование и активное использование общих пространств знаний;
- государственное управление всех уровней;
- новые государственные и коммерческие информационные системы.

2.2.1. Технологии-конкуренты и технологии, от которых зависит развитие тренда

Как представляется, в качестве основного конкурирующего технологического направления по отношению к тренду «Открытые связанные данные» можно указать тренд «Интеграция распределенных баз данных», основными преимуществами которого можно считать существенно более длительное время развития данного направления; наличие центров компетенции по данному технологическому направлению во всем мире; многолетнюю работу национальных и международных комитетов и организаций по стандартизации способов представления и обработки информации в базах данных; сформировавшиеся рынки производителей программного обеспечения СУБД и поставщиков контента, а также опыт потребителей в использовании продуктов этого технологического направления во всем мире.

Среди недостатков технологии-конкурента можно отметить неспособность в рамках базисной парадигмы легкой интеграции в среду Интернет; медленную адаптацию основных вендоров СУБД к потребностям потребителей в эффективных и легко масштабируемых RDF-хранилищах, а также отсутствие общезначимых и признанных результатов в области формирования и использования пространств знаний.

Развитие технологического тренда «Открытые связанные данные», по нашему мнению, зависит от таких технологий из смежных областей, как представление и обработка знаний; онтологический инжиниринг; семантический поиск информации и аналитика на знаниях.

2.2.5. Разрушающий потенциал, барьеры и риски внедрения тренда

Разрушающий потенциал технологии связанных данных состоит, прежде всего, в том, что продукты и сервисы этой технологии, будучи введенными в массовое использование, существенно изменят рынок поставки информации потребителям. Изменится и технологический ландшафт информационных технологий, в котором, по нашему мнению, ведущее положение займут поставщики

семантизированного контента. Нарушится существующий баланс сил в отраслевом сегменте за счет того, что наиболее успешными будут те технологические компании, которые первыми выйдут на рынок программных продуктов, поддерживающих создание и обработку больших и сверхбольших пространств открытых связанных данных.

При этом барьеры и риски внедрения тренда, по нашему мнению, определяются следующими факторами:

- Завышенные ожидания потребителей продуктов технологического тренда «Открытые связанные данные» относительно реальных возможностей производителей программных средств для эффективной поддержки процессов формирования и обработки сверхбольших объемов связанных данных.
- Слабость законодательной базы, регламентирующей условия и способы свободного использования открытых связанных данных.
- Относительно небольшие объемы коллекций открытых связанных данных, доступных потребителям.

2.2.6. Заключительные замечания по тренду

В настоящем разделе рассмотрены некоторые из основных аспектов технологического тренда «Открытые связанные данные», дано описание тренда, кратко представлены проекты, продукты и сервисы, а также новые потребительские свойства и ожидаемые эффекты тренда и дана оценка сроков внедрения приложений и возможные области его применения. Идентифицированы технологии-конкуренты и технологии, от которых зависит развитие тренда, фиксирован разрушающий потенциал, барьеры и риски внедрения тренда.

2.3. Социальный семантический Вэб

Одной из самых известных диаграмм развития Интернет является диаграмма Миллза Дэвиса, представленная на Рис. 3.

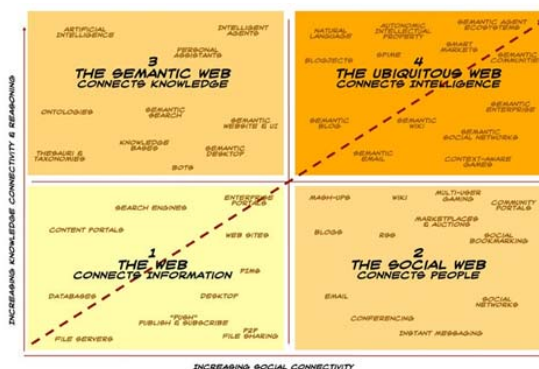


Рисунок 3 – Эволюция Интернет до 2020 года

Как следует из диаграммы, технологический тренд «Социальный Вэб» является одной из 4-х важнейших составляющих эволюции Интернет,

который сформировался в начале XXI века и устойчиво развивается уже более 5 лет. При этом основные социальные сети демонстрируют впечатляющие характеристики: 800 млн. мин. сообщений на площадке Facebook; 90 млн. «переговоров»/день на площадке Twitter; 24 часа видео/мин на площадке YouTube; более 200 языков в среде Wikipedia и т.д., и т.п., что показывает высокую востребованность социальных сетей.

Вместе с тем, в последние несколько лет появился новый технологический тренд «Социальный Семантический Вэб», обсуждению которого посвящен настоящий подраздел.

2.3.1. Описание тренда

Основными составляющими технологического тренда «Социальный Семантический Вэб» являются интеграция уже активно используемых в таких продуктах социальных сетей, как блоги, форумы, энциклопедии и т.п., средств коммуникации и совместной работы, со средствами и технологиями Семантического Вэба. При этом используются как общие средства уровня RDF(S), OWL, SPARQL, так и более простые модели внесения семантики, например, микроформаты.

По сути дела, этот тренд опирается на концепцию Open Distributed Semantic (Открытость-Распределенность-Семантичность), в рамках которой составляющая Open предполагает возможности разделения персональных данных в любой момент и в любой точке по желанию пользователя, составляющая Distributed – на пресуппозицию хранения персональных данных в персональном месте, а составляющая Semantic предполагает обогащение персональных данных таким образом, чтобы наполнить их смыслом.

Основными технологиями, обеспечивающими данную концепцию, являются Семантический Вэб (Semantic Web) и Связанные Открытые Данные (Linked Open Data), которые интегрируются в рамках технологии Социальных Сетей.

В новом технологическом тренде «Социальный Семантический Вэб» эти технологии используются на уровне формализмов RDF(S) и RDFa, SPARQL и описания онтологических моделей с помощью языков семейства OWL, а также применения средств семантического тэгирования.

В рамках нового тренда объединение людей происходит не на уровне общей площадки для коммуникации, а на уровне объектно-центрированной социализации (Object-centred sociality). Согласно данной концепции люди не просто объединяются друг с другом в социальной сети, но объединяются через общие разделяемые объекты. Поэтому полезные социальные продукты и сервисы должны обеспечивать создание социальных объектов, объединяющих людей. При этом формируются социальные сети по интересам – работа, учебные заведения, хобби и т.п. Так, например, площадка Flickr объединяет ее

посетителей по объекту «фото» (photos), площадка del.icio.us ориентирована на обмен «закладками» (bookmarks) и т.д. Вместе с тем, в настоящее время все большее внимание привлекают площадки, на которых возможна интеграция интересов, что обеспечивает развитие нового тренда «Социальный Семантический Вэб».

Таким образом, обсуждаемый технологический тренд не является результатом собственных фундаментальных и прикладных исследований, а интегрирует результаты других технологических трендов – «Семантический Вэб», «Связанные Открытые Данные» и «Социальный Вэб». При этом новый технологический тренд активно формирует собственные продукты и сервисы, кратко представленные в следующем подразделе.

2.3.2. Продукты, сервисы и проекты тренда

Продукты и услуги социального Вэба (Web 2.0) хорошо известны – это, в первую очередь, площадки Facebook, Flickr, LinkedIn, Twitter, Slideshare, Wikipedia и многие др., активно используемые во всем мире, а также отечественные социальные сети «ВКонтакте», «Мой мир», «Одноклассники», «Ютуб», «Википедия», «ЛайвДжорнал» и др.

Менее известны продукты и услуги технологического тренда «Социальный Семантический Вэб», поскольку здесь наблюдается 2 «потока»: вовлечение семантических технологий в уже известные продукты давно работающих на данном рынке производителей и в новые проекты, а также продукты/услуги новых компаний, которые появились совсем недавно.

Среди «ветеранов» социального Вэба, активно использующих семантические технологии, по нашему мнению, лидируют Freebase и Facebook, продукты которых входят в Top 10 Semantic Web Products в 2009-2010 г.г. по версии авторитетного издания ReadWriteWeb [ReadWriteWeb, 2010].

Как известно, Freebase открытая база данных, объекты в которой проиндексированы семантическими тэгами. В этом смысле база Freebase похожа на базу Wikipedia, но в базе Freebase представлена информация о структурированных данных и о том, что с ними можно делать. При этом Freebase является одним из явных адептов Semantic Web и Linked Data. Интересно, что база Freebase недавно была приобретена Google.

Новый «рывок» Facebook в сегменте социальных сетей, на наш взгляд, можно связать с анонсированной новой платформой Open Graph, целью которой является предоставление издателям возможностей интеграции их Вэб-страниц в социальные графы. При этом каждая страница становится объектом социального графа Facebook, через который пользователи могут связываться друг с другом в своих сетях. И более того, это означает, что страницы могут связываться в рамках социальной сети через профили пользователей,

сообщения в блогах, результаты поиска, новостные ленты Facebook и т.п.

Серьезные инструментальные средства для семантических социальных сетей выведены на рынок сервисом Zemanta [Zemanta, 2012]. По существу это инструменты блоггеров (blogging tool), которые базируются на семантических технологиях для включения релевантного контента в сообщения (заметки) пользователей. Основные возможности инструментария Zemanta связаны с легким поиском тематических фотографий к постам, быстрым поиском релевантных постов, автоматическим формированием ссылок на авторитетные сайты и подбором наиболее подходящих тэгов для SEO. Программное обеспечение Zemanta распространяется как Open Source и отвечает стандартам Semantic Web.

Активную позицию в области социальных семантических сервисов демонстрирует в последнее время и Google, который «запустил» новые сервисы Google Squared и Google Search Options and Rich Snippets.

Google Squared поддерживает обработку простых вопросов на естественном языке, дает простые ответы и вводит новую возможность в рамках Google's sidebar, которая называется "Something different", где представляется список результатов поиска, которые могут находиться в области интересов пользователя. Технология использует анализ текущих поисковых терминов пользователя. Сервис Google Search Options and Rich Snippets является расширением «ядерного» поисковика Google за счет обогащения найденных фрагментов структурным контентом. При этом используются открытые стандарты структурированных данных (микроформаты и RDFa).

Серьезной заявкой на социальный семантический Вэб является инициатива SIOC [SIOC, 2012] (Semantically-Interlinked Online Communities), цели которой – следующие:

- Семантически связать существующие online-сообщества.
- Полностью описать контент/структуру социальных Вэб-сайтов.
- Создать новые связи между online-дискуссиями, форумами и контейнерами.
- Обеспечить интеграцию информации online-сообществ.
- Поддерживать инновационный просмотр связанных элементов Social Web.
- Разрешить «chicken-and-egg» проблему с Semantic Web.

Основным подходом к решению поставленных задач инициатива SIOC считает онтологический инжиниринг. Уже в настоящее время SIOC-онтология, опубликованная как W3C Member Submission, поддерживается 16 организациями.

Другими интересными проектами в области

охвата технологического тренда «Социальный Семантический Вэб» являются:

- SCOT (Social Semantic Cloud of Tags) [SCOT, 2012] - проект, целями которого является создание модели для описания tagclouds (тэги и совместное их появление), обеспечение возможностей перемещения собственных tagcloud от одного сервиса к другому, разделение tagclouds между сервисами и между пользователями, а также обеспечение портативности тэгов (Tag portability).
- MOAT (Meaning Of A Tag) [MOAT, 2012] - проект с целями создания модели для описания смысла тэгов, «перевязки» тэгов, управляемой пользователем, интеграции тэгируемого контента с Вэбом связанных данных, а также коллаборативный подход к «разделению» смыслов в сообществе.
- CommonTag [CommonTag, 2012] - проект, в котором объединяются усилия AdaptiveBlue, DERI at NUI Galway, Faviki, Freebase, Yahoo!, Zemanta и Zigtag для разработки формата связи значимых ресурсов необходимыми тэгами.

Общая схема взаимосвязи рассмотренных выше проектов и объектов социального Вэба, что, собственно, и знаменует собой переход к социальному семантическому Вэбу, представлена на Рис. 4.

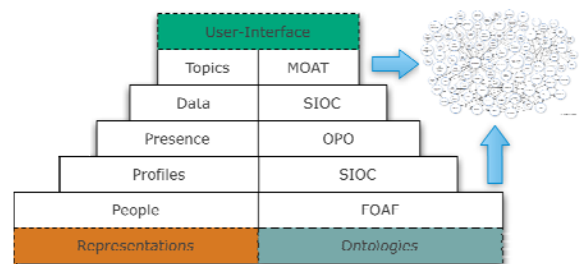


Рисунок 4 – Общая схема взаимосвязи проектов и объектов социального семантического Вэба

2.3.3. Новые потребительские свойства и ожидаемые эффекты тренда

Новые потребительские свойства, присущие технологическому тренду «Социальный Семантический Вэб», определяются, по нашему мнению, в значительной мере результатами развития семантических технологий в целом. При этом основную роль должны сыграть новые методы и средства извлечения информации из ЕЯ-текстов, методы семантического индексирования контента на основе онтологических моделей, методы семантического поиска информации, а также новые методы и средства семантического связывания данных в социальных сетях.

Такой переход будет характеризоваться принципиально новыми возможностями в социальных сетях за счет

- внесения семантики в данные, представленные в социальных сетях в настоящее время;

- явной спецификации смысловых связей между данными различных социальных сетей и за счет
- формирования новых сообществ, в которые будут вовлечены разные люди из уже существующих сообществ, и динамического изменения состава и размера существующих сообществ.

Представляется интересным и такое новое свойство социальных семантических сетей, как появление в них разделов многопользовательских игр, в которых участвуют члены социальной сети, а также зарождение нового технологического тренда Enterprise 2.0, в рамках которого формируются социальные бизнес-сети.

Среди важнейших ожидаемых эффектов нового технологического тренда, в первую очередь, целесообразно

выделить: социальный (существенное увеличение «осмысленности» данных, циркулирующих в сообществах, представленных на той или иной площадке социальных сетей, вовлечение новых слоев общества в активное использование социальных сетей в повседневной жизни, интеграция социальных сетей и, как следствие, с увеличение взаимодействия между разными слоями общества, появление нового класса социальных сетей – сетей экспертных сообществ, предоставляющих независимые консультации в области своей компетенции); экономический (появление социальных бизнес-сетей, где производители товаров и услуг смогут быстро получать реальный отклик на свои новые решения, а потребители получают возможности двусторонней оценки предоставляемых товаров и услуг, а также быстрый доступ к приобретению нужных товаров и услуг); политический (возможность создания социальных сетей, где формируется общественное мнение по поводу политики властей).

2.3.4. Оценка сроков внедрения приложений и возможные области применения тренда

Основные работы в рамках технологического тренда «Социальный Семантический Вэб» ведутся, как это не странно на первый взгляд, на стадиях создания опытных образцов и запуска их в массовое использование. В этом смысле технологический тренд «Социальный Семантический Вэб» имеет гораздо более благоприятную почву для своего развития в виде востребованного обществом социального Вэба и уже существующими продуктами и сервисами для его массового использования, чем другие технологические тренды, обсуждаемые в настоящей работе.

Прогнозная кривая технологического тренда «Социальный Семантический Вэб» представлена на Рис. 5 и показывает, что в данном случае имеется хорошее соответствие между мнением бизнес-сообщества и потребителями (хотя последние, по нашему мнению, дают более оптимистические прогнозы) и понятное расхождение во мнениях между бизнес-сообществом и сообществом R&D,

поскольку первое считает, что переход к семантическому социальному Вэбу будет естественным продолжением работ по Вэбу 2.0, а второе – что на пути к реально функционирующему семантическому социальному Вэбу предстоит решить много наукоемких и технически сложных задач.

2.3.5. Технологии-конкуренты и технологии, от которых зависит развитие тренда

Конкурирующими технологиями по отношению к семантическим социальным сетям, по нашему мнению, являются интерактивные медийные технологии и, в частности, средства коллективной мобильной связи, обеспечивающие целевую доставку контента потребителям, и интерактивное телевидение. При этом надо отметить, что социальные семантические сети очень быстро интегрируют конкурентные продукты в качестве своих сервисов и используют конкурирующие технологии для развития собственных продуктов и сервисов. Хорошим примером интеграции конкурентных продуктов являются социальные игры, приобретающие все большую популярность в социальных сетях.

Преимущества технологий-конкурентов, по-видимому, связаны с их большей «понятностью» для старших поколений населения, а недостатки определяются ограничениями технических средств поддержки этих технологий.

По нашему мнению, развитие технологического тренда «Социальные Семантические Сети» опирается на результаты «родительского» технологического тренда «Социальные Сети» и существенно зависит от таких технологий из смежных областей, как представление и обработка знаний; онтологический инжиниринг; семантизация контента; семантический поиск информации и экспертные системы.

Специфика обсуждаемого технологического тренда, по нашему мнению, состоит в том, что массовое производство продуктов здесь не требуется, а массовое использование существующих продуктов и их новых версий будет происходить одновременно с появлением первых важнейших продуктов и/или услуг. При этом, по нашему мнению, целесообразно эти продукты/услуги рассматривать в 2-х аспектах: продукты и сервисы первого поколения, к которым мы относим модификации существующих платформ путем частичного внедрения в них семантики, и продукты и сервисы второго поколения, где будут в полном объеме использоваться разработанные и реализованные к тому времени решения в области семантических технологий, в первую очередь, методов и средств семантизации контента, Semantic Web, L(O)D-продуктов и программно-аппаратных средств передачи информации в сетях. Для первого поколения наша экспертная оценка 2013 – 2015 г.г., а для второго –: 2014 – 2018 г.г.

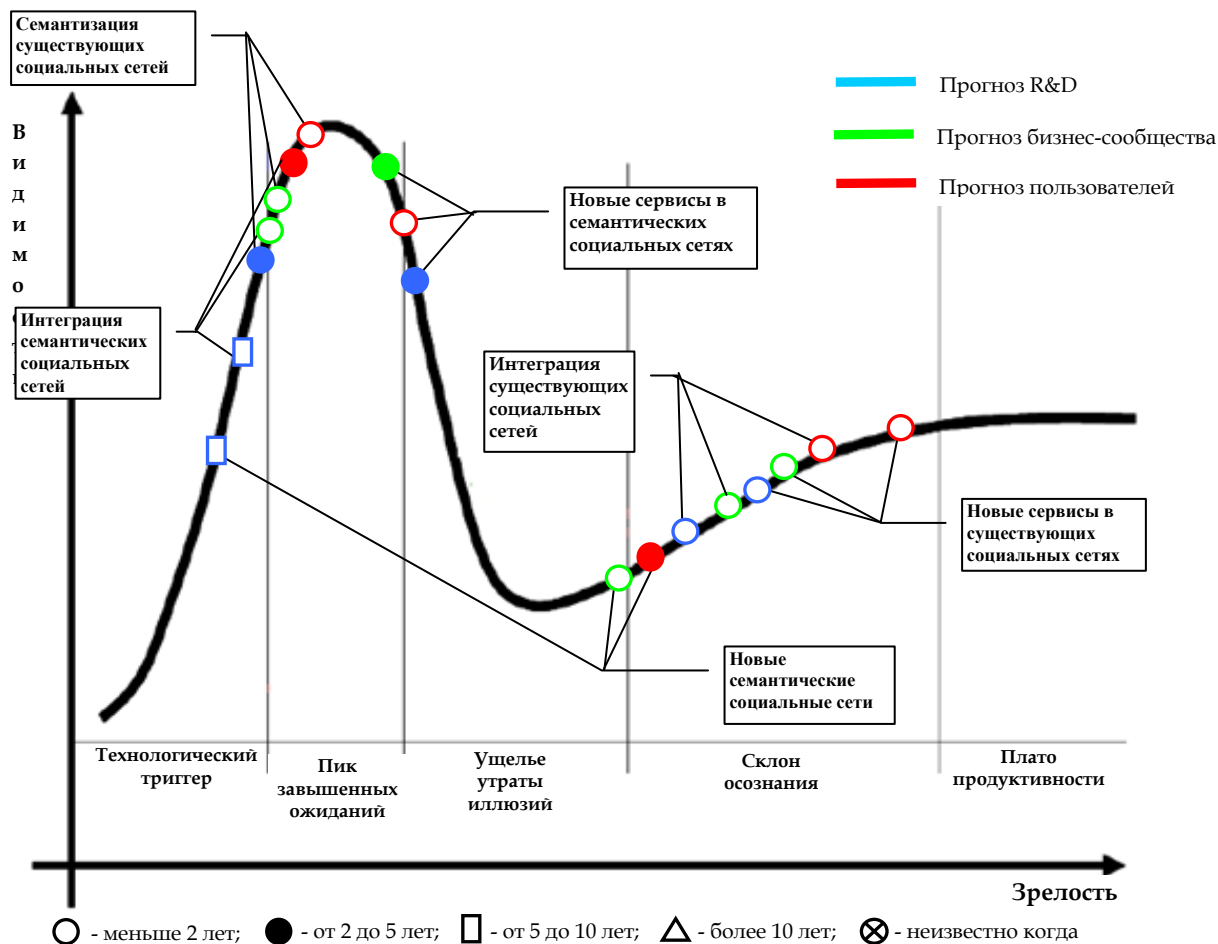


Рисунок 5 – Прогнозная кривая для технологического тренда «Социальный Семантический Вэб»

2.3.6. Разрушающий потенциал, барьеры и риски внедрения тренда

По нашему мнению, разрушающего потенциала у технологического тренда «Социальный Семантический Вэб» нет, поскольку его продукты и сервисы являются естественным развитием существующих продуктов и сервисов социального Вэба. Будучи введенными в массовое использование, продукты и услуги этого технологического тренда просто расширят рынок поставки информации потребителям, а технологический ландшафт при этом, по нашему мнению, практически не изменится. Не нарушится существенно и баланс сил в отраслевом сегменте, поскольку основные игроки сами активно продвигают данный технологический тренд.

Барьеры, риски внедрения технологии и факторы неопределенности развития технологии связаны, прежде всего, с неразвитостью законодательной базы для использования данных, циркулирующих в социальных сетях и в Интернет в целом, а также с научно-техническими проблемами семантизации контента и семантических сервисов.

2.3.7. Выводы по тренду

Как и в предыдущем случае, выше рассмотрены

некоторые из основных аспектов технологического тренда «Социальный Семантический Вэб», дано описание тренда, кратко представлены проекты, продукты и сервисы, а также новые потребительские свойства и ожидаемые эффекты тренда и дана оценка сроков внедрения приложений и возможные области его применения. Идентифицированы технологии-конкуренты и технологии, от которых зависит развитие тренда, фиксирован разрушающий потенциал, барьеры и риски внедрения тренда.

2.4. Семантические электронные библиотеки

С ростом компьютерной грамотности всё большее количество людей начинает пользоваться электронными книгами, а число читателей обычных библиотек снижается. Так за период 1997—2002 годы в Университете Айдахо число посетителей библиотеки снизилось более чем на 20 %, а число пользователей электронных версий только за период 1999—2002 увеличилось на 350 %. В связи с этим многие библиотеки начали создавать электронные версии хранящихся в их фондах книг и уже начиная с 70-х годов прошлого века стартовали проекты по созданию электронных библиотек.

Как известно, первым проектом по созданию электронной библиотеки был Проект «Гутенберг» (1971 г.). В Рунете первой электронной библиотекой стала библиотека Максима Мошкова. В 1990 году библиотекой конгресса США начат проект «Память Америки», в рамках которого предоставляется свободный и бесплатный доступ к электронным материалам по истории США. В 2002 году Google начал собственный проект по оцифровке книг, а в декабре 2004 года было объявлено о начале работы библиотечного проекта «Google Print», который в 2005 году был переименован в «Поиск книг Google». 20 ноября 2008 года начала функционировать общеевропейская цифровая библиотека Europeana, а 21 апреля 2009 года состоялось официальное открытие Всемирной цифровой библиотеки. Активно развиваются электронные библиотеки и в России. 27 мая 2009 года в Санкт-Петербурге была открыта президентская библиотека имени Бориса Ельцина, в задачи которой входит предоставления электронных материалов по истории России. Серьезные научно-практические проекты по тематике электронных библиотек ведутся и в таких организациях, как Российская Государственная Библиотека (РГБ), Национальная Электронная Библиотека (НЭБ), Государственная Публичная Научно-Техническая Библиотека (ГПНТБ), Центре Информационных Технологий и Систем (ЦИТиС), Библиотеке Естественных Наук РАН (БЕН) и в других организациях. Таким образом, технологический тренд «Электронные библиотеки» имеет достаточно долгую историю и впечатляющие результаты во всем мире и в нашей стране.

2.4.1. Описание тренда

Вместе с тем, в последние годы наблюдается формирование нового технологического тренда – «Семантические Электронные Библиотеки» (СЭБ), которые являются хранилищами семантических моделей библиотечных объектов. И, если в обычных электронных библиотеках «читатели» пользуются проиндексированными материалами на базе средств контекстного поиска, в семантической электронной библиотеке поиск материалов осуществляется на базе предварительной семантической разметки и индексации единиц хранения. Для этого используются семантические модели библиотечных объектов, где представлены модели библиотечных материалов (к-во стр., автор, и т.п.) и модели их содержания. При этом для описания таких моделей в СЭБ используются методы и средства онтологического моделирования пространств библиотечных знаний.

В настоящее время наблюдаются и процессы интеграции контента электронных библиотек, что ведет к появлению информационных пространств, основанных на федеративных принципах, а новым качеством, возникающим в результате этих изменений, является предоставление пользователям единых интерфейсов доступа к интегрированным разнородным данным на основе стандартизации

метаданных, а также унификация способов доступа к информации и способов ее извлечения для обработки и использования.

Для поддержки этого направления уже много лет ведутся исследования в области анализа семантики связей между информационными (в первую очередь, научными) материалами. Системным обобщением этих результатов стало появление комплекса онтологий SPAR [SPAR, 2012], обеспечивающих достаточно детальную категоризацию отношений, которые могут возникать между научными материалами в электронном виде, а также появление семантического раздела в модели научных данных CERIF [CERIF, 2010]. Объединение этих двух результатов – создание средств и сервисов информационных научных ресурсов, представляющих интегральный контент ЭБ, и разработка классификаторов отношений и семантических словарей, позволяющих выражать наличие определенных связей и отношений между объектами научного конкретного информационного ресурса, порождает важное новое качество – возможность разработки технологий семантического структурирования контента ЭБ. Серьезная работа в этом направлении ведется и в рамках консорциума W3C, где в проекте SKOS (Simple Knowledge Organization System) предлагается модель связывания научных данных, адаптированная для компьютерной обработки [SKOS, 2009]. В частности, SKOS включает контролируемые структурные словари семантических значений для связывания научных данных.

В дополнение к этому в различных научных дисциплинах (в первую очередь в биологии и медицине) были предприняты попытки разработать более подробную категоризацию отношений между научными текстами. Наиболее известными результатами этих попыток являются онтология SWAN (Semantic Web Applications in Neuromedicine), CiTO (Citation Typing Ontology), DoCo (Document Components Ontology) [Shotton, 2010] и др. В дальнейшем эти отдельные разработки были систематизированы, дополнены и объединены в единый комплекс под названием SPAR, включающий взаимосвязанную совокупность онтологий различного назначения. Независимо от этого, в рамках разработки концептуальной модели научных данных CERIF (Common European Research Information Format), ведутся работы по развитию стандартизированной формальной семантики для отображения отношений между объектами научных информационных систем (CRIS).

Форматы метаданных для описания электронных ресурсов, коллекций и авторитетных данных (таких как библиографические данные об электронных Интернет-ресурсах, записях о коллекции архивных материалов и др.) разрабатываются в течение многих лет и в рамках проектов Dublin Core; Metadata Object Description Schema (MODS) и Metadata Encoding and Transmission Standard,

которые поддерживаются Библиотекой Конгресса США, Encoded Archival Description (EAD), Encoded Archival Context (EAC) и др. Форматы для описания логических структур электронных изданий и отображения непосредственно их содержания (полнотекстовые и полноимиджевые статьи, электронные книги) предлагаются в проекте Text Encoding Initiative (TEI) и других проектах.

Таким образом, в настоящее время уже заложена научно-техническая база технологического тренда «Семантические Электронные Библиотеки» и можно ожидать новых продуктов и услуг, разработанных на основе использования семантических технологий.

2.4.2. Проекты, продукты и сервисы тренда

Следует сразу отметить, что в настоящее время продукты, которые бы четко позиционировались как СЭБ практически нет, а в области охвата данного технологического тренда наблюдается, в основном, активность проектной деятельности.

Среди проектов по семантическим электронным библиотекам явно выделяется проект СЭБ JeromeDL (eLibrary with Semantics) [JeromeDL, 2009] – совместный проект Main Library of Gdansk University of Technology [http://www.bg.pg.gda.pl/] и DERI.International [http://www.deri.org/]. Основными требованиями (от библиотечных работников и «читателей»), которые должны быть реализованы в рамках данного проекта, являются:

- Поддержка легальности классических библиотек.
- Обеспечение навигации по библиотечным фондам, ориентированной на пользователей (user-oriented browsing).
- Поддержка эффективного поиска нужных библиотечных единиц.
- Обеспечение защиты от несанкционированного доступа.
- Поддержка множественных форматов ресурсов.
- Возможность коммуникации с другими электронными библиотеками.
- Использование последних результатов, полученных в области Semantic Web и в области коммуникации и управления информационными ресурсами.

По сути дела, в проекте JeromeDL последовательно решаются проблемы, которые характерны для эволюции электронных библиотек (Рис. 6).

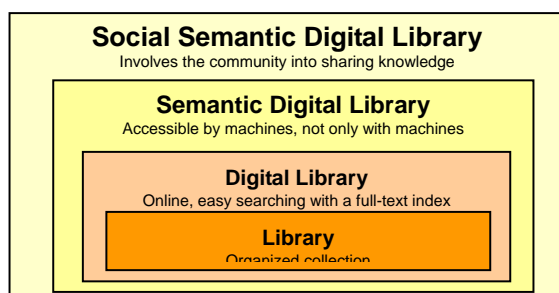


Рисунок 6 – Эволюция электронных библиотек

Проект Muruca – совместные усилия исследователей и индустрии библиотечного дела в области разработки набора Open Source приложений для создания семантических электронных библиотек и менеджирования их ресурсов [Muruca, 2012]. Инструментарий Muruca опирается на концепции Linked Open Data, Open Access content и технологии Semantic Web. Консорциум исполнителей по проекту Muruca включает Net7 – Internet Open Solutions – a european SME и Semedia Lab at Università Politecnica delle Marche, Ancona (Италия). Кроме того, проект поддерживается A-32 Open Scholarly Communities COST Action, где задействованы многие европейские центры исследований.

В проекте "Milan ICT Wired City - Open Digital Library of the city of Milan" [Occelli, 2006] выполняется при поддержке Lombard Interuniversity Consortium for Automatic Computation (CILEA) и Department of Research and Innovation of the City of Milan. В рамках проекта была создана первая семантическая электронная библиотека Италии.

Проект «A Semantic Digital Library of Urban Models and Resources» [COST, 2009] выполняется в университете Женевы (Université de Genève) в рамках инициативы COST. В проекте участвуют практически все европейские страны, а его целью является создание семантической электронной библиотеки для хранения урбанистических моделей и ресурсов. В данном проекте термин «семантика» подразумевает, что каждая модель будет семантически специфицирована связями с соответствующими концептами концептуального слоя из нескольких онтологий, ядром которого будет ontology of 3D city objects, построенная на основе CityGML. Предполагается, что библиотека будет «оборудована» Вэб-интерфейсами, с помощью которых пользователи будут импортировать модели и ресурсы, искать и сравнивать их, а также создавать и запоминать новые взаимосвязанные ресурсы.

Совместным проектом в области семантических электронных библиотек является проект SemLib [SemLib, 2012]. Консорциум исполнителей по этому проекту включает Net7 и Liberologico (Италия), IN2 (Великобритания), Knowledge Hives (Польша), National University of Ireland Galway (NUIG) - Digital Enterprise Research Institute (DERI, Ирландия), а также SeMedia group, A3Labs, Università Politecnica delle Marche, Ancona (Италия).

Мощным исследовательским проектом в области технологического тренда «Семантические Электронные Библиотеки» является проект Europeana [Europeana, 2012], поддерживаемый The Europeana Foundation, целью которого является кооперация музеев, архивов, коллекций аудио и видео материалов и библиотек таким образом, чтобы пользователи могли получить к ним доступ через сервисы Europeana.

Кроме рассмотренных выше проектов СЭБ можно отметить следующие более мелкие проекты в этой области: SWickyNotes (инструментарий для коллективного семантического аннотирования Вэб-страниц и, в частности, Digital Libraries. Первая версия инструментария (Philospace) реализована в рамках европейского проекта Discovery EU и ориентирована на изучающих философию); Talia (фреймворк на базе языка ruby для создания и кастомизации Semantic Digital Libraries); MPEG7AudioENC (активность по созданию в кооперации с Holger Crysandt (Aachen University) экстрактора метаданных MPEG-7, который бы покрывал практически полностью аудио спецификации) и др.

Таким образом, как показывает проведенный выше анализ, в области семантических электронных библиотек пока наблюдаются, в основном, проектная деятельность на уровне R&D.

2.4.3. Новые потребительские свойства и ожидаемые эффекты тренда

Как показывает анализ литературы и проектов по семантизации электронных библиотек, новые потребительские свойства, присущие технологическому тренду «Семантические Электронные Библиотеки», - следующие:

- Повышение эффективности использования многочисленных лицензионных баз данных в коллекциях библиотек, которое должно определяться
 - созданием средств получения статистики использования баз данных (Тратя значительные средства на платные базы данных, библиотеки столкнулись с проблемой оценки реального использования читателями как отдельных электронных изданий, так и баз данных в целом. Многочисленные вендоры баз данных или вообще не предоставляют такую статистику использования или предоставляют ее в разных форматах, что затрудняет ее анализ. Библиотекари настойчиво требуют включения пункта отчетности в формате COUNTER в лицензионные соглашения с владельцами баз данных);
 - обеспечением связи (linking) между различными представлениями одного и того же документа/статьи в разных базах данных. (Библиотеки, приобретая себе в коллекции тысячи электронных изданий и библиографические базы данных у разных фирм, сталкиваются с тем, что одна и та же статья может присутствовать одновременно в нескольких базах данных. При этом одна база данных, например, может содержать эту статью только в виде текста, другая в виде PDF, а третья—ее библиографическое описание);
 - одновременным поиском по различным базам данных (Multi-database или Federated

search). (Сегодня читатели вынуждены производить один и тот же поиск по интересующим их словам во все большем количестве баз данных. Последние отличаются друг от друга интерфейсами, навигацией по экранам и даже правилами запросов к системе. Федеративный поиск (Federated search) призван облегчить эту задачу путем осуществления поиска одновременно по многим базам данных и группировки/представления результатов поиска на одном экране для удобства читателей).

- Улучшение электронных каталогов (OPACs) и Вэб-сайтов библиотек, поскольку, например, американские библиотекари с беспокойством наблюдают процесс отчуждения читателей от библиотек и их электронных каталогов в пользу простых Интернет-поисковиков, которые индексируют Вэб сайты вне зависимости от их качества и авторитетности. Среди возможных способов повышения привлекательности и полезности библиотечных ресурсов и Вэб сайтов библиотеки видят, в частности, следующие:
 - Улучшение и повышение привлекательности для читателей электронных каталогов библиотек путем добавления (или линкования) дополнительной информации к традиционным кратким библиографическим описаниям изданий в формате MARC, интеграции с соответствующими метаданными, изображениями и полными текстами в других информационных системах, а также предоставления расширенных комплектов метаданных для интеграции с электронными библиографическими записями.
 - Создание и предоставление доступа к более обширным и авторитетным коллекциям электронных ресурсов путем сбора (harvesting) метаданных с использованием протокола Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) и перехода от устаревающих форматов представления записей к стандарту XML для описания коллекций.
 - Оптимизация электронных каталогов и Вэб-сайтов библиотек для облегчения их индексирования бесплатными поисковыми системами Интернета.
 - Внедрение круглосуточных справочно-библиографических услуг по электронной почте и через Вэб-сайты библиотек (electronic, online или red-eye reference service).
- Внедрение новых стандартов для электронных библиографических записей и документов, осуществления взаимосвязи между ними в базах данных и обмена информацией, что

должно обеспечить использование единого языка для обмена электронной информацией между библиотеками, архивами и другими организациями, внести единообразие в описания и структуры электронных ресурсов, способствовать их миграции с технологически устаревающих носителей и баз данных на новые, сохранению для потомков.

Интеграция метаданных отдельных ЭБ и новые средства семантического индексирования библиотечных ресурсов обеспечивает множество полезных возможностей как разработчикам, так и пользователям. При этом главным положительным моментом формирования информационного библиотечного пространства на основе контента отдельных ЭБ является предоставление пользователям единых интерфейсов доступа к интегрированным разнородным информационным ресурсам на основе стандартизации метаданных, а также унификация способов доступа к информации и способов ее извлечения для обработки и использования.

Среди важнейших ожидаемых эффектов технологического тренда СЭБ, в первую очередь, целесообразно выделить социальный (связан с существенным увеличением качества библиотечных ресурсов, представленных в электронных библиотеках, и, как следствие, повышением роли библиотек в образовательных процессах, с повышением эффективности доступа к библиотечным ресурсам и, как следствие, увеличением числа «читателей» в электронных библиотеках); экономический (связан с существенным увеличением количества «читателей» в семантических электронных библиотеках, и, как следствие, повышением отдачи от таких библиотек для государства и коммерческих организаций, поддерживающих библиотечные ресурсы) и политический (связан с формированием законодательной базы электронных библиотек и, как следствие, с повышением легитимности библиотечных ресурсов, представленных в семантических электронных библиотеках).

2.4.4. Оценка сроков внедрения приложений тренда

Работы в рамках технологического тренда «Социальный Семантический Вэб» ведутся, в основном, на стадиях фундаментальных и прикладных исследований и, редко, на стадии создания опытных образцов и запуска их в «опытную эксплуатацию». Учитывая это, и прогнозы по трендам «Открытые Связанные Данные» и «Семантический Социальный Вэб», представленные выше, реального появления на рынке первых важнейших продуктов и/или услуг в рамках технологического тренда СЭБ можно ожидать не ранее 2015 – 2017 г.г. При этом необходимо отметить, что первые семантические сервисы в существующих электронных библиотеках, которые, собственно, и сформируют сам технологический тренд «Семантические Электронные Библиотеки», появятся раньше. По

нашей оценке это произойдет в 2013 – 2015 г.г., а массового производства продуктов и услуг, предоставляемых семантическими электронными библиотеками «читателям», можно ожидать в 2016-2018 г.г.

Прогнозная кривая технологического тренда СЭБ, представленная на Рис. 7, показывает, что между бизнес-сообществом и потребителями нет согласованного мнения по поводу продуктов и сервисов, а сообщество R&D смотрит на научно-технические проблемы пессимистичнее, чем разработчики, поскольку считает, что переход к семантическим электронным библиотекам потребует серьезных исследований и разработок по семантической индексации информационных ресурсов существующих цифровых библиотек и практически значимых результатов в области семантизации контента и семантического поиска.

2.4.5. Технологии-конкуренты и технологии, от которых зависит развитие тренда

Технологиями-конкурентами к СЭБ, по нашему мнению, являются семантические социальные сети и открытые связанные данные. При этом надо отметить, что семантические электронные библиотеки, по существу, базируются на результатах указанных выше трендов и используют их для развития собственных продуктов и сервисов.

Преимущества конкурирующих направлений, по-видимому, связаны с их большей «популярностью» у молодого поколения населения, а недостатки определяются слишком широким охватом потребностей аудитории в случае социальных семантических сетей и ограничениями на открытые связанные данные, вытекающими из несовершенства законодательной базы в этой области.

По нашему мнению, развитие технологического тренда СЭБ опирается на результаты «родительского» технологического тренда ЭБ и существенно зависит от таких технологий из смежных областей, как семантические социальные сети, открытые связанные данные, онтологический инжиниринг, а также технологий семантизации контента и семантического поиска.

2.4.6. Разрушающий потенциал, барьеры и риски внедрения тренда

По нашему мнению, разрушающего потенциала у технологического тренда СЭБ нет, поскольку его продукты и сервисы являются естественным развитием существующих продуктов и сервисов технологического тренда ЭБ. Будучи введенными в массовое использование, продукты и услуги этого нового технологического тренда просто расширят рынок поставки информации потребителям, а технологический ландшафт при этом, по нашему мнению, практически не изменится. Не нарушится существенно и баланс сил в отраслевом сегменте, поскольку основные игроки сами активно продвигают данный технологический тренд.

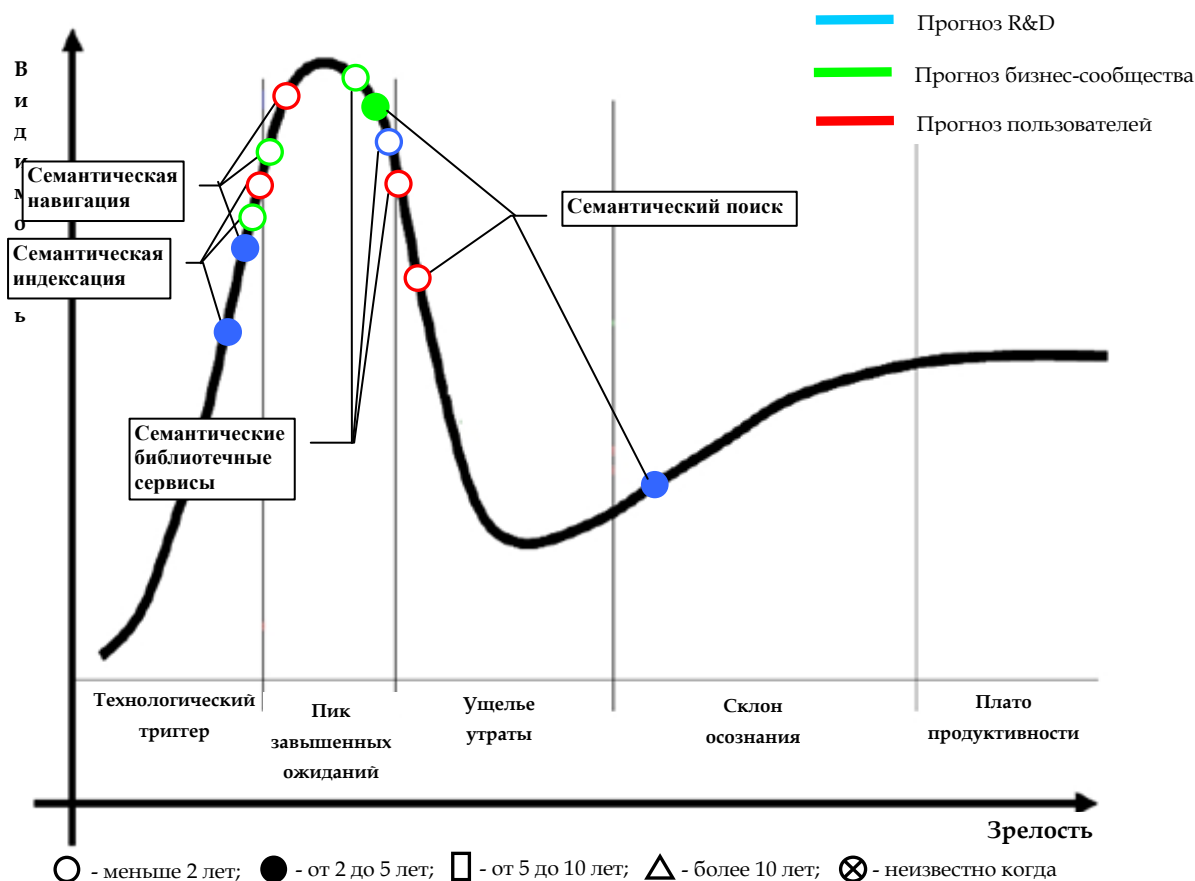


Рисунок 7 – Прогнозная кривая для технологического тренда «Семантические Электронные Библиотеки»

Барьеры, риски внедрения технологии и факторы неопределенности ее развития, по нашему мнению, связаны с недостаточностью законодательной базы для использования оцифрованных информационных ресурсов, уже представленных в электронных библиотечных коллекциях и циркулирующих в социальных сетях и в Интернет в целом, с научно-техническими проблемами семантизации контента и его семантической индексации, а также с проблемами создания действительно новых семантических сервисов.

2.4.7. Выводы по тренду

Как и в предыдущем случае, выше рассмотрены некоторые из основных аспектов технологического тренда «Семантические электронные библиотеки», дано описание тренда, кратко представлены проекты, продукты и сервисы, а также новые потребительские свойства и ожидаемые эффекты тренда и дана оценка сроков внедрения приложений. Идентифицированы технологии-конкуренты и технологии, от которых зависит развитие тренда, фиксирован разрушающий потенциал, барьеры и риски внедрения тренда.

ЗАКЛЮЧЕНИЕ

В работе представлены экспертные оценки по трем из основных трендов в области использования семантических технологий в настоящее время – трендам открытых связанных данных, семантического социального Вэба и семантическим электронным библиотекам. При этом рассмотрены некоторые из основных аспектов этих новых технологических трендов, приведены их описания, кратко представлены проекты, продукты и сервисы трендов, а также их новые потребительские свойства и ожидаемые эффекты и дана оценка сроков внедрения приложений. Идентифицированы технологии-конкуренты и технологии, от которых зависит развитие тренда, фиксирован разрушающий потенциал, барьеры и риски внедрения тренда.

Таким образом, в настоящей работе сформирована методологическая основа для исследований и разработок в области автоматизации процессов выявления новых технологических трендов, которая, в свою очередь, должна, по нашему мнению, базироваться на семантических технологиях.

Работа выполнялась при поддержке Программы фундаментальных исследований НИУ ВШЭ в 2011 г.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Лапшин, 2010] Лапшин В.А. Онтологии в компьютерных системах. М.: Научный мир, 2010.

[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.

[Хорошевский, 2009] Хорошевский, В.Ф., Пространства знаний в сети Интернет и Semantic Web (Часть 2), / В. Ф. Хорошевский // Искусственный Интеллект и Принятие решений, 2009. - № 4.

[Хорошевский, 2012] Хорошевский, В.Ф., Пространства знаний в сети Интернет и Semantic Web (Часть 3), / В. Ф. Хорошевский // Искусственный Интеллект и Принятие решений, 2012. - № 1 (в печати).

[Benjamins et al., 2002] Benjamins, V. R., Contreras, J., Corcho, O., Gomez-Perez, A. Six Challenges for the Semantic Web, URL: http://www.cs.man.ac.uk/~ocorcho/documents/KRR2002WS_BenjaminsEtAl.pdf

[Berners-Lee, 2003] Berners-Lee, T. The Semantic Web and Research Challenges, URL: <http://www.w3.org/2003/Talks/01-sweb-tbl/slide1-0.html>

[Berners-Lee, 2006] Berners-Lee, T. Linked Data: 2006-07-27, URL: <http://www.w3.org/DesignIssues/LinkedData.html>

[Bizer et al., 2009] Bizer, C., Heath, T., Berners-Lee, T. Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems, (2009) Vol. 5(3), Pages 1-22.

[CERIF, 2010] CERIF Data Models. URL: <http://www.eurocris.org/Index.php?page=CERIFreleases&t=1>

[CommonTag, 2012] CommonTag home page. URL: <http://com-montag.org/Home>

[COST, 2009] A Semantic Digital Library of Urban Models and Resources. Project Home Page. URL: <http://www.sbf.admin.ch/hfm/dokumentation/publikationen/international/cost/cd2011/cost/C09.0062.html>

[Europeana, 2012] Europeana Portal. URL: <http://www.europeana.eu/portal/>

[GARTNER, 2012] Gartner home page, URL: <http://www.gartner.com/technology/research.jsp>

[IDC, 2012] . IDC home page, URL: <http://www.idc.com>

[JeromeDL, 2009] JeromeDL (eLibrary with Semantics). URL: <http://www.jeromedl.org/>

[LATC, 2012] Linked Open Data Around the Clock project home page, URL: <http://latc-project.eu/>

[LOD, 2012] Linking Open Data project home page. W3C SWEO. URL: <http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

[LOD2, 2012] Linking Open Data 2 project home page, URL: <http://lod2.eu/Welcome.html>

[Mills, 2006] Mills, D. Semantic Wave 2006. Executive Guide to Billion Dollar Markets. A Project10X Special Report. January 2006.

[MOAT, 2012] MOAT (Meaning Of A Tag) home page. URL: <http://moat-project.org>

[Muruca, 2012] Muruca Home Page. URL: <http://www.muruca.org/>.

[Ocelli, 2006] Ocelli, S. Technological Convergence vs Knowledge Integration, Presented in: a Les Journées Annuelles Transdisciplinaires de réflexion au Moulin d'Andé, Colloques AFSCET, May 13-14, 2006.

[OWL, 2009] OWL, W3C Documentation, 2009, URL: <http://www.w3.org/TR/owl-guide/>

[PlanetData, 2012] PlanetData project home page, URL: <http://www.planet-data.eu/>

[ReadWriteWeb, 2010] ReadWriteWeb Top 10 Semantic Web Products. URL: http://www.readwriteweb.com/archives/top_10_semantic_web_products_of_2010.php

[SCOT, 2012] SCOT (Social Semantic Cloud of Tags) home page. URL: <http://www.scot-project.org/>

[SemLib, 2012] SemLib Project Home Page. URL: <http://www.sem-libproject.eu/>

[Shotton, 2010] Shotton, D. Introduction the Semantic Publishing and Referencing (SPAR) Ontologies. October 14,

2010. URL: <http://opencitations.wordpress.com/2010/10/14/introducing-the-semantic-publishing-and-referencing-spar-ontologies/>

[SIOC, 2012] SIOC initiative home page. URL: <http://sioc-project.org/>

[SKOS, 2009] SKOS (Simple Knowledge Organization System). W3C Recommendation 18 August 2009. URL: <http://www.w3.org/TR/2009/REC-skos-reference-20090818/>

[SPAR, 2012] Semantic Publishing and Referencing Ontologies. URL: <http://sempublishing.svn.sourceforge.net/viewvc/sempublishing/SPAR/index.html>

[SPARQL, 2008] SPARQL Query Language for RDF, W3C Recommendation 15 January 2008.

URL: <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>

[Tofler, 2006] Tofler, A. Third Wave, Bantam Books, 2006, ISBN 0-553-24698-4

[Zemanta, 2012] Zemanta home page. URL: <http://www.zemanta.com/about/>

SEMANTIC TECHNOLOGIES: EXPECTATIONS AND TRENDS

Khoroshevsky V.F.

*Federal State Budgetary Institution of Science,
Dorodnicyn Computing Centre of RAS,
Moscow, Russia*

khor@ccas.ru

State of art in the field of a new domain «Semantic technologies» is discussed in the paper. Main attention is given to the analysis of new technological trends in the field and an expert estimation of their importance and appearance of real applied products on the basis the usage of methods and tools of semantic processing of the information.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 517. 977. 52

О ВОЗМОЖНОСТИ ОРГАНИЗАЦИИ ЗНАНИЙ НА ОСНОВЕ КОГНИТИВНОЙ СЕМАНТИКИ

Кузнецов О.П.

*Институт проблем управления им. В.А.Трапезникова РАН,
г. Москва, Россия*

olkuznes@ipu.ru

В работе рассмотрены основные концепции когнитивной семантики, изложенные в книге [Lakoff, 1987]. В их основе лежит тезис, заключающийся в том, что когнитивные структуры и механизмы человека существенным образом зависят от его сенсорных механизмов, а также физического и социального опыта. Обсуждаются возможности использования этих концепций для решения задач искусственного интеллекта.

Ключевые слова: когнитивная семантика, категоризация, концептуальные структуры

ВВЕДЕНИЕ

Искусственный интеллект (ИИ), возникший в 50-х годах прошлого века как самостоятельная ветвь компьютерных наук, ставил в начале своего развития весьма амбициозные цели – моделирование интеллектуальных процессов человеческого мозга. На фоне царившей тогда романтической веры в беспредельные возможности компьютеров первые же успехи в автоматизации интеллектуальных процессов (решатель задач Ньюэлла - Саймона, подходы к распознаванию образов, моделирование некоторых простых игр) привели к формированию ведущей парадигмы ИИ, которую можно назвать компьютерной парадигмой. Она исходила из предположения, что мозг с информационной точки зрения - это большой и сложный компьютер, и поэтому интеллектуальные процессы нужно реализовать на принципах работы с информацией в компьютере, т.е. на основе символьных представлений информации и их обработки с помощью алгоритмических логико-комбинаторных процедур.

Довольно быстро стало понятно, что для решения многих интеллектуальных задач вовсе не обязательно имитировать процессы мозга. Высокие (и постоянно растущие) скорости компьютерных вычислений позволили использовать сложные последовательные алгоритмы, не характерные для мозга. Структурное несходство ИИ-процессов с процессами мозга рассматривалось как залог успеха. Ведь человек начал летать, только когда перестал пытаться имитировать полет птиц; а

вместо того, чтобы автоматизировать метлу, придумал пылесос. И действительно, в основе практически всех достижений ИИ последних десятилетий, воплощенных в интеллектуальных технологиях (экспертных системах, технологиях поддержки принятия решений, системах интеллектуального анализа данных) лежат логико-комбинаторные алгоритмы, как правило, имеющие мало общего с процессами мозга. Соответственно, компьютерная парадигма приобрела несколько другой вид: «неважно, что между процессами мозга и компьютера имеются существенные различия; важно, что все интеллектуальные задачи можно решать на компьютере».

Однако со временем стало очевидно, что не все интеллектуальные процессы поддаются компьютерному моделированию. Оставляя в стороне такие очевидные явления, как творческие процессы, озарение и интуицию, которые в компьютере не моделируются вообще, рассмотрим процессы, которые в ИИ моделируются. Эти процессы у человека и компьютера тоже различаются. Человек не только быстро распознает; он еще быстро рассуждает и быстро принимает решения. Эти скорости сравнимы с компьютерными - при том, что скорость передачи сигналов в нервных сетях в миллион раз меньше электронной. Это говорит не только о том, что информационные процессы мозга устроены не так, как компьютерные (что само по себе не слишком волнует сторонников компьютерной парадигмы), но прежде всего о том, что некоторые интеллектуальные задачи человек решает, как минимум, в миллион раз эффективнее компьютера. Несколько упрощенно это выражается

формулой «То, что сложно компьютеру – просто человеку и наоборот, то, что сложно человеку, просто компьютеру». Для задачи понимания текста об этом сказано в [Пинкер, 2004]: «Часть, относящаяся к памяти, легка для компьютеров и тяжела для людей (оперативная память – основное узкое место при переработке информации человеком), а часть, относящаяся к принятию решений, легка для людей (по крайней мере если предложение правильно построено) и тяжела для компьютеров».

Известен ряд попыток (или хотя бы проектов) построения нестандартных моделей мыслительных процессов и понятийной системы человека (например, [Sowa, 1984], [Hofstadter et al., 1994], [Кузнецов, 1995], [Голицын и др., 1996]). К сожалению, они либо локальны (т.е. способны объяснить только незначительную часть процессов мозга), либо неконструктивны, т.е. остаются на уровне нереализованных идей. Здесь мы хотим обратить внимание, на еще одно перспективное направление когнитивных наук, которое началось с исследований Э.Рош [Rosch, 1975] и наиболее полно изложено в книге Лакоффа [Lakoff, 1987]. Нас будет интересовать не вопрос адекватности теорий этого направления реальным процессам человеческого мышления, а возможности использования этих теорий в интеллектуальных технологиях.

1. Когнитивная категоризация

Концепция Лакоффа представляет собой проект решения двух проблем: проблемы категоризации и проблемы семантики. Предлагаемый подход к этим проблемам можно назвать когнитивным. В его основе лежит тезис, заключающийся в том, что когнитивные структуры и механизмы человека существенным образом зависят от его сенсорных механизмов, а также физического и социального опыта. Когнитивный подход резко противопоставляется традиционному подходу, который Лакофф называет объективизмом и который предполагает, что когнитивные структуры человека подобны структурам научного знания и в конечном счете – структурам объективного мира, описываемого этим знанием.

Проблема категоризации – это проблема формирования понятий (категорий) и структурирования понятийной системы человека. Традиционная формальная теория понятий рассматривает понятие как класс объектов, обладающих одинаковым набором признаков. Этими объектами могут быть либо подклассы, т.е. более узкие понятия, либо единичные объекты – экземпляры. С этой точки зрения все объекты класса равноправны, т.е. любой объект класса в равной мере может служить его представителем (примером). Иерархия понятий строится от элементарных объектов (экземпляров) к классам, которые в свою очередь являются объектами более общих классов и т.д. Понятия любой степени общности имеют одну и ту же древовидную

структуру – за исключением элементарных объектов, которые наиболее просты и структуры не имеют. Рош показала, что человеческая категоризация устроена не так. В категориях человека существуют прототипы – «хорошие» (центральные, репрезентативные, типичные) примеры. Хорошо известны стандартные ответы: поэт – Пушкин, птица – воробей (или ворона), фрукт – яблоко. Это говорит о том, что человеческие категории имеют внутреннюю структуру, которая для разных видов категорий может быть различной. В человеческой иерархии понятий базовые понятия, которые когнитивно наиболее просты, находятся «в середине» иерархии общего-конкретного. Обобщение происходит вверх от базового уровня, специализация – вниз. Примерами места базовых категорий в концептуальной иерархии могут служить цепочки: животное-собака-овчарка; мебель – стул – качалка, где базовыми категориями являются «собака» и «стул».

Для категорий базового уровня характерно следующее:

- они имеют единый ментальный образ (гештальт); быстро узнаются;
- в качестве их имен используются наиболее короткие и общеупотребительные слова;
- большинство признаков членов категории хранится на этом уровне;
- формирование категорий у детей начинается с категорий базового уровня.

Возможные структуры категорий весьма разнообразны и не исчерпываются множествами элементов с одинаковым набором признаков. В книге Лакоффа развита довольно детальная (хотя не вполне формальная) типология категорий, многие из которых имеют нестандартную структуру. Таковы, например, радиальные категории, имеющие структуру «центр-периферия». В этой структуре центром является некоторая наиболее представительная субкатегория, а периферию образуют ее расширения. Пример – категория «мать», где центр – субкатегория с традиционными признаками матери (родившей и воспитавшей ребенка), а периферия – субкатегории «приемная мать», «суррогатная мать» и т.д. Передача признаков от центра к расширениям в общем случае нетранзитивна, и может оказаться, что какая-то периферийная субкатегория не обладает многими существенными признаками центра. Прототипом в таких категориях служит центр. Другой вид категорий с прототипическими эффектами – градуированные категории с различными степенями членства. Такие категории удобно описывать в терминах функций принадлежности, известных в нечеткой логике.

(Интересно отметить, что идея центральности – одно из важных понятий гештальт-психологии. «Центрирование – то, как мы рассматриваем части, отдельные элементы ситуации, их значение и роль

по отношению к центру, сути или корню, - является наиболее важным фактором в мышлении» [Вертгеймер, 1987]. Существенность или несущественность признаков для гештальт-психологов связана не с их внутренними свойствами, а с их ролью в структуре образа. Если какой-то несущественный в данном образе признак объявить (или “увидеть”) существенным, то образ может стать “другим” - изменится его интерпретация. Этот эффект в гештальт-психологии называется перецентрированием.)

Среди механизмов, формирующих категории и их организацию в понятийные структуры, можно выделить 4 типа: *классический*, организующий категории по принципам, близким к традиционным классификациям; *образно-схематический*, в основе которого лежат схематические образы, такие как траектории, вместилища, верх-низ; *метафорический*, который отображает категории одной области в соответствующие структуры другой области; *метонимический*, в котором часть структуры (субкатегория) замещает или обозначает категорию в целом (например, «Кремль» может пониматься как «Администрация президента», а «Москва» - как обозначение федеральной власти России). Многочисленные примеры образования понятий с помощью метафор имеются в математике: пространства, числовая прямая, линейность, выпуклость, рост (функции), теория графов (пути, связность т.д.). Интересно, что теория бинарных отношений, которые формально изоморфны графам, имеет совершенно другую структуру – именно потому, что в отличие от теории графов, она не имеет общего гештальта.

2. Когнитивная семантика

Еще более радикальным является предлагаемый Лакоффом проект когнитивной семантики. Этот проект принципиально отказывается от общепринятого в настоящее время формально-логического подхода, согласно которому независимо существуют синтаксис, модельные структуры и интерпретация, т.е. принципы отображения синтаксиса на модели. Формально-логический подход, основанный на идеях Гильберта – Тарского, сыграл существенную роль в становлении компьютерной парадигмы, на которой основаны практически все достижения в области интеллектуальных систем (если не считать нейросетевых методов). Однако он не способен объяснить многие особенности человеческого мышления, которыми мозг отличается от компьютера.

В общих чертах когнитивный подход к семантике заключается в следующем. Термины, в которых мыслит человек, значимы с самого начала. В отличие от компьютера, люди не оперируют незначимыми символами. «Когда человек доказывает утверждение, он опирается на его смысл, не строя формального доказательства. Разумный человек, не изучавший логику, способен

точно выражаться и быстро определять противоречия, но часто не может построить формальное доказательство» [Sowa, 1984]. Более того, по мнению Лакоффа, значения возникают раньше, чем формируются концептуальные структуры: они возникают из нашего доконцептуального телесного опыта. Доконцептуальные структуры – это гештальты и образно-схематические схемы, такие как вместилище, верх-низ, часть-целое, центр-периферия и т.д. Человеческая логика рождается из этих схем. Например, схема «вместилище» порождает рассуждения типа «если вместилище A находится во вместилище B и x находится в A , то он находится в B », обобщением которых является *Modus Ponens*. Известные в теории множеств диаграммы Венна – фактически гештальты этой схемы.

Концепты (категории), связанные с доконцептуальными структурами, являются непосредственно значимыми. Они-то и являются категориями базового уровня. Все такие категории имеют гештальтную структуру. В этих терминах можно сформулировать одну из важнейших проблем психологии, совершенно чуждую компьютерной парадигме – *что такое понимание*. Предложение понимается непосредственно, если концепты, содержащиеся в нем, непосредственно значимы. Понимание – это способность соотносить концепты со своим опытом, включая доконцептуальный. Абстрактные структуры мы понимаем в терминах образных схем. В частности, иерархическая структура понимается в терминах схем часть-целое и верх-низ, радиальная структура категорий – в терминах центр-периферия, количественная шкала – в терминах *верх-низ* и *линейный порядок*.

Различия концептуальных систем в разных культурах проистекают в основном из различий в доконцептуальном опыте. Например, концептуальная система эскимосов сильно отличается от концептуальных систем горных племен. Именно с различиями концептуальных систем связаны трудности перевода.

В заключение укажем еще на одну проблему, которая у Лакоффа только намечена – проблему когнитивной сложности. Это понятие подразумевает объем когнитивных усилий, которые прилагаются при запоминании и использовании различных концептов. С этой точки зрения наиболее когнитивно просты концепты базового уровня.

3. Когнитивный подход к задачам искусственного интеллекта

Резюмируя, выделим следующие основные идеи и проблемы когнитивной семантики:

- центральная роль гештальта,
- наличие понятий базового уровня,

- нестандартные структуры категорий: радиальные, метонимические и т.д.,
- доконцептуальное структурирование и базовые структуры опыта;
- проблема понимания;
- проблема когнитивной сложности.

Концепции Лакоффа дают возможность по-новому взглянуть на некоторые проблемы ИИ. В первую очередь следует выделить две важных области ИИ: организация знаний и формализация рассуждений. Для организации знаний и, в частности, построения онтологий представляют существенный интерес идеи, связанные с категоризацией (на это обращается внимание в недавнем докладе [Гаврилова и др., 2011]). Что касается формализации рассуждений, то здесь важно иметь в виду следующее. Человек не рассуждает по законам формальной логики. Такие рассуждения нереализуемы в реальном времени ввиду огромного количества последовательных элементарных шагов. Даже строгие математические доказательства на много порядков короче рассуждений, формализованных в стиле логики предикатов. Еще в книге [Sowa, 1984] отмечалось, что человеческая способность быстро рассуждать (на которую, к сожалению, обращается гораздо меньше внимания, чем на способность быстро узнавать) основана на использовании образно-схематических структур. В книге Лакоффа, по-видимому, впервые развита подробная типология таких схем.

Можно наметить следующие направления исследований в русле изложенных идей:

- формализация типологий когнитивных категорий и организация знаний на их основе;
- исследование и формализация проблемы гештальта и связи гештальтов с понятиями базового уровня; на решение этой проблемы была, в частности, ориентирована модель псевдооптической нейронной сети [Кузнецов, 1996];
- формализация быстрых рассуждений на основе образно-схематических структур;
- формализация понятия когнитивной сложности;
- реализация формализованных моделей на микроуровне, в частности, на уровне различных видов нейронных сетей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Вертгеймер, 1987] Вертгеймер. М. Продуктивное мышление. / М.: Прогресс, 1987.

[Гаврилова и др., 2011] Гаврилова Т.А., Болотникова Е.С., Гулякина Н.А. 2011. Категоризация знаний для создания онтологий. / Материалы 4-й Всероссийской мультиконференции по проблемам управления МКПУ-2011, Т.1. Таганрог: изд. ТТИ ЮФУ, с. 62-66.

[Голицын и др., 1996] Голицын Г.А., Фоминых И.Б. Нейронные сети и экспертные системы: перспективы интеграции. // Новости искусственного интеллекта, 1996, №4.

[Кузнецов, 1995] Кузнецов О.П. Неклассические парадигмы искусственного интеллекта. //Теория и системы управления, 1995, N5, с. 3-23.

[Кузнецов, 1996] Кузнецов О.П. Псевдооптические нейронные сети - прямолинейные модели. //Автоматика и телемеханика, N12, с.160-172.

[Пинкер] Пинкер С. Язык как инстинкт. / М.: Едиториал УРСС, 2004.

[Hofstadter et al., 1994] Hofstadter D and Fluid Analogies Research Group. Fluid concepts and creative analogies: computer models of fundamental mechanisms of thought. - Basic Books, 1994.

[Lakoff, 1987] Lakoff J. Women, Fire, and Dangerous Things: What Categories Reveal About the Mind. - University of Chicago Press. 1987. (Русский перевод: Лакофф Д. Женщины, огонь и опасные вещи: Что категории языка говорят нам о мышлении, - М. 2004).

[Rosch, 1975] Rosch E. Cognitive representations of semantic categories. Journal of Experimental Psychology, 1975, 104, pp.192-233.

[Sowa, 1984] Sowa J.F. Conceptual Structures - Information Processing in Mind and Machines. - Addison-Wesley Publ.Comp. 1984.

THE POSSIBILITY OF KNOWLEDGE STRUCTURING BASED ON COGNITIVE SEMANTICS

Kuznetsov O. P.

*Trapeznikov Institute of Control Sciences, Russian
Academy of Sciences, Profsoyuznaya ul., 65,
Moscow, 117997 Russia*

olkuznes@ipu.ru

The essential conceptions of cognitive semantics [Lakoff, 1987] are considered. These conceptions based on the thesis that cognitive structures and mechanisms of a man depend on his sensor mechanisms, physical and social experience. The possibilities of applications of these conceptions for artificial intelligence problems are discussed.

Keywords: cognitive semantics, categorization, conceptual structures.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8

ФОРМАЛЬНЫЕ ОНТОЛОГИИ

Плесневич Г.С.

*Московский энергетический институт (Национальный исследовательский университет)
г. Москва, Россия*

salve777@mail.ru

Излагаются основные понятия, касающиеся онтологий. Даются уточненные определения формальной онтологии и связанных с ней понятий. Также дается краткое описание системы «Бинарная Модель Знаний», предназначенной для построения онтологий.

Ключевые слова: искусственный интеллект, представление знаний, онтология, Семантический Веб.

ВВЕДЕНИЕ

С первых работ по искусственному интеллекту проблемы представления знаний рассматривались как центральные в этой области. В 80-х годах прошлого века в отдельных работах, связанных с представлением знаний, стал применяться термин «онтология» как вычислительная модель предметной области или как компонента системы, основанной на знаниях. Но широкое использование этого термина наблюдалось с начала 90-х годов. В настоящее время онтологии стали важнейшим инструментом в различных областях информатики (а не только в искусственном интеллекте).

В настоящей статье мы даем некоторые уточнения понятия онтологии и связанных с ней понятий. Определение онтологии основано на формальной модели понятия в экстенциональном аспекте.

1. Что такое онтология?

Решение задачи в любой области (математика, физика, экономика, инженерное дело, информатика и т.д.) включает рассмотрение объектов и манипуляцию ими. Объектами могут быть как физические (конкретные) сущности, так и ментальные (абстрактные). Таким образом, с задачей ассоциируется множество объектов, называемое *проблемной*, или *предметной областью*. Предметная область, как правило, состоит из объектов, обладающих структурой, и имеются также структуры, связывающие эти объекты друг с другом. Понимание предметной области достигается с помощью понятий, которые классифицируют ее объекты и связи между ними. *Концептуализация* предметной области – это результат фиксации этих понятий.

Концептуализация представляет знание о

предметной области, а не о конкретном состоянии дел в этой области. Иными словами, концептуализация неподвержена изменениям или меняется очень редко. Состояние дел, напротив, часто изменяется. Конкретное состояние дел в предметной области представляется множеством фактов.

Если мы хотим автоматизировать решение задач (что, в частности, служит основной целью искусственного интеллекта и, вообще, информатики), то нужно иметь точные, формальные спецификации концептуализаций.

Каждый *агент* (человек или программная система) имеет свою базу знаний, и только то, что выразимо с помощью онтологии, может быть запомнено и использовано в этой базе знаний. Когда агент желает общаться с другим агентом, он использует конструкторы из некоторой онтологии. Для того чтобы общение было понятным агентам, понимание онтологии должно *разделяться* этими агентами. Таким образом, можно дать следующее определение понятия «онтология» (принадлежащее Т. Груберу [Gruber, 1993]). *Онтология* – это явная формальная спецификация концептуализации, разделяемая некоторым сообществом агентов.

Замечание. Термин «онтология» взят из философии, где он понимается как ветвь метафизики, изучающая наиболее фундаментальные категории бытия, или существования. В информатике «существовать» – значит «быть представленным в компьютере». Поэтому термин «онтология» в информатике имеет значительно более конкретный смысл.

Онтология представляет знание о предметной области, выраженное в общих ее связях и закономерностях. Конкретное знание выражается при помощи фактов, структурированных по этой онтологии. Совокупность этих фактов, называемую *базой фактов*, можно рассматривать как *состояние* онтологии. Онтология, взятая вместе с базой

фактов, составляет *базу знаний* о предметной области. Таким образом, если O – онтология и FB – база фактов (структурированных по онтологии O), то база знаний есть $KB = O \cup FB$.

В базе знаний KB онтология O является стабильной компонентой, а база фактов FB , как правило, переменная. Можно ввести параметр γ , значение которого фиксирует базу фактов и, следовательно, базу знаний: $KB_\gamma = O \cup FB_\gamma$. Этим параметром может быть время, контекст, положение в пространстве, ситуация и т.п. Параметр γ будем называть *точкой соотнесения*.

Онтология состоит из предложений некоторого языка, который должен быть формальным. Факты также являются предложениями (существенно более простого) формального языка. Другими словами, базы знаний (онтологии и базы фактов) строятся как описания, сделанные в (формальных) языках представления знаний.

Концептуализация предметной области осуществляется с помощью понятий. Онтологии поэтому, как формальные описания, должны использовать формальные представления понятий. В разделе 3 мы рассмотрим общую формальную модель понятия.

2. О применении онтологий

С самого начала исследований по искусственному интеллекту центральными были (и остаются) проблемы представления знаний. Понятие онтологии, в сущности, относится к представлению знаний. Но в последние годы работы по онтологиям вышли за пределы теоретических исследований. Разработка онтологий стала занятием инженеров – экспертов по предметным областям.

Онтология дает общий словарь терминов для экспертов и пользователей, которым необходимо совместно использовать информацию в данной предметной области.

Практическое значение онтологий резко возросло в связи с распространением Интернета. В Вебе (WWW) онтологии используются в Google для классификации веб-сайтов. Компания Amazon, осуществляющая продажу через Интернет, разработала онтологию товаров и услуг с их характеристиками. Другой пример – онтология UNSPSC (United Nations Standard Products and Services Code – система ООН стандартных кодов для товаров и услуг).

В настоящее время разрабатываются стандартные онтологии, которые могут использоваться совместно экспертами по предметным областям. Имеются различные факторы, определяющие потребность в построении онтологий. Главные из них следующие:

- необходимость общего понимания экспертами и пользователями (или программными агентами) знания о предметной области для совместного использования этого знания;
- возможность повторного использования знания о предметной области;
- запись в точной явной форме неявных

допущений о предметной области;

- анализ знания о предметной области.

Совместное использование знаний – одна из главных целей построения онтологий. Пусть, например, несколько веб-сайтов содержат информацию в некоторой области. Если эти веб-сайты совместно имеют одну и ту же онтологию терминов, то становится возможным применение программных агентов, которые могут извлекать информацию из этих сайтов, создавая соответствующую базу знаний. Программные агенты могут использовать эту базу знаний для ответов на запросы пользователей или использовать эти знания как входную информацию для других приложений.

Возможность повторного использования знания – важная черта процесса построения онтологий. Если одни эксперты построили онтологию некоторых понятий, то другие могут просто повторно использовать ее при построении онтологий для своих предметных областей. Если эксперту нужно построить большую онтологию, то он может интегрировать несколько известных онтологий, описывающих части большой предметной области.

Формулирование явных допущений о моделируемой предметной области позволяет в дальнейшем менять эти допущения, когда изменяется наше знание о предметной области. Это формулирование выполняется в декларативной форме. Заметим, если использовать процедурное представление допущений (в обычных языках программирования), то эти допущения не только было бы сложно понять, но также сложно изменять.

Декларативное представление знаний в онтологиях позволяет использовать логические методы анализа онтологий. Важный вопрос анализа – является ли данная онтология непротиворечивой или противоречивой. Этот вопрос возникает, например, при интеграции онтологий. Другой вопрос – являются ли данные предложения онтологии избыточными.

В настоящее время онтологии рассматриваются как весьма полезный формальный инструмент для представления семантики для многих приложений. Например, онтологии применяются в следующих областях:

- Программная инженерия;
- Семантический Веб;
- веб-сервисы;
- бизнес-информатика;
- биоинформатика;
- многоагентная технология;
- электронное обучение;
- машинная обработка естественного языка.

3. Формальные понятия и формальные онтологии

Американский толковый словарь Webster следующим образом определяет термин «понятие»: An abstract or general idea inferred or derived from specific instances (Абстрактная или общая идея,

выведенная или произведенная из конкретных примеров.)

Понятие есть ментальная сущность (объект), а ее примерами (экземплярами) могут быть как физические, так и ментальные сущности. Для того чтобы моделировать понятия в компьютере, необходимо символическое, формальное представление понятий и их примеров. Такое представление основано на именах, которые присваиваются (реально или потенциально) моделируемым объектам. Между именами и моделируемыми объектами имеется отношение референции *ref*, которое назначает объекту его имя. Вообще говоря, данный объект может иметь не одно имя, а несколько. Мы скажем, что имена *a* и *b* *коррелативны* и запишем $a \sim b$, если они обозначают один и тот же объект. Таким образом, $a \sim b$ тогда и только тогда, когда существует объект *o* такой, что $o \text{ ref } a$ и $o \text{ ref } b$. Ясно, что коррелативность является отношением эквивалентности.

Понятие имеет два аспекта – экстенционал и интенционал. *Экстенционал* (или объем) понятия – это множество всех его примеров. *Интенционал* – это совокупность условий, характеризующих примеры понятия.

Формальное понятие строится из имен. В экстенциональном аспекте *формальное понятие* имеет следующие компоненты:

- имя понятия *C*;
- *универсум* понятия U^C – множество всех имен, которые могут обозначать примеры понятия *C*;
- *множество точек соотнесения* Γ ;
- подмножество $E_\gamma^C \subseteq U^C$ для каждой точки соотнесения $\gamma \in \Gamma$ – *множество всех примеров* понятия *C* в этой точке (точнее, имен примеров);
- отношение эквивалентности \sim_γ , заданное на множестве E_γ^C для каждой точки соотнесения $\gamma \in \Gamma$ – *коррелативность* в этой точке;
- пара $\text{Ext}_\gamma^C = (E_\gamma^C, \sim_\gamma)$ для каждой точки соотнесения $\gamma \in \Gamma$ – *экстенционал* понятия *C* в этой точке;
- семейство $\text{Ext}^C = \{ \text{Ext}_\gamma^C \mid \gamma \in \Gamma \}$ – *полный экстенционал* понятия *C*.

Объекты предметной области обладают, как правило, свойствами, или *атрибутами*. Атрибут *A* формального понятия *C* можно рассматривать как функцию, заданную на множестве $E^C = \bigcup \{ E_\gamma^C \mid \gamma \in \Gamma \}$. Значением этой функции служит имя примера другого понятия (или этого же понятия) или элемент некоторого типа данных.

Пример 1. Рассмотрим задачу построения системы, хранящей сведения об учебном процессе. Тогда можно ввести такие понятия, как «студент», «преподаватель», «предмет» и т.п. Предположим, что понятие «студент» имеет атрибуты ФИО, ГодРожд (год рождения), Руков (руководитель). Формальное понятие «студент» определим как имеющее следующие компоненты.

- Имя понятия: Студент;
- Универсум понятия:
 $U^{\text{Студент}} = \text{iName} \cup \text{Surr}$

{[ФИО:х, ГодРожд:у, Руков:z] |
 $x \in \text{String}, y \in \text{Integer}, z \in \text{Surr}$ }

Здесь *iName* – тип данных, элементы которого используются для имен индивидуальных объектов, а *Surr* – тип данных, элементами которого служат так называемые *суррогаты*, т.е. стандартные системные имена, идентифицирующие объекты – примеры понятий. Элементами типа данных *Surr* будем считать #1, #2, #3, ...

- Множество точек соотнесения:

$\Gamma = \{ \text{сем1}, \text{сем2}, \dots, \text{сем10} \}$.

Эти точки представляют семестры. Они выбраны потому, что информация об учебном процессе может меняться каждый семестр.

- Экстенционалы $\text{Ext}_{\text{сем}j}^{\text{Студент}}$ ($j = 1, 2, \dots, 10$) могут быть представлены таблицами. Например, для $\text{Ext}_{\text{сем2}}^{\text{Студент}}$ можно указать следующую таблицу.

Студент

Surr	Coref	ФИО	ГодРожд	Руков
#60	['староста группы А13']	'Андреев В.М.'	1993	#11
#61	[]	'Борисов И.В.'	1992	#11
:				
#97	[]	'Яковлев Л.Я.'	1992	#12

Здесь *Surr* – атрибут, значениями которого служат суррогаты, а *Coref* – атрибут, значениями которого служат списки коррелативных имен. Первая строка таблицы представляет три коррелативных имени:

#61 ~ 'староста группы А13' ~ [ФИО:'Андреев В.М.', ГодРожд:1993, Руков:#11].

Таким образом, имеем

$E_{\text{сем2}}^{\text{Студент}} = \{ \#60, \#61, \#97, \text{'староста группы А13'}, [\text{ФИО:'Борисов И.В.', ГодРожд:1993, Руков:#11}], \dots, [\text{ФИО:'Яковлев Л.Я.', ГодРожд:1993, Руков:#11}] \}$.

- Отношение коррелативности $\sim_{\text{Студент}_{\text{сем}j}}$ определяется из строк таблицы для точки соотнесения *семj* так, как это было показано выше.

- Экстенционал понятия *Студент* в точке соотнесения *семj* есть пара $\text{Ext}_{\text{сем}j}^{\text{Студент}} = (E_{\text{сем}j}^{\text{Студент}}, \sim_{\text{Студент}_{\text{сем}j}})$.

- Полный экстенционал понятия *Студент* есть семейство

$\text{Ext}^{\text{Студент}} = \{ \text{Ext}_{\text{сем1}}^{\text{Студент}}, \text{Ext}_{\text{сем2}}^{\text{Студент}}, \dots, \text{Ext}_{\text{сем10}}^{\text{Студент}} \}$.

Мы предполагаем, что руководителем студента является некоторый преподаватель. В формальном понятии *Студент* этому соответствует атрибут *Руков*, значениями которого служат суррогаты, обозначающие примеры понятия *Препо*. Экстенционал этого понятия также может быть представлен таблицей, например, таблицей с атрибутами *ФИО*, *Должность* и *Кафедра* (кроме

стандартных атрибутов Surf и Coref). Предположим, что эта таблица содержит следующую строку.

#11	[]	‘Вагин В.Н.’	Профессор	ПМ
-----	----	--------------	-----------	----

Применение атрибута A к имени e , обозначающего пример понятия, записывается как $e.A$. В данном случае, например, имеем
 #60.ФИО = ‘Андреев В.М.’,
 #60.ГодРожд = 1993,
 ‘староста группы А13’.ФИО = ‘Андреев В.М.’,
 #60.Руков = #11, #60.Руков.ФИО = #11.ФИО = ‘Вагин В.Н.’.

Как мы видели, экстенционал понятия C в каждой точке соотнесения $\gamma \in \Gamma$ есть пара $(E_\gamma^C, \sim_\gamma)$, состоящая из множества имен E_γ^C , обозначающих примеры понятия, и отношения кореференции \sim_γ , заданного на этом множестве имен E_γ^C . Если взять фактор-множество E_γ^C / \sim_γ множества E_γ^C по отношению эквивалентности \sim_γ , то это множество будет состоять из абстрактных объектов, которые взаимно однозначно соответствуют объектам предметной области, рассматриваемых в точке соотнесения γ .

Заметим, что в примере 1 экстенционалы в каждой точке соотнесения конечны. Но в общем случае экстенционалы формальных понятий могут быть счетными.

Данное выше определение формального понятия относится к экстенциональному аспекту термина «понятие». Но имеется также интенциональный аспект этого термина, связанный со спецификацией компонентов формального понятия. Мы используем термин «спецификация», а не термин «определение», имея в виду, что результат определения однозначен, но результат спецификации может быть неоднозначен.

В примере 1 два понятия Студент и Преподаватель связаны друг с другом с помощью атрибута Руководитель. Кроме того, эти понятия имеют одни и те же точки соотнесения.

Вообще, пусть E – конечное множество формальных понятий, имеющих одни и те же точки соотнесения. Тогда мы скажем, что E – *система понятий*. Некоторые из понятий системы могут быть связаны друг с другом. Но каждую связь можно также определить как формальное понятие. Поэтому можно считать, что система понятий E включает понятия, определяющие связи между понятиями.

Формальной онтологией для системы понятий E назовем конечное множество предложений O некоторого формального языка, специфицирующих формальные понятия из E . (В дальнейшем, говоря о формальных понятиях, будем опускать слово «формальные».)

Предположим, что все понятия $C \in E$ специфицированы с помощью предложений некоторого формального языка. Предложения этого языка в соответствии с тем, какому из компонент понятия они специфицируют, классифицируются

следующим образом.

- *Структуральные предложения.* Специфицируют универсумы понятий.
- *Логические предложения.* Специфицируют экстенционалы понятий равномерно по точкам соотнесения.

• *Транзитные предложения.* Специфицируют изменения экстенционалов при переходе от одной точки соотнесения к другой. Эти переходы определяются некоторым бинарным отношением, заданным на точках соотнесения.

Замечание. Фактически спецификация универсума является определением, т.е. результат спецификации однозначен (как это видно в примере 1).

Пример 2. Для спецификации указанного в примере 1 универсума $U^{\text{Студент}}$ понятия Студент можно взять структуральное предложение

Студент[ФИО:String, ГодРожд:Integer, Руков:Преподаватель].

Логические предложения мы понимаем очень широко. Это предложения, интерпретируемые как произвольные ограничения на экстенционалы для любой точки соотнесения. Возьмем, например, ограничение на экстенционалы понятия Студент, состоящее в том, что студенты одной и той же группы должны иметь только таких руководителей, которые работают на одной и той же кафедре. Это ограничение можно представить следующим логическим предложением:

FORALL Студент X, Y (IF X.Группа = Y.Группа THEN X.Руков.Кафедра = Y.Руков.Кафедра).

Транзитные предложения представляют ограничения, выражающие изменения экстенционалов при переходах в соответствие с бинарным отношением, заданным на точках соотнесения. В данном случае естественно взять отношение Позже, определяемое следующим образом: $\text{сем}_j \text{ Позже } \text{сем}_k \Leftrightarrow_{\text{df}} j > k$. Возьмем ограничение, состоящее в том, что для каждого студента допускается только одна смена руководителя. Тогда это ограничение можно представить следующим предложением:

FORALL Студент X; Рог Y
 (IF EXIST Рог Z; Z Позже Y; (X.Руков < Y > =/= X.Руков < Z >)
 THEN FORALL Рог V; V Позже Z; (X.Руков < V > = X.Руков < Z >).

Здесь Рог обозначает тип данных «точка соотнесения» (point of reference). Выражение X.Руков < Y > обозначает значение X.Руков в точке соотнесения Y.

Замечание. В примере 1 были приведены предложения из языков системы «Бинарная Модель Знаний» (см. раздел 4) [Plesniewicz, 2004], [Плесневич, 2005].

Пусть E – система понятий, отвечающая данной концептуализации, а O – формальная онтология, специфицирующая E . Тогда E служит моделью онтологии O . В соответствующей интерпретации все предложения из O будут истинными. Эту интерпретацию назовем *главной*. Но, поскольку результат спецификации, как правило, неоднозначен, существуют другие интерпретации онтологии O (соответствующие другим системам

понятий).

На основе этих интерпретаций можно определить отношение \models логического следствия между онтологией \mathcal{O} и логическим или транзитным предложением σ языка логической спецификации:

$\mathcal{O} \models \sigma \Leftrightarrow_{df}$ не существует интерпретации, в которой все логические и транзитные предложения из \mathcal{O} истинны, а предложение σ ложно.

Понятие логического следствия используется при анализе онтологии. Это анализ включает, в частности, решение следующих задач:

- выяснить, является ли онтология \mathcal{O} противоречивой;
- для данного логического или транзитного предложения σ выяснить, верно ли $\mathcal{O} \models \sigma$;
- для данного логического или транзитного предложения σ выяснить, является ли это предложение избыточным, т.е. верно ли, что $\mathcal{O} \setminus \{\sigma\} \models \sigma$.

База фактов FB для данной онтологии \mathcal{O} состоит из конечного множества примеров и контрпримеров понятий, входящих в эту онтологию. Точнее, пусть Δ – конечное подмножество множества Γ . Для каждой точки соотнесения $\gamma \in \Delta$ и каждого понятия $C \in \mathcal{O}$ возьмем конечное подмножество $+F_{\gamma}^C$ множества E_{γ}^C и конечное подмножество $-F_{\gamma}^C$ множества $U^C \setminus E_{\gamma}^C$. Для любых элемента $e \in U$, понятия C и точки соотнесения γ выражения $+(e, C, \gamma)$ и $-(e, C, \gamma)$ назовем фактами. Первое выражение интерпретируется как утверждение « e – пример понятия C в точке соотнесения γ », а второе – как утверждение « e – контрпример понятия C в точке соотнесения γ ». База фактов – это множество

$$FB = \{+(e, C, \gamma) \mid e \in +F_{\gamma}^C, \gamma \in \Gamma\} \cup \{- (e, C, \gamma) \mid e \in -F_{\gamma}^C, \gamma \in \Gamma\}.$$

База знаний – это множество $KB = \mathcal{O} \cup FB$.

Определение ответов на запросы к базе знаний можно определить на основе отношения логического следствия. Запрос можно рассматривать как предложение σ , составленное из принадлежащих онтологии \mathcal{O} имен и содержащее свободные индивидуальные переменные x_1, x_2, \dots, x_n . Ответ на запрос σ – это множество

$$\{(x_1, x_2, \dots, x_n) \mid KB \models \sigma\}.$$

В Семантическом Вебе онтологии играют центральную роль (см., например, [Antonioni & Harmelen, 2008]). Различные языки для онтологий в Семантическом Вебе были разработаны в последнее десятилетие. Организация W3C (World Wide Web Consortium) предложила стандарты для основных языков. Стандартным синтаксисом для этих языков является синтаксис XML. Предложения языков XML, XML Schema и RDF являются структуральными. Язык RDF Schema содержит простейшие логические предложения. Языки семейства OWL содержат логические предложения, эквивалентные языкам дескриптивной логики (description logic) [Baader et al., 2003].

Разработанный М. Кифером язык F-logic (фреймовая логика) также используется для построения онтологий. Он содержит логические

предложения, равные по выразительной силе логике первого порядка [Kifer et al., 1995]. М. Кифер также предложил так называемую логику транзакций (transaction logic), содержащую предложения, которые можно классифицировать как транзитные [Bonner & Kifer, 1994].

Замечание. В логических языках, являющимися вариантами логики первого порядка формальные понятия представляются одноместными предикатами, а их атрибуты – одноместными функциями. В языках модальной пропозициональной логики формальные понятия представляются пропозициональными переменными, причем точками соотнесения служат миры.

4. Бинарная Модель Знаний

Бинарная Модель Знаний (БМЗ) – это система концептуальных (понятийно-ориентированных) языков, предназначенная для построения онтологий.

В БМЗ имеются два типа понятий: *классы* и *бинарные связи*. (Бинарная связь представляет бинарное отношение между элементами классов).

В БМЗ используются как простые, так и составные имена для объектов, классов и бинарных связей. Простое имя класса или имя бинарной связи представляется слитной строчкой символов, начинающееся с большой буквы, и без знаков препинания и специальных символов, а также произвольной строчкой символов, взятых в одиночные кавычки. Составные имена классов и бинарных связей получаются применением соответствующих операций.

• $C(*)$ обозначает класс, примерами которого служат конечные множества примеров понятия C , т.е.

$$U^{C(*)} = \{x \mid x \text{ – конечное подмножество } U^C\}, \\ E^{C(*)} = \{x \mid x \text{ – конечное подмножество } E^C\}.$$

• $C(p, q)$, где p и q – натуральные числа и $p \leq q$, обозначает понятие, примерами которого служат конечные множества примеров понятия C , причем размеры этих множеств заключены в пределах чисел p и q .

• $L((p, q)(r, s))$, где $p \leq q$ и $r \leq s$, обозначает бинарную связь, примерами которой служат конечные множества примеров связи L . При этом, если множество X из $L((p, q)(r, s))$ рассматривать как двудольный граф, то степень любой левой (правой) его вершины должна быть заключена между числами p и q (r и s).

• $C_1 \mid C_2 \mid \dots \mid C_n$ обозначает понятие, примерами которого служат примеры всех понятий C_j .

В БМЗ входят следующие языки:

• язык *структурной спецификации* (ЯСС). Предложения ЯСС определяют универсумы понятий;

• язык *спецификации типов данных* (ЯТД). Предложения ЯТД задают абстрактные типы данных, которые используются как домены атрибутов;

• язык *атрибутивных условий* (ЯАУ). Предложения ЯАУ определяют выраженные в

терминах атрибутов ограничения на примеры понятий;

- язык запросов (ЯЗ). Предложениями ЯЗ служат запросы к базам фактов;
- язык логической спецификации (ЯЛС). ЯЛС содержит несколько подязыков.

Онтология в БМЗ – это конечное множество O предложений из ЯСС U ЯТД U ЯЛС. Обозначим через $O/ЯСС$, $O/ЯТД$ и $O/ЯЛС$ части онтологии O , состоящие соответственно из предложений языков ЯСС, ЯТД и ЯЛС. В синтаксически корректной онтологии O предложения из этих частей должны быть согласованы в следующем смысле:

- структура понятий, участвующих в предложении из $O/ЯЛС$, определяется предложениями из $O/ЯСС$;
- атрибуты, содержащиеся в предложении из $O/ЯСС$, имеют домены, которые являются предложениями из $O/ЯТД$.

Предложения ЯСС состояются из примитивных предложений, имеющих следующие формы:

$C[D]$, $C[A:D]$, $C[A:T]$, $(C L D)$, $(C L D)[E]$,
 $(C L D)[A:E]$, $(C L D)[A:T]$,

где C , D , E – имена понятий, L – имя бинарной связи, а T – спецификация типа данных.

Имя D в предложении $C[D]$ является компонентой класса C . Также E в предложении $(C L D)[E]$ является компонентой бинарной связи L . Имя C в первых трех предложениях или выражение $(C L D)$ в остальных четырех предложениях называются *головой* этих предложений. Предложения с одинаковыми головами могут быть соединены в одно предложение. Например, из примитивных предложений $C[D]$, $C[A:E]$ и $C[A:T]$ получается предложение $C[D, A:E, A:T]$.

Приведем пример онтологии в языке ЯСС.

Пример 3.

<‘Учебный процесс’>

1. Студент[ФИО:String, Группа:String].
2. Препо[ФИО:String, Должность:String
Работает_на:Кафедра, Стаж:Integer].
3. Сотрудник[Препо | НаучСотр | Иженер |
Лаборант | Секретарь].
4. Предмет[Назв:String, Зависит_от:Предмет,
КоличЧасов:Integer].
5. Кафедра[Назв:String ЗавКафедрой:Препо,
Состав:Сотрудник(*), Факультет:String].
6. (Студент СдалЭкз Предмет)[Дата:Date,
Оценка: {3,4,5}, Кому:Препо].

</‘Учебный процесс’>

ЗАКЛЮЧЕНИЕ

В настоящей статье даны некоторые уточнения понятия онтологии и связанных с нею понятий. Определение онтологии основано на формальной модели понятия в экстенциональном аспекте. Дано краткое описание системы «Бинарная Модель Знаний», предназначенной для построения онтологий.

Работа выполнена при финансовой поддержке Российского Фонда Фундаментальных Исследований по проекту №11-01-00538.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Antoniou & Hamerlen, 2003] Antoniou G., van Hamerlen F. A Semantic Web Primer. – MIT Press, 2003.
- [Baader et al. 2003] Baader F., Calvanese D., McGuinness D., Nardi D., Patel-Schneider P. (eds.) Description Logic Handbook. – Cambridge University Press, 2003.
- [Bonner & Kifer, 1994] Bonner A.J., Kifer M. An overview of transaction logic // Theoretical Computer Science, 193 (2), 1994.
- [Gruber, 1993] Gruber T.R. A translation approach to portable ontology specifications // Knowledge Acquisition, 5 (2), 1993.
- [Kifer et al., 1995] Kifer M., Lausen G., Wu J. Logical foundation of object-oriented and frame-based languages // Journal of ACM, 42 (4), 1995.
- [Plesniewicz, 2004] Plesniewicz G.S. Binary Data and Knowledge Model // In: Stefanuk, Kaijiri (eds.) Knowledge-based software engineering. – IOS, 2004.
- [Плесневич, 2005] Плесневич Г.С. Бинарные модели знаний // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов III-го Международного научно-практического семинара (Коломна, 15-17 мая 2005 г.). – М.: Физматлит, 2005.

FORMAL ONTOLOGIES

Plesniewicz G.S.

Moscow Power Engineering University, Russia

salve777@mail.ru

Main concepts related to ontologies are presented. Some refined definition of formal ontology is given.

INTRODUCTION

In the paper we give a definition of formal ontologies. The definition is based on the formal model of concept which is extensional in essence. Using this definition we define the notion of formal ontology.

MAIN PART

A formal concept has the following components: (a) the name C ; (b) the universe U^C ; (c) the set of points of reference Γ ; (d) for every $\gamma \in \Gamma$ a subset $E_\gamma^C \subseteq U^C$ – the set of all instances at the point of reference γ ; (e) for every $\gamma \in \Gamma$ an equivalence \sim_γ on E_γ^C – the coreference relation at the point of reference γ ; (f) for every $\gamma \in \Gamma$ the pair $Ext_\gamma^C = (E_\gamma^C, \sim_\gamma)$ – the extension of C at the point of reference γ ; (g) the family $Ext^C = \{Ext_\gamma^C | \gamma \in \Gamma\}$ – the full extension of the concept C .

A concept system is a finite set of formal concepts with the same points of reference. A formal ontology is a specification (by some formal languages) of a given concept system.

Using these definitions we also define main notions related to formal ontologies (interpretations, logical consequence, consistence et al.)

CONCLUSION

In the paper, the refined notion of formal ontologies and related notions are defined.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 007:519.816

ИНТЕГРАЦИЯ АППАРАТА ХРАНИЛИЩ ДАННЫХ И ТЕМПОРАЛЬНЫХ МОДЕЛЕЙ

Еремеев А.П., Еремеев А.А.

*НИУ «Московский энергетический институт»,
г. Москва, Российская Федерация*

eremeev@appmat.ru

YeremeevAA@mpei.ru

Хранилища данных – основное средство для консолидированного представления данных в пределах организации. Время – одна из размерностей, позволяющая сравнивать различные периоды, а темпоральные модели – средство для представления и оперирования временными (темпоральными) зависимостями в данных и знаниях. В работе рассматриваются возможности реализации темпорального хранилища данных, позволяющего оперировать с темпоральными данными и знаниями и передавать их между различными временными версиями, что крайне необходимо для современных интеллектуальных систем типа интеллектуальных динамических систем и систем поддержки принятия решений реального времени. Работа выполнена при финансовой поддержке РФФИ (проект № 11-01-00140).

Ключевые слова: интеллектуальная система, хранилище данных, темпоральная база данных, технология OLAP.

ВВЕДЕНИЕ

Большинство крупных компаний и организаций для долгосрочного анализа, сбора и хранения данных с целью предоставления результирующей информации пользователям в настоящее время используют *хранилища данных* (ХД) [Kimball et al., 2002]. ХД ориентировано на хранение и анализ данных, охватывающих большой временной промежуток. Интеграция различных информационных систем организации – трудоемкое и достаточно длительное действие, особенно, если используются различные программные продукты, которые были разработаны независимо. Чтобы получить полную информацию об организации и использовать перспективные интеллектуальные системы поддержки принятия управленческих решений, необходимо интегрировать данные, полученные от различных источников, учитывая временной фактор, и сделать эти данные доступными для использования.

При помощи технологии ХД может быть выполнена интеграция, в которой оперативные данные водятся и обрабатываются совместно с информацией из классических информационных системам (баз данных), содержащими данные, необходимыми для функционирования

организации. Данные, сохраненные в ХД, предоставляют информацию на протяжении времени (временном интервале), которое необходимо для того, чтобы эффективно управлять (принимать решения) организацией или ее компонентами (объектами). В результате процессов принятия решений данные могут быть возвращены в системы оперативной обработки информации.

Темпоральное ХД можно охарактеризовать как объектно-ориентированный, интегрируемый, поддерживающий временной фактор и долговременный (неразрушающийся) набор данных (совокупность наборов данных), ориентированных на использование в интеллектуальных системах (ИС), в том числе ИС поддержки принятия решений реального времени (ИСППР РВ) [Вагин и др., 2001]. В ИСППР РВ вместе с ХД могут использоваться различные модели и методы для поддержки принятия решений, в том числе и темпоральные модели, ориентированные на представление и оперирование данными и знаниями, содержащими временные (темпоральные) зависимости [Еремеев и др., 2011 а, б]. Для того, чтобы дополнить стандартные средства ХД по подготовке различных отчетов средствами анализа и интерактивного доступа к данным применяются средства (технология) оперативной аналитической обработки данных (On-Line Analytical Processing, OLAP), или

OLAP-технология многомерного анализа данных. Темпоральные ХД часто используются, чтобы интегрировать различные данные, включая темпоральные [Herden, 2002]. Такие ХД часто являются основой для систем управления взаимоотношениями с клиентами (CRM-систем), систем планирования ресурсов предприятия (ERP-систем) и других современных информационных систем. Перспективным использованием технологии ХД является интеграция с OLAP-технологией с применением многомерного представления данных. Средства OLAP-технологии позволяют агрегировать и сравнить данные по различным измерениям (размерностям), относящимся к области определения приложения. К типичным размерностям, содержащимся часто в бизнес-ХД, относятся – время, организационная структура (подразделения, отделы, и т.д.), территориальное нахождение (города, области, страны) и данные управляемого объекта или производимого товара.

Такое многомерное представление обеспечивает наличие долгосрочных данных, которые могут быть проанализированы по оси времени, тогда как большинство ХД только предоставляют снимки данных во времени. Известные OLAP-системы позволяют иметь дело с изменениями значений, например, параметров или характеристик организации (объекта, товара и т.п.), но они не в состоянии оперировать с модификациями в размерностях, например, при новом ответвлении или разделении, несмотря на то, что время обычно явно представляется как размерность в ХД.

Причина этого явления состоит в том, что технология ХД базируется на предположении (ограничении), что размерности являются ортогональными. Ортогональность относительно размерности времени означает, что другие размерности должны быть независимыми от времени. Это ограничение запрещает надлежащую обработку изменений в данных размерности.

Для корректности результатов OLAP-запросов необходимо, чтобы модификации во времени данных (размерности) были приняты во внимание. Например, когда анализируется функционирование некоторой системы и ее компонентов за достаточно длительный период времени, необходимо учитывать, добавление новых или удаление некоторых компонентов за этот период. По данной проблематике в настоящее время проводятся серьезные исследования ввиду ее актуальности.

1. Темпоральные хранилища данных

Для того, чтобы получить корректные результаты OLAP-запросов важно отследить

модификации размерностей данных. В связи с этим необходимо ввести временные расширения в средства ХД. Следовательно, все элементы измерения и все иерархические ссылки между этими элементами измерения должны быть представлены с временными метками из некоторого интервала $[T_s; T_e]$, где T_s и T_e – соответственно начало и конец действительного времени, $T_e > T_s$.

Используя временную проекцию и временную выборку, как это определено в Глоссарии темпоральных баз данных [Dyreson et al., 1994], можно построить базовую структуру темпорального ХД, ориентированную на определенный временной период (интервал). При этом каждая модификация элемента измерения или иерархического отношения приводит к новой версии структуры, если такая версия для данного временного интервала не существует. Такие модификации могут быть сделаны при помощи трех основных операций ХД – INSERT (Добавить), UPDATE (Обновить) и DELETE (Удалить) и некоторых более сложных операций, например, SPLIT (Разделить) и MERGE (Слить).

Рассмотрим возможные подходы к организации запросов к темпоральному ХД. Первый, так называемый *защитный (defensive) подход*, базируется на том, что для каждого сформулированного запроса выполняется проверка - пересекает ли запрос временные границы определенных версий структуры. Если границы между версиями пересекаются, то для такого запроса возможны следующие альтернативы: отклонить запрос или выдать предупреждение.

Однако данный подход не обеспечивает правильность всех возможных запросов. Рассмотрим пример. Пусть отдел А организации с сентября 2011 года разделяется на два подразделения - А1 и А2. Если мы хотим проанализировать все месяцы 2011 года, то для отдела А есть данные за январь-август, а с сентября есть данные только для подразделений А1 и А2. Поэтому специалист по анализу должен иметь достаточное знание предметной области и, в частности, знать, как отдел А и подразделения А1 и А2 связаны между собой.

Следовательно, в темпоральных ХД необходимо иметь механизм для представления зависимостей и отношений между элементами измерения в различных версиях структуры.

Для приведенного примера возможно представить оборот подразделения А1 в течение периодов до сентября 2011 с помощью функции **turnover(A1; period) = 30 % turnover (A; period)**. Также можно показать, что в течение всех периодов с сентября 2011 года численность персонала М#

подразделения A соответствует функции $M\#(A; \text{period}) = M\#(A1; \text{period}) + M\#(A2; \text{period})$. Используя такие функции, можно гарантировать, что успешный анализ может быть сделан даже в тех случаях, когда могут быть изменения в структуре.

Такого типа функции называются *функциями преобразования* и для их определения предназначена операция MapF.

Темпоральная модель ХД позволяет иметь дело с изменениями на уровне экземпляра. При этом используются следующие типы данных [Salzberg et al., 1999]:

- *структурированные данные*, которые содержат необходимую информацию относительно всех структур ХД, т.е. информацию о размерности и временных метках иерархической структуры;
- *отображение структурных данных*, содержащее все функции MapF, т.е. все функции преобразования, которые необходимы для преобразования данных из одной структуры в другую;
- *фактические данные*, представляющие всю фактическую информацию, содержащуюся в ХД.

Используя приведенные типы данных, можно организовать запросы на основе двух видов так называемого *наступательного (offensive) подхода*, который будет рассмотрен далее.

Темпоральное ХД задается как ряд размерностей, например размерности *Время*, *Факты* и *Продукты*. При этом, по сути не различаются размерность и факты, так как факты могут быть определены при помощи размерности *Факты*. Следовательно, рассматриваются факты, которые находятся в иерархическом порядке. Подобно тому, как можно описать размерность *Рынок*, где хранится, например, принадлежность к областям или государствам, можно описать размерность *Факты* с введенным иерархическим порядком, в частности, между *Прибылью*, *Оборотом* и *Затратами*.

Каждая размерность состоит из уровней, например, уровни *День*, *Месяц* и *Год* принадлежат размерности *Время*. Иерархическая структура для этих уровней размерности определена через ряд иерархических назначений (присваиваний) (Hierarchical Level Assignment), например, уровни

измерения в течение размерности *Время* размерности находятся в иерархическом порядке $\text{День} \rightarrow \text{Месяц} \rightarrow \text{Год}$. Экземпляр уровня размерности называют элементом размерности, например, июнь - экземпляр уровня размерности *Месяц*. Эти элементы измерения находятся снова в иерархическом порядке (Hierarchical Member Assignment). Каждая размерность – данные некоторой ячейки, использует ссылку на ряд элементов размерности.

2. Организация запросов к темпоральным хранилищам данных

Для выполнения запроса пользователь сначала должен выбрать определенную (основную) версию структуры. Эта основная версия определяет, какая структура должна использоваться для анализа. В большинстве случаев это будет текущая версия структуры. Однако, в некоторых случаях пользователю необходимо использовать "более старую" (одну из предыдущих) версию структуры. Рассмотрим, например, темпоральное ХД, которое содержит данные обо всех странах Европейского союза (ЕС) с 1988 года [Eder et al, 2001]. В пределах этого темпорального ХД произошли два изменения структуры, связанные с объединением Германии в 1990 г. и вступлением Швеции, Австрии и Финляндии в ЕС в 1995 г. Поэтому можно выделить три версии структуры, которые допустимы для периодов времени, отмеченных в табл. 1, где T_s - начало действительного времени, T_e - конец действительного времени для соответствующей версии структуры.

Таблица 1 – пример версий структур.

Версия структуры	T_s	T_e
SV ₁	1988	1989
SV ₂	1990	1994
SV ₃	1995	∞

Предположим, что пользователь выбирает версию структуры SV₃ в качестве основной версии и запрашивает данные на 1994 г. и 1995 г. для анализа. В этом случае требуются функции преобразования для отображения данных, которые

допустимы и для версии SV2, и для версии SV3. Однако тот же самый анализ мог быть выполнен и с использованием версии структуры SV2 в качестве основной версии. Тогда для выполнения запроса потребуются функции для отображения данных из SV3 в SV2.

Организация запросов должна быть такова, чтобы система не только была способна правильно ответить на запросы, охватывающие многократные периоды и возможно различные версии структуры (данных размерности), но и могла сообщить пользователю, какие структурные модификации имели место. Например, если пользователь утверждает, что запрос "дать число жителей Германии в 1998, 1999 и 1990 гг.", то необходимо сообщить пользователю, что для размерности *Государство* в 1990 г. произошла модификация структуры, а именно - объединение Германии. В зависимости от выбранной основной версии структуры результат запроса может иметь вид, как показано на рис. 1 а) или на рис. 1 б).

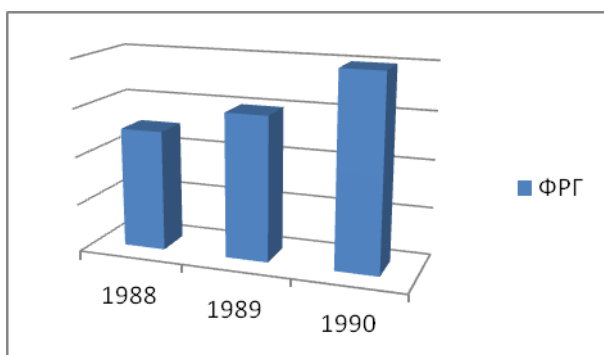


Рисунок 1а) - результат запроса с SV₂ или SV₃

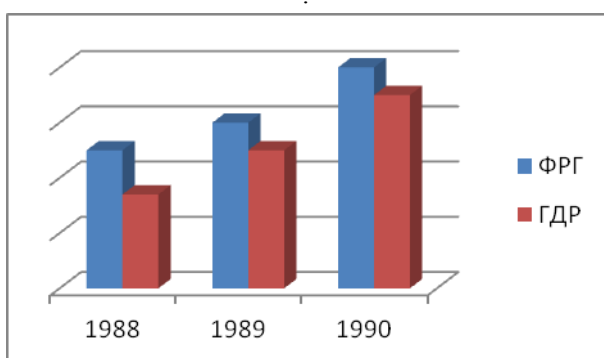


Рисунок 1б) - результат запроса с SV₁

Для решения поставленной задачи по организации запросов возможны два разных

подхода. Оба подхода базируются на следующих общих компонентах:

- *административный инструмент*, позволяющий администратору импортировать новые данные в ячейки темпорального ХД и, конечно, задавать изменения многомерной структуры (в предлагаемом прототипе реализуется с помощью языка программирования Java);
- *темпоральное ХД*, содержащее необходимую информацию о версии структуры, данные и функции преобразования, как было описано ранее (в качестве основы для организации темпорального хранилища данных можно использовать, например, СУБД Oracle);
- *преобразователь (трансформатор)*, который преобразует все требуемые значения ячеек данных из всех используемых версий структуры в выбранную пользователем версию структуры с помощью определенных функций преобразования, то есть, используя необходимые функции MapF (реализуется с помощью языка программирования Java).

Для организации запросов в качестве первого шага используется, так называемый косвенный подход, который проще реализовать, нежели базовый прямой подход, так как при косвенном подходе не требуется реализация интерфейса для исследований.

Основная идея косвенного подхода состоит в том, что преобразователь генерирует одно ХД для каждой версии структуры, необходимой пользователю (рис. 2 а, где DM(Dimension member) - один из элементов списка, который составляет измерение). В большинстве случаев это будет фактической версией структуры. Каждое ХД при этом состоит из всех фактических данных, которые допустимы для того же временного интервала, что и соответствующая версия структуры, а также из всех фактических данных, которые могли быть получены (преобразованы) функциями MapF из других версий структуры. Поэтому пользователь определяет основную версию структуры, выбирая определенное ХД.

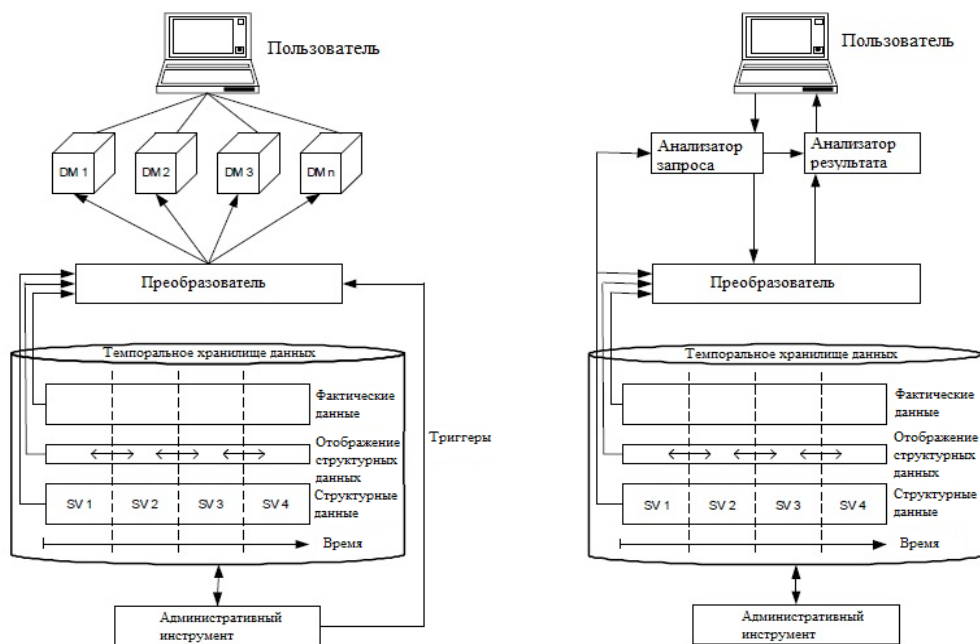


Рисунок 2 - Структура а) косвенного подхода и б) прямого подхода.

Данный подход называется косвенным, потому что преобразователь инициирован административным инструментом, а не пользователем непосредственно. Другими словами - преобразователь начинает генерировать ХД только после того, как была сгенерирована новая версия структуры или были импортированы новые измерения. В обоих случаях преобразователь должен повторно анализировать все существующие ХД, чтобы вернуть непротиворечивую информацию.

Косвенный подход реализован при помощи СУБД Oracle в качестве основы для организации темпорального ХД и Hyperion Essbase в качестве интерфейса, содержащего ХД.

У косвенного подхода имеются и недостатки (ограничения). Основной недостаток заключается в том, что пользователю предоставлен инструмент для организации запросов даже после структурных изменений, однако в этом случае необходимо предварительно сообщить пользователю о том, какие структурные изменения оказали влияние на поставленный запрос. Следовательно, необходимо пополнить результат запроса некоторой дополнительной информацией, как показано на рис. 2 б).

Организация (структура) прямого подхода включает следующие составляющие (см. рис. 2 б):

- анализатор запроса: берет запрос, поставленный пользователем, в качестве входных данных и определяет, какие данные из какой версии структуры необходимы, чтобы ответить на запрос. Результат далее передается в преобразователь и в анализатор результата;
- преобразователь: в отличие от косвенного подхода, преобразователь инициируется пользователем или, другими словами, для каждого установленного запроса преобразователь преобразовывает все необходимые значения ячеек с целью получения ответа на запрос;
- анализатор результата: использует данные, переданные из анализатора запроса и преобразователя, что обогащает результат, полученный преобразователем дополнительной информацией о пользователе, т.е. информацией о том, какие структурные модификации оказали влияние на установленный запрос. В проводимой работе анализатор результата является предметом дальнейшего исследования;
- темпоральное ХД и административный инструмент: эти компоненты описаны ранее.

Можно констатировать, что прямой подход улучшает не только правильность OLAP-запросов после модификации многомерной структуры, но и качество интерпретации ответов на эти запросы.

Заключение

В работе рассмотрены два подхода (структуры) организации темпорального ХД на основе интеграции аппарата ХД и темпоральной модели. Для обоих подходов строится базовое темпоральное ХД, которые отличаются по способу организации запросов и соответственно ответов на запросы. Прямой подход использует анализатор запроса и анализатор результата, чтобы обогатить результат каждого установленного запроса информацией о воздействии структурных модификаций на запрос. Косвенный подход генерирует ХД для определенной версии структуры и преобразовывает данные от всех других версий структуры в это ХД, используя заданную спецификацию функций преобразования. Таким образом, прямой подход предлагает большую гибкость, а косвенный подход предпочтителен по времени отклика. Таким образом, каждый подход имеет свою область предпочтения.

В контексте использования темпоральных ХД (БД) совместно с ИСППР РВ планируется дальнейшее исследование обоих подходов относительно производительности, требований пространства памяти и т.д. Другая важная область исследования связана с построением функций преобразования не только между элементами измерения, но и между размерностями и уровнями измерения. Следует подчеркнуть, что представление функций преобразования на уровне схемы намного сложнее, чем представление на уровне экземпляра темпорального ХД.

Отметим, что рассмотренные подходы улучшают правильность интерпретации ответов на OLAP-запросы и избавляют пользователя от необходимости иметь подробные знания об истории изменения размерности данных.

В настоящее время на кафедре прикладной математики Национального исследовательского университета «МЭИ» ведутся научно-исследовательские работы по созданию базового математического и программного обеспечения для конструирования ИСППР РВ для управления сложными техническими и организационными системами типа объектов энергетики и транспортных систем [Вагин и др., 2001] с использованием нетрадиционных логик, включая темпоральные логики и соответствующие темпоральные БД [Еремеев и др., 2011 а, б].

Библиографический список

- [Kimball et al., 2002] Kimball R., Ross M. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition). John Wiley & Sons. 2002. P. 188 - 198.
- [Вагин и др., 2001] Вагин В.Н., Еремеев А.П. Некоторые базовые принципы построения интеллектуальных систем поддержки принятия решений реального

времени // Известия РАН. Теория и системы управления. 2001. № 6. С. 114-123.

- [Еремеев и др., 2010 а] Еремеев А.А., Еремеев А.П., Пантелеев А.А. Темпоральная модель данных и возможности ее реализации на основе технологии OLAP // Двенадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2010 (20-24 сентября 2010гю, г.Тверь, Россия): Труды конференции. В 4-х томах. Т. 3. - М.: Физматлит, 2010. С. 345-353.
- [Еремеев и др., 2010 б] Еремеев А.П., Еремеев А.А., Пантелеев А.А. Темпоральные базы данных и их применение в интеллектуальных системах // Интеллектуальные системы. Коллективная монография. Выпуск 4/ Под. ред. В.М. Курейчика. - М.: Физматлит, 2010. С. 253-276.
- [Herden, 2002] Herden O. TOLAP: Temporal Online Analytical Processing // International Baltic Conference on Databases and Information Systems 2, 2002. P. 55-66.
- [Dyreson et al., 1994] Dyreson C. E., Grandi F., Snodgrass R. T. et al. A Consensus Glossary of Temporal Database Concepts // *ACM SIGMOD Record*, 23(1), 1994. P. 52-64.
- [Salzberg et al., 1999] Betty Salzberg and Vassilis J.Tsortas. A Comparison of Access Methods for Temporal Data. *ACM Computing Surveys*, 31(2), 1999. P.158-221.
- [Eder et al., 2001] Johann Eder, Christian Koncilia, Tadeusz Morzy, A model for a temporal warehouse // *Proceedings of the Open enterprise solutions: systems, experiences and organizations Workshop (OES-SEO 2001)*, Rome, Italy. P. 48-54
- [Еремеев и др., 2011 а] Еремеев А.П., Еремеев А.А., Пантелеев А.А. Возможности реализации темпоральной базы данных для интеллектуальных систем // Программные продукты и системы. №2(94), 2011. С. 3-7.
- [Еремеев и др., 2011 б] Еремеев А.П., Еремеев А.А., Пантелеев А.А. Применение технологий хранилищ данных и темпоральных баз данных в интеллектуальных системах // *Вестник РГУПС*. №3, 2011. С. 66-72.

INTEGRATION OF THE DATA WAREHOUSES TOOLS AND THE TEMPORAL MODELS

Eremeev A.P., Eremeev A.A.

National Research University 'Moscow Power Engineering Institute', Moscow, Russia

Data warehouses are the main means for consolidated data representation in frames of an organization. The time is one of measurements allowing to compare various periods, and temporal models are the means to present and operate by temporal dependencies in data and knowledge. The possibilities of implementing temporal date warehouses allowing to operate with temporal data and knowledge and transfer them between different temporal versions are viewed. It is extremely important for modern intelligent systems of the type of intelligent dynamic systems and real time decision support ones. The works is performed by financial support of RFBR (the project No 11-01-00140).

Keywords: intelligent system, data warehouse, temporal data base, OLAP-technology



УДК 004.822:514

ИНТЕГРАЦИЯ ЛОГИКИ ДВИЖЕНИЯ И БИНАРНОЙ МОДЕЛИ ДАННЫХ И ЗНАНИЙ
(INTEGRATION OF THE LOGIC OF MOTION AND BINARY DATA AND
KNOWLEDGE MODEL)

Горкина А.А.

*Московский энергетический институт (Национальный исследовательский университет),
г.Москва, Россия***agorkina@mail.ru**

В статье рассмотрены логика движения и бинарная модель знаний. Показано, как можно объединить их в одну систему. Благодаря этой интеграции имеется возможность адекватно описывать в одной модели структуру движущихся объектов и в то же время описывать свойства их движения, а также прогнозировать дальнейшее их поведение.

Ключевые слова: логика движения, модели данных, модели знаний, бинарная модель знаний.

ВВЕДЕНИЕ

В теории интеллектуальных информационных систем имеются многочисленные работы, в которых рассматривается движение объектов во времени и пространстве.

В работе [Yaman et al., 2004] была введена так называемая логика движения, которая соединяет понятия Ньютоновой механики и логики. В дальнейшем эта логика была развита в работах [Yaman et al., 2005a], [Yaman et al., 2005b], [Yaman et al., 2005c] и [Parker et al., 2007].

В логике движений можно рассуждать о планах для движущихся объектах, в частности, предсказывать их дальнейшее расположение. Такие рассуждения важны для многих приложений (управление воздушным трафиком, планирование тактических военных задач и т.п.).

В логике движений объекты не имеют структуры, т.е. фактически являются материальными точками. Однако, во многих задачах требуется представление структуры объектов и необходимо манипулировать этой структурой. Обычно структура объекта определяется его связями с другими объектами, его компонентами и атрибутами. Такими возможностями обладают языки фреймового типа.

В работах [Plesniewicz, 2004] и [Плесневич, 2005] была введена Бинарная Модель Данных и Знаний (БМДЗ), которая представляет собой систему концептуальных (т.е. ориентированных на представление понятий) языков фреймового типа. В

БМДЗ естественным образом можно моделировать структуру объектов, рассуждать о структурированных объектах и вычислять ответы на запросы, относящиеся к данным, представляющих эти объекты.

В этой статье мы кратко опишем способ интеграции логики движений и БМДЗ. Благодаря этой интеграции имеется возможность адекватно описывать в одной модели структуру объектов и, в то же время, описывать свойства движения объектов.

1. Краткие сведения о логике движений

Пусть R – множество всех вещественных чисел, R^+ – множество положительных чисел, O – конечное множество имен объектов, V_R – множество переменных со значениями в R и V_O – множество переменных со значениями в O . *Вещественный терм* – это элемент $R \cup V_R$, *объектный терм* – это элемент $O \cup V_O$.

Атомы логики движений определяются следующим образом:

- Если o_1 и o_2 – объектные термы, d , t_1 , t_2 – положительные вещественные термы, то $\mathit{near}(o_1, o_2, d, t_1, t_2)$ – атом. Интуитивный смысл: «объекты o_1 и o_2 находятся друг от друга на расстоянии d в моменты времени из интервала $[t_1, t_2]$ »;

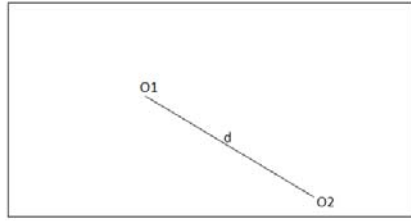


Рисунок 1. Графическая иллюстрация значения near-атома

- Если o_1 и o_2 – объектные термы, d , t_1 , t_2 – положительные вещественные термы, то $\mathbf{far}(o_1, o_2, d, t_1, t_2)$ – атом. Интуитивный смысл: «расстояние между объектами o_1 и o_2 в моменты времени из интервала $[t_1, t_2]$ больше d »;

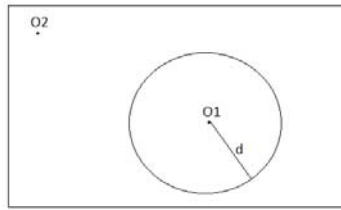


Рисунок 2. Графическая иллюстрация значения far-атома

- Если o – объектный терм, t_1 , t_2 – положительные вещественные термы, x_1, y_1, x_2, y_2 – вещественные термы, то $\mathbf{in}(o, x_1, y_1, x_2, y_2, t_1, t_2)$ – атом. Интуитивный смысл: «объект o находится в некоторый момент времени из интервала $[t_1, t_2]$ в прямоугольнике с левой нижней вершиной (x_1, y_1) и правой верхней вершиной (x_2, y_2) »;

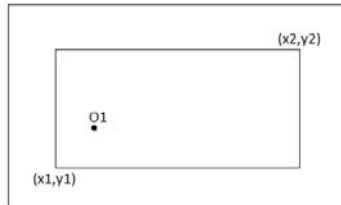


Рисунок 3. Графическая иллюстрация значения in-атома

- Если o – объектный терм, $t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+$ – положительные вещественные термы, x_1, y_1, x_2, y_2 – вещественные термы, то $\mathbf{go}(o, x_1, y_1, x_2, y_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ – атом. Интуитивный смысл: «объект o покидает точку (x_1, y_1) в некоторый момент времени из $[t_1^-, t_1^+]$, движется по прямой с некоторой скоростью из интервала $[v^-, v^+]$ и прибывает в точку (x_2, y_2) в некоторый момент времени из $[t_2^-, t_2^+]$ ».

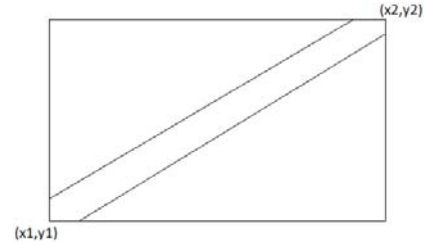


Рисунок 4. Графическая иллюстрация значения go-атома

Предложение логики движения – это атом или булева комбинация атомов, т.е. если α и β – предложения, то предложениями являются $\sim\alpha$, $\alpha\wedge\beta$ и $\alpha\vee\beta$. Выражение $\alpha\rightarrow\beta$ также считается предложением и рассматривается как сокращение предложения $\sim\alpha\vee\beta$.

Семантика логики движений определяется через понятие интерпретации. *Интерпретация* – это непрерывная функция

$I: O \times \mathbb{R}^+ \rightarrow \mathbb{R} \times \mathbb{R}$. Интуитивный смысл: $\langle I(o, t) \rangle$ – положение объекта o в момент времени t .

Атом $\mathbf{go}(o, x_1, y_1, x_2, y_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ *истинен* в интерпретации I *относительно* интервала $[t_1, t_2]$, если:

- $t_1 \in [t_1^-, t_1^+]$, $t_2 \in [t_2^-, t_2^+]$ и $I(o, t_1) = (x_1, y_1)$, $I(o, t_2) = (x_2, y_2)$;
- для всякого момента времени $t \in [t_1, t_2]$ точка $I(o, t)$ лежит на отрезке $[(x_1, y_1), (x_2, y_2)]$;
- для всех $t, t' \in [t_1, t_2]$, если $t < t'$, то имеет место неравенство $\text{dist}(I(o, t), (x_1, y_1)) < \text{dist}(I(o, t'), (x_1, y_1))$, где dist обозначает евклидово расстояние между двумя точками;
- почти всегда в интервале $[t_1, t_2]$, т.е. кроме конечного числа моментов времени t , существует производная $v = d(I(o, t)) / dt$ и $v \in [v_1, v_2]$.

Атомы $\mathbf{near}(o_1, o_2, d, t_1, t_2)$, $\mathbf{far}(o_1, o_2, d, t_1, t_2)$, $\mathbf{in}(o, x_1, y_1, x_2, y_2, t_1, t_2)$ *истинны* в интерпретации I , если соответственно:

- $\text{dist}(I(o_1, t), I(o_2, t)) \leq d$ для всякого $t \in [t_1, t_2]$;
- $\text{dist}(I(o_1, t), I(o_2, t)) > d$ для всякого $t \in [t_1, t_2]$;
- существуют $t \in [t_1, t_2]$, $x \in [x_1, x_2]$, $y \in [y_1, y_2]$ такие, что $I(o_1, t) = (x, y)$.

Атом $\mathbf{go}(o, x_1, y_1, x_2, y_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ *истинен* в интерпретации I , если существует интервал времени $[t_1, t_2]$, что он истинен относительно этого интервала.

Истинность предложений определяется стандартным образом. Если α и β – произвольные предложения, то:

- $I(\sim\alpha) = 1$ (истина) $\Leftrightarrow I(\alpha) = 0$ (ложь);
- $I(\alpha\wedge\beta) = 1 \Leftrightarrow I(\alpha) = 1$ и $I(\beta) = 1$;
- $I(\alpha\vee\beta) = 1 \Leftrightarrow I(\alpha) = 1$ или $I(\beta) = 1$;
- $I(\alpha\rightarrow\beta) = 1 \Leftrightarrow I(\alpha) = 0$ или $I(\beta) = 1$.

Как и во всякой логике, в логике движения имеются понятия логического следствия и логической выполнимости. Пусть E – множество предложений логики движения и α – предложение.

Тогда из E логически следует α (в записи: $E \models \alpha$), если не существует интерпретации, при которой все предложения из E истинны, а предложение α ложно. Множество E выполнимо, если существует интерпретация, при которой все предложения из E истинны.

Проблема логического следствия сводится к проблеме невыполнимости, так как ясно, что $E \models \alpha \Leftrightarrow$ множество $E \cup \{\sim\alpha\}$ невыполнимо. На самом деле, проблему логического следствия (для конечных E) можно свести к проблеме невыполнимости литералов (т.е. атомов или отрицаний атомов). Для этого можно воспользоваться методом аналитических таблиц. Рассмотрим это на примере.

Пример 1.

Пусть $E = \{\sim at_1 \wedge (at_2 \rightarrow at_3), at_2 \rightarrow at_1\}$ и $\alpha = at_3 \vee at_1$. На рисунке 5 показано дерево вывода по методу аналитических таблиц для исходного множества $E \cup \{\sim\alpha\} = \{\sim at_1 \wedge (at_2 \rightarrow at_3), at_2 \rightarrow at_1, \sim(at_3 \vee at_1)\}$. Это дерево имеет три ветви, из которых только одна ветвь открыта. Открытая ветвь содержит множество литералов $F = \{\sim at_1, \sim at_2, \sim at_3\}$. Таким образом, имеет место логическое следствие $E \models \alpha$ тогда и только тогда, когда множество F невыполнимо.

Ф. Яман разработала алгоритмы для выяснения выполнимости конечных множеств атомов логики движения.

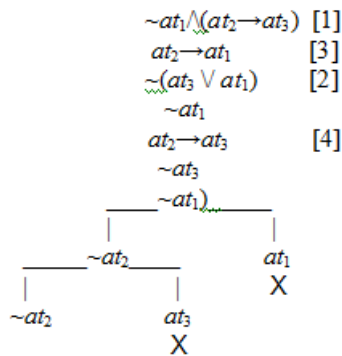


Рисунок 5. Дерево вывода

База знаний в логике движений – это конечное множество предложений этой логики. Запросом назовем выражение вида $?\alpha$, где α – предложение логики движений.

Пусть E – база знаний и $?\alpha$ – запрос к этой базе. Ответом на этот запрос может быть одно из сообщений: «да», «нет» или «нет информации». Алгоритмы, выясняющие логические следствия, можно использовать для вычисления ответов на запросы к базе знаний: ответ на запрос $?\alpha$ есть:

- «да», если $E \models \alpha$;
- «нет», если $E \models \sim\alpha$;
- «нет информации», если $E \not\models \alpha$ и $E \not\models \sim\alpha$.

2. Краткие сведения о бинарной модели данных и знаний

Семантическими примитивами для концептуальных языков служат формальные понятия, смысл которых близок к смыслу понятий класса или типа объектов.

В БМДЗ входит язык структурной спецификации (ЯСС), с помощью которого определяется структура объектов – примеров понятия. В ЯСС можно определять схемы, экземплярами которых служат состояния баз данных.

Мы рассматриваем два вида понятий – классы и бинарные связи (т.е. бинарные отношения между классами). Имена классов бывают простыми или составными. Составные имена классов или бинарных связей обозначают классы или бинарные связи, полученные при помощи соответствующих операций. Используются следующие операции над понятиями:

- $C(*)$ обозначает класс, экземплярами которых служат конечные множества экземпляров понятия C
- $C(p,q)$, где p, q – натуральные числа и $p \leq q$, обозначает понятие, экземплярами которого служат конечные множества экземпляров понятия (класса или бинарной связи) C , причем размеры этих множеств заключены в пределах чисел p и q . Число q можно заменить на символ $*$, который означает неограниченность размера множеств – экземпляров класса. Вместо $C(1,1)$ пишем просто C . Кроме того, вводятся следующие сокращения: $(*)$, $(+)$, (p) , $(\leq p)$, (p) соответственно для $(0,*)$, $(1,*)$, (p,p) , $(0,p)$, $(p,*)$;
- $L((p,q),(r,s))$, где $p \leq q$ и $r \leq s$, обозначает бинарную связь, экземплярами которой служат конечные множества экземпляров L . При этом, если множество $F L((p,q),(r,s))$ рассматривать как двудольный граф, то степень любой левой (правой) вершины должна быть заключена между числами p и q (соответственно, между числами r и s). Для пар (p,q) и (r,s) применяются также вышеупомянутые сокращения;
- $C_1 | C_2 | \dots | C_n$ обозначает класс (бинарную связь), экземплярами которого служат все экземпляры классов (соответственно, бинарных связей) C_j .

Произвольные предложения ЯСС составляется из примитивных предложений, которые имеют следующие формы:

$$C[D], C[A:D], C[A:T], (C L D), (C L D)[E], (C L D)[A:D], (C L D)[A:T],$$

где C, D, E – имена классов, L – имя бинарной связи, A – имя атрибута и T – имя типа данных. Типы данных могут быть примитивными или составными. (Фактически, составной тип данных можно рассматривать как абстрактный тип данных, если для него задать операции.)

Примитивные предложения имеют в БМДЗ естественную денотативную семантику. Например, если декларируется предложение $(C L D)[A:E]$, то в универсум U^L для бинарной связи L включаются

имена вида [C:x,D:y,A:z], где x, y, z – произвольные идентификаторы объектов (суррогаты). (Универсум понятия – это множество всех имен для возможных экземпляров понятия.)

Произвольные предложения ЯСС получаются из примитивных путем их соединения способом, который можно понять на примере.

Пример 2.

Самолет[№Борта:Integer, Тип:{пассаж, груз, воен}, Марка:String, СрокЭкспл:Integer],

Полет[Самолет, АэропортВылета: Аэропорт, Расписание: (Отпр:Time, Прилет:(Time | {?}), ОбъемГорюч(T):(Integer | String), Локация(T):(Real,Real)], T IN (Time | {now, before}), Аэропорт[Назв:String, Индекс:Char(3), Зона:Rectangle], Rectangle = (Ll: (Real,Real), Rt: (Real,Real)).

БМДЗ содержит язык запросов (ЯЗ). Дадим пример простого запроса к схеме примера 2.

Пример 3.

Запрос1: «Найти номера бортов марки Boeing 747, выполняющих полеты из аэропорта Домодедово».

?X.№Борта–Полет(Самолет=X; АэропортВылета = Домодедово; X.Марка = 'Boeing 747').

3. Интеграция логики движения и бинарной модели данных и знаний

При интеграции в предложения ЯЗ включаются атомы логики движений как методы (параметрические атрибуты). Рассмотрим примеры запросов, содержащие обращения к базе данных схемы из примера 2.

Пример 4.

Запрос 2:

- «Найти все пассажирские самолеты марки Boeing 747, находящиеся близко к зоне аэропорта Домодедова и имеющие в настоящий момент мало горючего».

X.№Борта – Полет(Самолет = X;

ОбъемГорюч(now)= мало;

Локация(now); Near Z);

Аэропорт(Назв=Домодедово; Зона= Z).

Запрос 3:

- «Существует ли небезопасный полет в данном прямоугольнике?». Полет считается небезопасным, если самолет эксплуатируются без ремонта более трех лет или имеется другой самолет в том же прямоугольнике.

?X – X IN НебезопПолет;

НебезопПолет= Полет(Самолет.ВремяЭксп > 3

OR X IN Полет(Самолет = Y);

Самолет In (Ll: a, Rt: b); Y In (Ll: c, Rt: d)).

При построении интегрированной системы используются два способа интеграции: интеграция с помощью интерфейсов и логическая интеграция.

3.1. Интеграция с помощью интерфейсов

В нашем случае использование интерфейсов помогает нам соединить модели логики движения и бинарной модели знаний в оптимальном для них виде.

Соответственно, блоки реализации логики движения, а именно задание и проверка на выполнимость атомов go, in, near все так же необходимы, но их внешний интерфейс, как то, что будет показано пользователю, несколько видоизменяется.

Во-первых, каждый из атомов будет внутренне, на уровне бинарной модели знаний, определен с помощью кортежа и линейного конструктора LLIST. Расширенные функции манипуляции с бинарной моделью знаний позволяют так же разбирать полученные кортежи на составляющие, а, значит, и работать с более глубокими знаниями, чем те, которыми изначально позволяет манипулировать логика движения. Для этого будет необходимо использовать деконструкторы кортежей.

В чем преимущества такого подхода? Во-первых, упрощается анализ для различных объектов движения. Во-вторых, на много легче становится добавление размеров материальной точке, которую мы сейчас рассматриваем в наших расчетах. Так же приведение оси к объемному пространству намного упрощается.

3.2. Логическая интеграция.

Логическая интеграция схожа с интерфейсной интеграцией, но требует дополнительного задания связывания доступа к go-атомам через бинарную модель знаний. В частности, необходимо создать конструкторы для сборки каждого атома через бинарную модель. Легче всего это сделать с помощью конструкторов и деконструкторов, указанных в предыдущем способе.

Выводы

В данной работе были проанализированы логика движения, дающая возможность расчета полосы движения в реальном времени, бинарная модель знаний, а так же сделана их интеграция.

Этап интеграции очень важен, так как базовая логика движения без вспомогательных функций несколько громоздка и трудноупотребима. Использование бинарной модели знаний позволяет облегчить как внешнее общение с реализацией логики движения, а так же дальнейший анализ и

расширение логики движения.

При интеграции бинарной модели знаний с логикой движения, упрощается не только графический интерфейс работы с программой, но также возникают преимущества самой получившейся системы перед начальной. Первым преимуществом является то, что упрощается поиск по объектам. Вторым преимуществом является возможность более легкого перевода теории с плоскости на пространство.

Таким образом, интеграция бинарной модели знаний и логики движений – оправдана и полезна

БЛАГОДАРНОСТЬ

Автор благодарит Российский Фонд Фундаментальных Исследований за финансовую поддержку работы, выполненной по проекту №11-01-00538.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Parker et al., 2007] A. Parker, F. Yaman, D. Nau and V.S. Subrahmanian. Probabilistic Go Theories // Proc. 2007 International Joint Conference on Artificial Intelligence (IJCAI-2007), Hyderabad, India, Jan 6-12, 2007.

[Yaman et al., 2004] F. Yaman, D. Nau, V.S. Subrahmanian. The Logic of Motion // Proc. 2004 International Conference on the Principles of Knowledge Representation and Reasoning (KR-2004), Whistler, Canada, June 2-5, 2004.

[Yaman et al., 2005a] F. Yaman, D. Nau and V.S. Subrahmanian. A Motion Closed World Assumption // Proc. 2005 International Joint Conference on Artificial Intelligence (IJCAI-2005), Edinburgh, Scotland, Aug 2-5, 2005.

[Yaman et al., 2005b] F. Yaman, D. Nau, V.S. Subrahmanian. Going Far, Logically // Proc. 2005 International Joint Conference on Artificial Intelligence (IJCAI-2005), Edinburgh, Scotland, Aug 2-5, 2005.

[Yaman et al., 2005c] F. Yaman, D. Nau and V.S. Subrahmanian. A Motion Closed World Assumption // Proc. 2005 International Joint Conference on Artificial Intelligence (IJCAI-2005), Edinburgh, Scotland, Aug 2-5, 2005.

[Plesniewicz, 2004] G.S. Plesniewicz. Binary Data and Knowledge Model // In: Stefanuk, Kajiri (eds.) Knowledge-based software engineering. – IOS, 2004.

[Плесневич, 2005] Г.С. Плесневич. Бинарные модели знаний // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов III-го Международного научно-практического семинара (Коломна, 15-17 мая 2005 г.). – М.: Физматлит, 2005.

INTEGRATION OF THE LOGIC OF MOTION AND BINARY DATA AND KNOWLEDGE MODEL

Gorkina A.A. *

* National Research University "Moscow Power
Engineering Institute", Moscow, Russia

agorkina@mail.ru

The article considers the logic of the movement and the binary model of knowledge. It is shown how to combine them into one system. Through this integration there is a possibility to adequately describe in a model of the structure of moving objects and at the same time describe the properties of their movement, as well as predict future behavior.

INTRODUCTION

In the theory of intelligent information systems, there are a lot of works in that it is discussed the motion of objects in space and time.

In the logic of movements it can be discussed about the plans for moving objects, in particular, to predict their future location. These considerations are important for many applications (air traffic management, planning of tactical military objectives, etc.).

In the logic of motion objects have no structure, that means that in fact are material points. But many tasks require the representation of the object structure and to manipulate with this structure. Typically, the object structure is defined by its relationships with other objects, their components and attributes. These capabilities have languages of framing type.

In works [Plesniewicz, 2004] and [Plesnevich, 2005] was defined Binary Data and Knowledge Model (BDKM), which is a conceptual system (that means, presentation-oriented concepts) languages of framing type. In BDKM naturally can represent the structure of objects and reason about structured objects and calculate answers to queries relating to data representing the objects.

In this article, we briefly describe how to integrate logic of motion and BDKM Through this integration it is possible to describe adequately in a model of the structure of objects, and at the same time, to describe the properties of moving objects.

MAIN PART

This article consists of those parts. The first part gives a general information about the logic of the motion and also describes the basic atoms and the rules of representation of logic of motion. The second part gives general information about the binary model of data and knowledge. It describes the general rules of BDKM and gives examples of using of this theory. The third part describes how to integrate them and examples are considered to illustrate the call to atoms after the integration.

The logic of motion consists of three main and one additional atoms. They are defined as follows:

- If $O1$ and $O2$ are object terms, d , $t1$, $t2$ are positive real terms, then near ($O1$, $O2$, d , $t1$, $t2$) is atom. The intuitive meaning: "the objects $O1$ and $O2$ are separated by a distance d at time in interval of $[t1, t2]$ ";
- If $O1$ and $O2$ are object terms, d , $t1$, $t2$ are positive real terms, then far ($a1$, $a2$, d , $t1$, $t2$) is atom. The intuitive meaning: "the distance between objects $O1$ and $O2$ at time in the interval $[t1, t2]$ more than d ";
- If a is object term, $t1$, $t2$ are positive real terms, $x1$, $y1$, $x2$, $y2$ are real terms, then in (a , $x1$, $y1$, $x2$, $y2$, $t1$, $t2$) is atom. The intuitive meaning: "the object is at some time in the interval $[t1, t2]$ in the rectangle with lower left vertex ($x1$, $y1$) and upper right vertex ($x2$, $y2$);
- If o is object term, $t1-$, $t1+$, $t2-$, $t2+$, $v-$, $v+$ are positive real terms, $x1$, $y1$, $x2$, $y2$ are real terms, then

Semantic primitives for conceptual languages are formal concepts whose meaning is close to the meaning of the concepts of a class or object type.

In BDKM includes structural specification language (LSN), used to define the structure of objects such as examples of the concept. In LSN can be defined the schema instances which serve as the state databases.

We consider two kinds of concepts: classes and binary relation (that means, binary relations between classes). Class names can be simple or compound. Compound names of the classes or binary relations represent classes or binary communications received by means of the transactions. The following operations are on concepts:

- $C(*)$ means class whose instances are finite sets of instances of concept C
- $C(p,q)$, where p, q are natural numbers and $p \leq q$, means concept whose instances are finite sets of instances of the concept (a class or binary relation) C , and the sizes of these sets are in the range of numbers p and q . The number q can be replaced by the symbol $*$, which means unlimited size of sets of instances of the class. Together $C(1,1)$ we write simply C . In addition, we introduce the following abbreviations: $(*)$ $(+)$ (p) , $(\leq p)$, (p) , respectively, for $(0,*)$, $(1,*)$, (p,p) , $(0,p)$, $(p,*)$;
- $L((p,q),(r,s))$, where $p \leq q$ and $r \leq s$ means a binary relation, instances of which are finite sets of instances L . In this case, if the set $FL((p, q), (r, s))$ viewed as a bipartite graph, then the degree of each left (right) vertex must be enclosed between the numbers p and q (respectively, between numbers r and s). For pairs (p, q) and (r, s) shall be also applied aforementioned reductions;
- $C1 | C2 | \dots | Cn$ means class (binary relation), instances of which are all instances of classes (respectively, binary relations) Cj .

Arbitrary proposals LSN compiled from primitive proposals, which have the following forms:

$C[D]$, $C[A:D]$, $C[A:T]$, $(C \ L \ D)$, $(C \ L \ D)[E]$, $(C \ L \ D)[A:D]$, $(C \ L \ D)[A:T]$,

where C, D, E are class names, L is the name of a binary relation, A is attribute name and the T is the data type name. Data types can be primitive or composite. (In fact, a composite data type can be considered as an abstract data type, if we can set him the set of operations.)

Primitive sentences in BMDZ have natural denotative semantics. For example, if the offer is declared (with LD) $[A: E]$, then the universe of UL for a binary relation L includes the names of the form $[C: x, D: y, A: z]$, where x, y, z are arbitrary object identifiers

(substitutes). (The universe of concepts is the set of names for possible instances of the concept.)

Arbitrary proposals can be obtained from LSN primitive by their connection.

During the integration proposals of LSN include atoms of logic of motion as methods (parametric attributes).

There are two approaches described above, the integration of models at the logical level and using the interface approach. In this work presents both methods.

Integration with Interfaces as basic

In our case, the use of interfaces helps us to connect the logic of model of and the binary model of the movement of data and knowledge in an optimal form for them. Accordingly, a block of logic of motion, namely the task and check on the feasibility of the atoms go, in, near and far still necessary but their external interface is multiplicity modified from the case without integration.

At first, each atom will be internally at the level of the binary model of data and knowledge, is defined by the tuple and constructor of linear lists LLIST. Advanced functions manipulate binary data model and knowledge allows also to disassemble obtained tuples into components and, therefore, to work with deeper data and knowledge than that which was originally to manipulate the logic of the motion. This would involve to use for deconstructors of tuples.

The advantages of this approach are:

First of all, it simplifies the analysis of motion of various objects.

Second, it becomes much easier to add structure to the material points.

Logical integration

Logical integration is similar to the interface integration, but requires an additional set of binding to access go-atoms via binary model of data and knowledge. In particular, it is necessary to create constructor for the assembly of each atom in the binary model. The easiest way to do it with constructors and deconstructors mentioned in the previous method.

CONCLUSION

In this work were analyzed logic of the motion giving an opportunity to calculate the traffic lanes in real-time, binary model of knowledge, as well as their integration was done.

Integration phase is very important, because the basic logic of the motion without auxiliary functions is several cumbersome and difficult to implement. The use of binary model of data and knowledge can facilitate the external communication with the implementation of logic of motion, as well as further analysis and extension of logic of the motion.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

ПОСТРОЕНИЕ МНОГОЯЗЫЧНЫХ ТЕЗАУРУСОВ СРЕДСТВАМИ СЕМАНТИЧЕСКОЙ ТЕХНОЛОГИИ

Загорулько Ю.А.^{*,**}, Боровикова О.И.^{*}, Загорулько Г.Б.^{*,**}

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

*** Новосибирский Государственный Университет, г. Новосибирск, Россия*

zagor@iis.nsk.su

olesya@iis.nsk.su

gal@iis.nsk.su

В докладе представлен подход к разработке многоязычного электронного тезауруса для произвольной предметной области, особенностью которого является использование в качестве инструмента разработки формальных и программных средств, предоставляемых семантической технологией, разработанной для построения порталов научных знаний. Благодаря тому, что эта технология базируется на онтологии, обеспечивается не только возможность расширения, целостность и непротиворечивость терминологической системы тезауруса, но и удобный доступ к его контенту.

Ключевые слова: многоязычный тезаурус, концептуальная схема тезауруса, онтология, технология построения порталов научных знаний.

ВВЕДЕНИЕ

В связи с глобализацией и интеграцией научного знания часто возникают потребности в разработке многоязычных тезаурусов для различных предметных областей. Однако проблема состоит в том, что на данный момент отсутствуют доступные, гибкие, простые в использовании, но в то же время достаточно мощные средства разработки таких тезаурусов.

При этом под доступными мы понимаем свободно распространяемые или недорогие средства разработки многоязычных тезаурусов, обязательно включающие поддержку разработки русскоязычной версии. Под простыми в использовании – средства разработки, которыми могли бы легко воспользоваться эксперты в предметной области без помощи IT-специалистов и дополнительного обучения. Гибкими мы считаем средства, которые позволяют легко настраивать тезаурус на требуемую предметную область, т.е. в любой момент вводить дополнительные свойства в описания терминов и расширять множество отношений между ними. Важным требованием к таким средствам является поддержание логической целостности терминологической системы тезауруса.

В качестве инструмента построения многоязычного тезауруса, удовлетворяющего всем

описанным выше свойствам, в данной работе предлагается использовать методологию и программные компоненты семантической технологии, разработанной для построения порталов научных знаний [Загорулько и др., 2008; Загорулько и др., 2009], которая ранее была применена для создания порталов знаний по археологии [Андреева и др., 2006] и компьютерной лингвистике [Боровикова и др., 2008].

Данная технология базируется на онтологии и предоставляет средства настройки на предметную область и управления контентом информационной системы, а также средства навигации и поиска. Средства настройки на предметную область и поддерживаемая ими методология достаточно хорошо подходят для разработки концептуальной схемы тезауруса, а остальные из перечисленных средств могут выполнять роль его основных программных компонентов, обеспечивающих создание, сопровождение и использование тезауруса.

1. Требования к многоязычному тезаурусу

Перед рассмотрением требований, предъявляемых к многоязычному тезаурусу, определим, какого рода информационный ресурс мы понимаем под тезаурусом.

В современной лингвистике принято следующее определение тезауруса: «Тезаурус (от греч. *сокровище*) — особая разновидность словарей общей или специальной лексики, в которых указаны семантические отношения (синонимы, антонимы, паронимы, гипонимы, гиперонимы и т. п.) между лексическими единицами» [Википедия, 2011]. Из этого определения следует, что главным отличием тезауруса от словарей, в том числе толковых, состоит в том, что в тезаурусе смысл термина представляется главным образом посредством соотношения его с другими терминами путем установления между ним и этими терминами семантических отношений.

Таким образом, все термины тезауруса оказываются связанными в одну семантическую сеть, представляющую систему знаний о некоторой предметной области (ПрО). Наличие у тезауруса сетевой структуры создает предпосылки для его использования в задачах индексирования и информационного поиска [Лукашевич, 2011].

Для того, чтобы тезаурусом было легко пользоваться и он представлял собой более полную картину моделируемой ПрО, в него включаются определения наиболее важных терминов.

В настоящее время наиболее полно разработаны требования к построению информационно-поисковых тезаурусов (ИПТ), т.е. тезаурусов, ориентированных на индексирование и информационный поиск документов. Эти требования представлены в соответствующих отечественных и международных стандартах [ISO 5964-1985; ГОСТ 7.24-2007; ISO 2788-1986; ГОСТ 7.25-2001; ANSI/NISO Z39.19-2005], которые определяют основные единицы, которые могут включаться в тезаурус, и набор отношений между ними, устанавливают правила сбора массива лексических единиц, формирования словника, построения словарных статей и оформления ИПТ.

По своему составу ИПТ подразделяют на тезаурусы, все единицы которых являются дескрипторами (или предпочтительными терминами), и тезаурусы, включающие как дескрипторы, так и аскрипторы (обычные термины). При этом дескрипторы могут использоваться при индексировании документов и в поисковых запросах, а аскрипторы (как текстовые входы) подлежат замене одним или несколькими дескрипторами.

В зависимости от языковой направленности тезаурусы разделяются на одноязычные и многоязычные. Многоязычный ИПТ содержит термины на нескольких естественных языках и представляет эквивалентные по смыслу понятия на каждом из них.

Согласно стандартам, в словарную статью многоязычного тезауруса входят такие атрибуты, как название, язык термина, релятор (помета, служащая для различения омонимичных терминов),

комментарий. Термины тезауруса связываются различными семантическими отношениями, отражающими их место в системе понятий выбранной ПрО. Отношения могут снабжаться набором свойств (в том числе математических), отражающих их семантику в тезаурусе.

Для подтверждения актуальности терминов и ознакомления пользователей тезауруса с практикой их употребления в тезаурус могут также включаться описания источников терминов.

В тезаурусах, разрабатываемых для сложных предметных областей (областей знаний), термины могут соотноситься с подобластями знаний.

2. Концептуальная схема тезауруса

На основе анализа требований, изложенных в разделе 1, была предложена структура тезауруса, основными единицами которого являются термины предметной области, связанные между собой семантическими отношениями, а также описания областей знаний и источников, т.е. текстовых документов или коллекций текстовых документов, в которых термины тезауруса встречаются или определяются. При этом термины подразделяются на дескрипторы и аскрипторы.

Так как предлагаемый в данной работе подход нацелен на создание тезаурусов двойного назначения, т.е. тезаурусов, ориентированных как на решение задач индексирования и информационного поиска, так и на непосредственное использование людьми, желающими обратиться к системе понятий данной предметной области, тезаурусы включают также и определения наиболее важных терминов (дескрипторов).

Как было сказано выше, в качестве инструмента построения многоязычного тезауруса предлагается использовать методологию и программные компоненты семантической технологии, разработанной для построения порталов научных знаний. Так как данная технология базируется на онтологии, то первым шагом построения тезауруса в ее рамках является представление концептуальной схемы тезауруса в виде онтологии, которую мы в дальнейшем будем называть онтологией представления тезауруса. Эта онтология будет не только определять структуры для информационного наполнения (контента) тезауруса, но и служить базисом для организации содержательного доступа к содержащимся в нем знаниям и данным.

Для описания онтологии данная технология предоставляет формализм и поддерживающий его редактор онтологии. С помощью этих средств была построена онтология представления тезауруса O_{Th} , задающая его концептуальную схему:

$$O_{Th} = \langle C, R, T, D, At, P, Axt \rangle,$$

где $C = \{Tr, S_T, S_K\}$ – конечное непустое множество классов, представляющих основные

сущности тезауруса; здесь $Tr = Asc \cup Des$ – класс терминов, представляющих понятия ПрО «Компьютерная лингвистика», включающий два подкласса – Asc (термины-аскрипторы) и Des (термины-дескрипторы); S_T – класс источников терминов; S_K – класс областей/подобластей знаний;

$R = R^{TT} \cup R^{TST} \cup R^{TSK}$ – конечное множество отношений, где

$R^{TT} = \{R_1^{TT}, \dots, R_m^{TT}\}, R_i^{TT} \subseteq Tr \times Tr$ – конечное множество бинарных отношений, заданных на терминах,

$R^{TST} = \{R^{TSF}, R^{TSP}, R^{TSD}\}, R_i^{TST} \subseteq Tr \times S_T$ – бинарные отношения, связывающие термины тезауруса с источниками, причем R^{TSF} связывает термин с источником, где он встречается, R^{TSP} связывает термин с источником, где он встречается в предметном указателе или глоссарии, а R^{TSD} – связывает термин с источником, где дается его определение;

$R^{TSK} = \{R^{SKT}, R^{SKS}\}$ – бинарные отношения, служащие для встраивания областей знаний в тезаурус, где $R^{SKT} \subseteq Tr \times S_K$ связывает термины тезауруса с областями знаний, а $R^{SKS} \subseteq S_K \times S_K$ – задает иерархию на подобластях знаний;

T – множество стандартных типов;

$D = \{d_1, \dots, d_n\}$ – множество доменов $d_i = \{s_1, \dots, s_k\}$, где s_i – значение стандартного типа string;

$At = \{at_1, \dots, at_w\}$ – конечное множество атрибутов, описывающих свойства основных сущностей тезауруса и отношений между ними; значения этих свойств определены на множестве $T \cup D$;

$P = \{P_1, \dots, P_n\}$ – множество формальных (математических) свойств отношений R^{TT} ;

Axt – множество аксиом, задающих дополнительные ограничения на связи между терминами.

Таким образом, онтология представления тезауруса описывает классы, представляющие основные сущности тезауруса (термины тезауруса, их источники, области/подобласти знаний), отношения, связывающие объекты этих классов между собой, свойства понятий и отношений, а также аксиомы, определяющие их дополнительную семантику. Кроме того, в онтологии задается множество доменов, т.е. возможных значений

атрибутов классов и отношений, что позволяет уменьшить число ошибок при создании/редактировании конкретного тезауруса.

Рассмотрим подробнее классы онтологии, представляющие основные сущности тезауруса.

Класс Tr («Термин») включает следующие атрибуты: *название термина, язык, комментарий, автор статьи*. Заметим, что название термина может задаваться словом, словосочетанием или лексически значимым компонентом сложного слова естественного языка. Автор статьи задается для контроля процесса коллективной разработки тезауруса и в финальную версию тезауруса может не включаться.

Классы Des («Дескриптор») и Asc («Аскриптор»), являясь подклассами класса Tr , наследуют перечисленные выше его атрибуты. Кроме того, класс Des включает следующие дополнительные атрибуты: *релятор, определение термина, признак корневого термина*.

Определение термина поясняет на языке термина его смысл или значение. Заметим, что наличие в тезаурусе определений терминов делает возможным его использование не только в качестве инструмента для ручного или автоматизированного индексирования, но и в качестве источника систематизированных знаний о данной ПрО.

Признак корневого термина указывает на то, что дескриптор находится на самом верхнем уровне одной из иерархии понятий.

Класс S_T («Источник терминов») описывается следующими атрибутами: *название, библиографическая ссылка, язык, тип* (книга, монография, научная статья, документация, учебник, словарь, тезаурус, интернет-ресурс, коллекция текстов и др.), *краткое описание и адрес в сети Интернет*. Для коллекции текстов дополнительно может быть задано количество содержащихся в ней текстов и словоупотреблений.

Класс S_K , предназначенный для описания областей/подобластей знаний, включает такие атрибуты как *название и описание подобласти*.

В онтологии представлено три типа отношений – отношения между терминами (R^{TT}), отношения между терминами и источниками (R^{TST}) и отношения, служащие для встраивания областей знаний в тезаурус (R^{TSK}).

Главную роль в тезаурусе играют отношения между терминами. Именно они, определяя место каждого термина в системе понятий тезауруса, задают его смысл.

Для того чтобы тезаурус был полезным информационным ресурсом, он должен представлять целостную и непротиворечивую систему понятий ПрО. В единую систему понятия

связываются с помощью семантических отношений, поэтому непротиворечивость системы понятий может быть обеспечена заданием ограничений на связи, устанавливаемые между терминами тезауруса. Такие ограничения могут быть заданы специальными аксиомами, а также путем приписывания отношениям структурных и математических свойств.

Структурные свойства отношений определяют класс их аргументов и тип атрибутов (при их наличии). Для любого отношения из R^{TT} может быть задано одно или несколько непротиворечивых математических свойств из следующего набора: симметричность (*symmetry*), рефлексивность (*reflexivity*), транзитивность (*transitivity*), антирефлексивность (*antireflexivity*), асимметричность (*asymmetry*).

В контексте тезауруса семантика этих свойств описывается следующими аксиомами:

$$\forall x, y \in Tr, x \neq y, R \in R^{TT}, \text{symmetry}(R): \\ R(x, y) \rightarrow R(y, x).$$

$$\forall x, y \in Tr, x \neq y, R \in R^{TT}, \text{asymmetry}(R): \\ R(x, y) \rightarrow \neg R(y, x).$$

$$\forall x, y, z \in Tr, R \in R^{TT}, \text{transitivity}(R): \\ (R(x, y) \& R(y, z)) \rightarrow R(x, z).$$

$$\forall x, y \in Tr, x \neq y, R \in R^{TT}, \text{reflexivity}(R): \\ R(x, y) \rightarrow (R(x, x) \& R(y, y)).$$

$$\forall x, y \in Tr, x \neq y, R \in R^{TT}, \text{antireflexivity}(R): \\ R(x, y) \rightarrow (\neg R(x, x) \& \neg R(y, y)).$$

Кроме того, для любого отношения из R^{TT} может быть задано обратное (инверсное) отношение. Семантика такого свойства описывается следующей аксиомой:

$$\forall x, y \in Tr, x \neq y, R \in R^T, \text{invertibility}(R): \\ R, R^{-1} \in R^T : R(x, y) \rightarrow R^{-1}(y, x).$$

В соответствии с описанными выше аксиомами: если отношение R обладает свойством симметричности, то для любых терминов x и y наличие связи $R(x, y)$ влечет существование связи $R(y, x)$; если отношение R обладает свойством асимметричности, то при наличии связи $R(x, y)$ между любой парой терминов x и y запрещается существование связи $R(y, x)$; наличие свойства антирефлексивности для $R(x, y)$ накладывает запрет на существование связей $R(x, x)$ и $R(y, y)$; если отношения R_1 и R_2 являются обратными друг к другу, то для любых терминов x и y появление

связи $R_1(x, y)$ влечет существование обратной связи $R_2(y, x)$.

Кроме того, для терминов, принадлежащих определенному классу (подмножеству терминов), например, дескрипторам (*Des*) или аскрипторам (*Asc*), могут задаваться ограничения на существование или количество каких-либо связей между терминами (x) этого класса и терминами (y, z) этого или других классов терминов. Такие ограничения могут описываться следующими аксиомами:

$$\forall x, y, z \in Des, Des \subseteq Tr, y \neq z, R \in R^{TT}, \\ \text{iniqueness}(R) : R(x, y) \rightarrow \neg R(x, z).$$

$$\forall x \in Asc, y_i \in Des, R \in R^{TT}, \\ \text{obligatoriness}(R) : \exists^N R(x, y_i), i = 1, \dots, N.$$

Первая аксиома задает единственность (*uniqueness*) связи R для некоторого класса терминов (например, *Des*), т.е. устанавливает, что при существовании у термина x этого класса связи $R(x, y)$ невозможно добавление для него еще одной связи R с другим термином. Вторая аксиома устанавливает обязательность (*obligatoriness*) связи R для некоторого класса терминов; тем самым она контролирует наличие N связей вида $R(x, y)$ для термина x этого класса.

Для отражения семантических связей между понятиями, выражаемыми дескрипторами, устанавливаются иерархические, ассоциативные отношения, а также отношения эквивалентности. Заметим, что если первые два вида отношений устанавливаются между одноязычными дескрипторами, то последнее – между разноязычными.

Между дескрипторами вводятся следующие иерархические отношения:

- Недифференцированная иерархическая связь «Выше», направленная от нижестоящего дескриптора к вышестоящему. Для этого отношения задается обратное отношение «Ниже» и свойство транзитивности.
- Родовидовая связь «ВышеРод», устанавливаемая между двумя дескрипторами, когда объем понятия нижестоящего дескриптора входит в объем понятия вышестоящего дескриптора. Это отношение имеет обратное отношение «НижеВид» и обладает свойством транзитивности.
- «ВышеКласс» служит для задания связи между дескрипторами, представляющими класс понятий и экземпляр этого класса. Данное отношение является нетранзитивным и асимметричным и имеет обратное отношение «ЭкземплярКласса».

- Партонимическая связь «ВышеЦелое», задаваемая между двумя дескрипторами в том случае, когда нижестоящий дескриптор представляет компонент объекта, обозначаемого вышестоящим дескриптором. Для этого отношения задается обратное отношение «НижеЧасть» и свойство транзитивности.

Так как один и тот же дескриптор может одновременно входить в несколько иерархий, построенных не только по различным отношениям, но и по различным основаниям классификации, вводится дополнительный признак «Аспект деления иерархии». Этот признак (в качестве значения атрибута соответствующего отношения) приписывается связям, организующим какое-либо множество дескрипторов в иерархию по одному и тому же отношению и основанию классификации. Тот факт, что какой-либо дескриптор находится в самом верху иерархии, отражается установлением его атрибута *признак корневого термина* в значение *Истина*.

Связи между дескрипторами, отличные от иерархических отношений и отношений синонимии, задаются отношением «Ассоциируется с». Это отношение позволяет задавать произвольные ассоциативные связи между дескрипторами, например, отношения, выражающие зависимости вида «процесс-объект», «причина-следствие» и др. Следует заметить, что если в тезаурусе требуется отразить более богатый набор отношений, специфичных для его ПрО, то такие отношения могут быть введены в онтологию представления тезауруса вместо отношения «Ассоциируется с».

Между одноязычными дескрипторами и аскрипторами устанавливаются отношения синонимии.

Если дескриптор может однозначно во всех контекстах заменить некоторый аскриптор, то он связывается с ним отношением «Синоним». Для этого отношения вводится обратное к нему отношение «Смотри».

Если нет однозначного соответствия между дескрипторами и аскрипторами, то используются следующие отношения:

«Используй альтернативно» – устанавливает связь между аскриптором и множеством альтернативных дескрипторов; обратное ему отношение – «Сравни альтернативный выбор»;

«Используй комбинацию» – позволяет представлять аскриптор комбинацией дескрипторов; обратное ему отношение – «Сравни комбинацию».

Чтобы указать эквивалентность дескрипторов из разных одноязычных версий между ними устанавливается симметричное отношение «Эквивалент на другом языке».

Отношение R^{SKT} («Подобласть знаний») связывает термины тезауруса с областями знаний.

Отношение R^{SKS} («Включает») служит для задания иерархий на подобластях знаний.

В онтологии для связи термина с источниками введены три отношения R^{TSF} («Встречается в»), R^{TSP} («Встречается в части документа») и R^{TSD} («Дается определение в»).

Отношение «Встречается в» служит для связывания любого термина с источником; при этом, если источник – коллекция текстов, то в качестве значения специального атрибута этого отношения можно указать частоту встречаемости термина в источнике.

С помощью отношения «Встречается в части документа» можно сообщить, что данный термин встречается в предметном указателе или глоссарии источника, что указывает на важность термина и повышает степень доверия к нему.

С помощью отношения «Дается определение в» термины-дескрипторы, снабженные толкованиями-определениями, связываются с источниками определений.

3. Создание и сопровождение тезауруса

Для задания конкретных терминов, их определений и источников, а также для установления связей между ними используется редактор данных, предоставляемый технологией построения порталов знаний и работающий под управлением онтологии представления тезауруса. Этот редактор обеспечивает экспертов-лингвистов удобным web-интерфейсом для ведения тезауруса. После ввода или редактирования терминов, источников и связей между ними, новая информация становится сразу доступной пользователям тезауруса через пользовательский web-интерфейс.

С целью обеспечения распределенной коллективной разработки тезаурусов используемая технология поддерживает механизм делегирования прав экспертам разных уровней. В соответствии с этим механизмом только эксперты самого высокого уровня могут редактировать структуры тезауруса (с помощью редактора онтологий), а эксперты более низких уровней – только его содержание (с помощью редактора данных).

Кроме того, действует правило, согласно которому редактировать словарную статью может только ее автор. Если кто-то из экспертов захочет внести изменения в «чужую» статью, он должен согласовать свои действия с ее автором.

Следует заметить, что управление контентом тезауруса значительно упрощается благодаря тому, что логическая целостность и непротиворечивость системы понятий тезауруса обеспечивается встроенными в редактор данных специальными механизмами контроля и вывода знаний, работа

которых базируется на аксиомах, описывающих свойства отношений и классов терминов.

В частности, механизмы контроля и вывода знаний контролируют такие свойства отношений, как транзитивность, симметричность, асимметричность, рефлексивность, антирефлексивность, обратимость (наличие обратных отношений), а также ограничения на существование (количество) и обязательность связей. На основе аксиом происходит корректное установление связей между терминами тезауруса, при необходимости осуществляется автоматическое добавление и удаление таких связей.

Например, если рассмотренное выше отношение «Смотри» обладает свойством обратимости и имеет ограничение на число связей (разрешается одна связь), то при связывании аскриптора x и дескриптора y отношением $Смотри(x,y)$ дополнительно будет создана обратная связь $Синоним(y,x)$ (если таковой не существовало), а также будет контролироваться запрет на создание связей $Смотри(x,z)$ и $Синоним(z,x)$ с другим дескриптором z .

4. Обеспечение доступа к терминам тезауруса

Доступ к терминам тезауруса и другим его сущностям обеспечивается пользовательским web-интерфейсом, также предоставляемым технологией построения порталов знаний. В этом интерфейсе содержимое тезауруса представляется пользователю в виде сети взаимосвязанных информационных объектов, представляющих термины, описания под областей знаний, а также описания источников терминов и их определений.

При навигации по тезаурусу обеспечивается возможность выбора необходимых пользователю терминов, детального просмотра их описаний (тезаурусных статей), а также источников (публикаций или коллекций текстов), в которых встречается термин и/или его определение.

Пользователь может указать, какой тип информации его интересует – все термины, дескрипторы, аскрипторы или источники терминов. При этом ему выдается упорядоченный по алфавиту полный список имеющихся в тезаурусе объектов выбранного класса, который отображается в виде html-страницы, содержащей набор ссылок на эти объекты. Дальнейшая навигация по тезаурусу представляет собой процесс перехода от одних элементов тезауруса к другим по заданным между ними связям, отражающим существующие между ними – тезаурусные (между терминами) или библиографические (между терминами и источниками) – отношения.

Например, при просмотре информации о термине-дескрипторе «Семантическое отношение» мы можем видеть значения его атрибутов, а также его связи с другими терминами и источниками

терминов (см. Рисунок 1). В частности, мы можем просмотреть определение термина, узнать имя разработчика его тезаурусной статьи. Используя представленные связи термина в качестве элементов навигации, мы можем перейти к просмотру ассоциирующихся с ним терминов, его эквивалента на английском языке, понятий, стоящих выше и ниже его в иерархии, соотношенным с ним под областями знаний, а также описаний источников, в которых он встречается и определяется.

Дескриптор	
название	семантическое отношение
язык	русский
релятор	
определение 1	отношение, определяющее смысловые взаимосвязи между наименованиями понятий
автор словарной статьи	Логинава Е.

Связи объекта	
Ассоциируется с (RelatedTerm)	
Дескриптор	
лексическая функция	
Встречается дескриптор в (SourceDescriptor)	
Источник	частота
Коллекция текстов Диалог 2000-2010	248
Выше (BT)	
Дескриптор	
семантическое поле	
Дается определение в (SourceDef)	
Источник	определение
Интернет энциклопедия «Википедия»	1
Ниже (NT)	
Дескриптор	
антонимия	
гиперонимия	
гипонимия	
конверсивность	
омонимия	
(Всего: 6)	
Подобласть знаний (SubArea)	
Подобласть знаний	
3.1. Компьютерная лексикология и лексикография - Computational lexicology and lexicography	
3.2. Онтологии и тезаурусы - Ontologies and thesauri	
CO1.3. Синтаксис и лексическая семантика - Syntax and Lexical semantics	
Эквивалент на другом языке (Trans)	
Дескриптор	
semantic relation	

Рисунок 1 – Тезаурусная статья дескриптора «Семантическое отношение»

Для обеспечения доступа к терминам тезауруса из внешних систем разработан программный интерфейс, благодаря которому тезаурус может использоваться при решении задач индексирования и поиска текстовых документов, относящихся к моделируемой тезаурусом предметной области.

ЗАКЛЮЧЕНИЕ

В докладе представлен подход к разработке многоязычных электронных тезаурусов, общий состав и структура которых удовлетворяют международным и отечественным стандартам. Особенностью подхода является использование в качестве инструмента разработки ранее созданных формальных и программных средств, предоставляемых технологией построения порталов научных знаний. Благодаря тому, что эта технология базируется на онтологии, с помощью которой описывается концептуальная схема данных тезауруса (онтология представления тезауруса),

обеспечивается не только возможность расширения, целостность и непротиворечивость терминологической системы тезауруса, но и удобный доступ к его контенту.

Предлагаемый подход в настоящее время используется для создания русско-английского тезауруса по компьютерной лингвистике [Загорулько и др., 2011; Соколова и др., 2011]. Этот тезаурус разрабатывается как набор одноязычных версий многоязычных ИПТ, при этом выполняется согласованное построение одновременно двух версий тезауруса – русскоязычной и англоязычной.

Заметим, что благодаря наличию средств настройки структуры тезауруса и поддержки ее семантических свойств данный подход может использоваться при построении многоязычных тезаурусов для любых языков и предметных областей.

Работа выполнена при финансовой поддержке РФФИ (проект № 10-04-12108в).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Андреева и др., 2006] Андреева О.А., Боровикова О.И., Булгаков С.В., Загорулько Ю.А., Сидорова Е.А., Циркин Б.Г., Холушкин Ю.П. Археологический портал знаний: содержательный доступ к знаниям и информационным ресурсам по археологии // Труды 10-й национальной конференции по искусственному интеллекту с международным участием - КИИ'2006. – Москва: Физматлит, 2006. -Т.3. -С.832-840.

[Боровикова и др., 2008] Боровикова О.И., Загорулько Ю.А., Загорулько Г.Б., Кононенко И.С., Соколова Е.Г. Разработка портала знаний по компьютерной лингвистике // Труды 11-ой национальной конференции по искусственному интеллекту с международным участием КИИ-2008. М.: ЛЕНАНД, 2008. Т.3. С.380-388.

[Википедия, 2011] Википедия. Статья «Тезаурус». <http://ru.wikipedia.org/wiki/Тезаурус>

[ГОСТ 7.24-2007] ГОСТ 7.24-2007. Система стандартов по информации, библиотечному и издательскому делу. Тезаурус информационно-поисковый многоязычный. Состав, структура и основные требования к построению (введен в действие с 1 июля 2008 г.).

[ГОСТ 7.25-2001] ГОСТ 7.25-2001. Система стандартов по информации, библиотечному и издательскому делу. Тезаурус информационно-поисковый одноязычный. Правила разработки, структура, состав и форма представления (введен в действие с 1 июля 2002 г.).

[Загорулько и др., 2008] Загорулько Ю.А., Боровикова О.И. Подход к построению порталов научных знаний // Автометрия. Новосибирск: 2008. Т. 44. № 1. С. 100–110.

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

[Загорулько и др., 2011] Загорулько Ю.А., Боровикова О.И., Кононенко И.С., Соколова Е.Г. Подход к разработке русско-английского тезауруса по компьютерной лингвистике // Труды XIII Всероссийской научной конференции RCDL'2011 «Электронные библиотеки: перспективные методы и технологии, электронные коллекции». Воронеж, 19-22 октября 2011 г. – Воронеж: Издательско-полиграфический центр Воронежского государственного университета, 2011.- С.27-34.

[Лукашевич, 2011] Лукашевич Н.В. Тезаурусы в задачах информационного поиска. –М.: Изд-во МГУ, 2011.

[Соколова и др., 2011] Соколова Е.Г., Семенова С.Ю., Кононенко И.С., Загорулько Ю.А., Кривнова О.Ф., Захаров В.П. Особенности подготовки терминов для русско-английского тезауруса по компьютерной лингвистике // Компьютерная лингвистика и интеллектуальные технологии. По материалам ежегодной международной конференции «Диалог» (Бекасово, 25-29 мая 2011 г.). - М.: РГГУ, 2011, Вып. 10(17), С. 644–655.

[ANSI/NISO Z39.19-2005] ANSI/NISO Z39.19-2005 Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies (Periodic Review).

[ISO 2788-1986] ISO 2788-1986. Documentation – Guidelines for the establishment and development of monolingual thesauri. Ed. 2.

[ISO 5964-1985] ISO 5964-1985. Documentation - Guidelines for the establishment and development of multilingual thesauri, IDT (Revised by: ISO/DIS 25964-1 Under development).

DEVELOPMENT OF MULTILINGUAL THESAURUS BY MEANS OF SEMANTIC TECHNOLOGY

Zagorulko Yu.A. ^{*,**}, Borovikova O.I. ^{*,**},
Zagorulko G.B. ^{*}

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

^{**} *Novosibirsk State University, Novosibirsk,
Russia*

zagor@iis.nsk.su

olesya@iis.nsk.su

gal@iis.nsk.su

The paper presents an approach to the development of multilingual electronic thesaurus for an arbitrary domain. A feature of the approach is the use of the formal and software tools provided by the semantic technology that was developed for the construction of scientific knowledge portals.

INTRODUCTION

Because of globalization and integration of scientific knowledge often the need for multilingual thesauri for different subject domains arises. However, the problem lies in the fact that at the moment there are no available, flexible, easy to use, but at the same time sufficiently powerful tools for development of such thesaurus.

As a tool for building a multilingual thesaurus, which satisfies all the above properties, we propose to use the methodology and software components of the semantic technology developed for the building of scientific knowledge portals and which has been previously used to create the knowledge portals on archaeology and computational linguistics.

MAIN PART

On the basis of the analysis of the requirements of the Russian and international standards that define guidelines and conventions for the construction of information retrieval thesauri, a general structure of the thesaurus and composition of the thesaurus entries was developed. The main units of the thesaurus are the terms of some subject domain, connected by the semantic relations, as well as descriptions of the knowledge areas and sources, i.e. text documents or

collections of text documents which include the terms of the thesaurus or their definitions. The terms of the thesaurus are divided into descriptors and ascriptors (text entries).

The methodology and software components of the semantic technology developed for the building of scientific knowledge portals was used as a tool for building the multilingual thesaurus. As this technology is based on ontology, the first step in the thesaurus building in its framework is a representation of a conceptual scheme of the thesaurus in the form of ontology, which we call the thesaurus representation ontology. This ontology will not only define the structure of information content of the thesaurus, but also serve as a basis for an organization of content-based access to the knowledge and data contained in it.

The developed thesaurus representation ontology describes the classes that represent the basic entities of the thesaurus (thesaurus terms, their sources, the areas/subareas of knowledge), relations linking the objects of these classes, properties of concepts and relations, as well as the axioms that define their additional semantics. In addition, the ontology specifies a set of domains, i.e. the possible values of attributes of the classes and relations to reduce the number of errors when creating/editing the thesaurus.

It should be noted, that the management of the thesaurus content is considerably simplified due to the fact that the logical consistency and integrity of the thesaurus concept system is provided by the special mechanisms of control and inference of knowledge which are built-in the data editor and the work of which is based on axioms describing properties of the relations and classes of terms.

To insert concrete terms, their definitions and sources into the thesaurus, as well as to link them by relations, the data editor provided by the technology of building of knowledge portals and run by an thesaurus representation ontology is used. This editor provides expert-linguists with a convenient web-interface for management of the thesaurus content.

With the aim of providing a distributed collaborative development of the thesaurus the used technology supports the mechanism of delegating the rights to experts of various levels. In accordance with this mechanism only experts of the high level can edit the structure of the thesaurus (using the ontology editor), while the experts of the lower level – only the thesaurus content (using the data editor).

Access to the terms of the thesaurus and its other entities is provided by the user web-interface, which is also provided by the technology of building of the knowledge portals. In this interface, the contents of the thesaurus is presented to the user in the form of a

network of interconnected information objects which present the thesaurus terms, descriptions of sub-areas of knowledge, as well as descriptions of the sources of the terms and their definitions.

When navigating the thesaurus a user can a possibility for selection of the required terms, a detailed view of their descriptions (thesaurus entries), as well as sources where the terms and/or them definitions appear.

CONCLUSION

The paper presents an approach to the development of multilingual electronic thesaurus, a general structure of which, composition of the thesaurus entries and set of relations between the terms meet the international and Russian standards. Feature of the approach is the use of formal and program facilities of technology of the building of the scientific knowledge portals as a tool for development. Due to the fact that this technology is based on ontology, with the help of which the conceptual scheme of the thesaurus (the thesaurus representation ontology) is described, the possibility of extension, the integrity and consistency of the terminology system of the thesaurus, and a convenient access to its content are provided.

The proposed approach is currently used for the creation of a Russian-English thesaurus on computational linguistics.

Note that due to the availability of facilities for adjustment of the thesaurus structure and support of its semantic properties this approach can be used for building of a multilingual thesaurus for any language and subject domain.



ДОСТОВЕРНОСТЬ ПРАВИЛ В ЭКСПЕРТНЫХ СИСТЕМАХ

Моросанова Н.А.

*Московский государственный университет имени М.В. Ломоносова, факультет ВМК
г. Москва, Россия*

nmosanova@gmail.com

В работе рассматриваются основные подходы к описанию достоверности правил вывода в экспертных системах, а также влияние на процесс вывода изоморфных преобразований коэффициентов уверенности.

Ключевые слова: достоверный вывод, коэффициенты уверенности, схема Шортлиффа.

ВВЕДЕНИЕ

В экспертных системах, основанных на правилах, знания представляются в виде фактов и правил [Abraham, 2005]. Как правила, так и факты могут являться гипотезами, которые считаются достоверными только с некоторой степенью уверенности.

Степень уверенности в фактах и правилах может быть описана в одних терминах, поскольку это степень уверенности в некоторой гипотезе. Однако при описании уверенности в правилах дополнительно требуется определить механизм влияния степени уверенности в правиле на степень уверенности в заключении правила. Степень уверенности в правиле вывода влияет на ход вывода и его результат.

1. Подходы к описанию уверенности в правилах

Для описания степени уверенности в правилах были разработаны различные способы [Liu, 1986]:

- байесовский подход;
- теория Демпстера-Шефера;
- теория подтверждений;
- теория ролей;
- коэффициенты уверенности.

В рамках байесовского подхода для описания уверенности правила используется формула условной вероятности, таким образом, определяется достоверность правила в различных условиях.

В оригинальной теории Демпстера-Шефера не определяется способа для описания уверенности в правилах. Однако в рамках этой теории были

разработаны несколько способов описания уверенности. Причем один из них аналогичен формуле условной вероятности, поскольку теория Демпстера-Шефера считается обобщением теории вероятностей. Также в рамках теории Демпстера-Шефера теория ролей [Liu, 1986] определяет роли правила, которые описывают его смысловое содержание. Для каждой из ролей описывается зависимость степени уверенности в заключении правила от уверенности в правиле. Ассоциативная роль описывает эмпирически установленные зависимости, для нее вводятся две вспомогательные роли: поддерживающая и противоречащая роли. Необходимая и достаточная роли описывают причинно-следственные зависимости, а для правил, используемых для опровержения фактов, вводится опровергающая роль.

В теории подтверждений определяются нечисленные описания степени уверенности в фактах и отношениях между правилами. Этот подход отличается от всех остальных тем, что в нем делается попытка вообще не использовать количественное описание степени уверенности в правиле.

1.1. Коэффициенты уверенности

Коэффициенты уверенности были предложены в экспертной системе MYCIN [Buchanan et al., 1984]. Этот подход также называется схемой Шортлиффа. В нем для описания степени уверенности используется число из отрезка $[-1, 1]$, называемое коэффициентом уверенности, причем -1 означает полную уверенность в опровержении гипотезы, а 1 – полную уверенность в ее подтверждении. Каждому правилу вывода приписывается некоторый коэффициент уверенности, что позволяет рассматривать правило вывода как гипотезу. Если в результате срабатывания такого правила его

консеквент получает коэффициент уверенности CF_1 , то его окончательное значение определяется как $CF_1 * CF_2$, где CF_2 – коэффициент уверенности правила. Поскольку в классической схеме Шортлиффа для получения этого значения используется умножение, будем называть эту операцию функцией умножения $tmx(a, b)$, где a и b – коэффициенты уверенности. Несмотря на то, что коэффициенты уверенности для правил неотличимы от коэффициентов уверенности для фактов, в правилах на практике используются лишь значения из полуинтервала $(0,1]$, поскольку для вывода используются достоверные в некоторой степени правила. Таким образом, один из аргументов функции $tmx(a, b)$ принимает значения из отрезка $[-1,1]$, а второй – из полуинтервала $(0,1]$.

1.2. Изоморфные трансформации схемы Шортлиффа

Изоморфные трансформации схемы Шортлиффа [Моросанова и др., 2011] с помощью взаимно однозначных отображений ставят своей целью получить более простой или удобный вид для некоторых из операций, входящих в схему Шортлиффа. Каждая трансформация задается с помощью функции $h: [-1, 1] \rightarrow R$, где R – некоторое множество. Поскольку функция h осуществляет взаимно однозначное отображение, то определена обратная функция h^{-1} . На множестве R определяются образы всех операций и отношений, имеющих место для коэффициентов уверенности на множестве $[-1, 1]$. В дальнейшем будем образы операций на R обозначать с помощью префикса $h|$ перед названием функции. Таким образом, в (1) определяется образ операции умножения $tmx(a, b)$.

$$h|tmx(A, B) = h(tmx(h^{-1}(A), h^{-1}(B))) \quad (1)$$

Обязательным требованием к h является взаимная однозначность задаваемого отображения. Однако часто бывает удобно рассматривать также монотонно возрастающие функции h , так как это ведет к упрощению сравнения коэффициентов уверенности и их образов с пороговыми значениями и их образами, соответственно. Одним из свойств монотонно возрастающих преобразований является также то, что при изоморфном отображении сохраняется отношение порядка для образов коэффициентов уверенности. Таким образом, если один из фактов более достоверен, чем другой, то и в новых терминах это будет так же.

1.3. Особенности трансформации функции умножения

Выбор конкретной трансформации зависит от вида информации, доступной для построения соответствующего отображения. Например, может быть известен вид функции комбинирования в изоморфной схеме, или известны некоторые значения функции, осуществляющей отображение.

В любом случае, функция умножения подчинена функции комбинирования, поскольку изначально [Buchanan et al., 1984] определяется как вспомогательная для комбинирования для организации вывода. Функция комбинирования, в свою очередь, в основном определяет процесс вывода, влияя таким образом на выбор изоморфной трансформации. К тому же, в классической схеме Шортлиффа наиболее сложный вид имеет как раз функция комбинирования, поэтому задача упрощения ее вида может решаться с помощью аппарата изоморфных трансформаций.

Пример 1. [Моросанова и др., 2011]

Пусть известно, что $h(0)=1$, $h(1)=+\infty$, $h(-1)=0$ и определен образ операции комбинирования $h|cmb(A,B)=A*B$. Тогда отображение h определяется как

$$h(x, \alpha) = \begin{cases} (1+x)^\alpha, & -1 \leq x < 0; \\ (1-x)^{-\alpha}, & 0 \leq x < 1; \\ +\infty, & x = 1. \end{cases} \quad \alpha > 0 \quad (2)$$

При таком определении функции отображения $h|tmx(A,B)$ имеет вид, представленный в табл.1.

Таблица 1 – Вид функции $h|tmx(A,B)$

Значение A	Значение B	Значение $h tmx(A,B)$
$[0, 1)$	$[1, +\infty)$	$\left(\frac{1}{A^\alpha} + B^{-\frac{1}{\alpha}} - \left(\frac{A}{B} \right)^{\frac{1}{\alpha}} \right)^\alpha$
$[1, +\infty)$	$[1, +\infty)$	$\left(\frac{1}{A^\alpha} + \frac{1}{B^\alpha} - \frac{1}{AB^\alpha} \right)^\alpha$
$+\infty$	$[1, +\infty)$	B
$[0, 1)$	$+\infty$	A
$[1, +\infty)$	$+\infty$	A
$+\infty$	$+\infty$	$+\infty$

Для полученной в примере 1 функции умножения можно построить приближения, имеющие более простой вид.

Пример 2. Приближение $h|tmx(A,B)$.

Зафиксируем B и α – некоторые положительные вещественные числа. Тогда обозначим

$$k = 1 - B^{-\frac{1}{\alpha}}, \quad l = B^{-\frac{1}{\alpha}}, \quad f(A) = h|tmx(A,B):$$

$$f(A) = \begin{cases} \left(k \cdot A^{\frac{1}{\alpha}} + l \right)^\alpha, & A \in [0, 1), \\ \left(k \cdot A^{-\frac{1}{\alpha}} + l \right)^{-\alpha}, & A \in [1, +\infty), \end{cases} \quad (3)$$

Рассмотрим для примера значения $\alpha = 2$, $B = 4$ и обозначим за $h(a)$ функцию отображения при таком значении α . Тогда $k = \frac{1}{2}$, $l = \frac{1}{2}$ и

$$f(A) = \begin{cases} \frac{1}{4}(A + 2\sqrt{A} + 1), & A \in [0, 1), \\ \frac{4A}{A + 2\sqrt{A} + 1}, & A \in [1, +\infty). \end{cases} \quad (4)$$

Попробуем приблизить $f(A)$ с помощью функции

$$g(A) = \frac{4A}{A + 2\sqrt{A} + 1} + \frac{1/4}{(A+1)^6} \quad (5)$$

Оценим расстояние между значениями $f(A)$ и $g(A)$. Расстояние между a и b , определяемое как $|a - b|$, порождает расстояние между $A = h(a)$ и $B = h(b)$, определяемое как $\text{dist}(A, B) = h(|h^{-1}(A) - h^{-1}(B)|)$.

Тогда расстояние между $f(A)$ и $g(A)$ определяется как $\text{dist_tmx}(A) = \text{dist}(f(A), g(A))$:

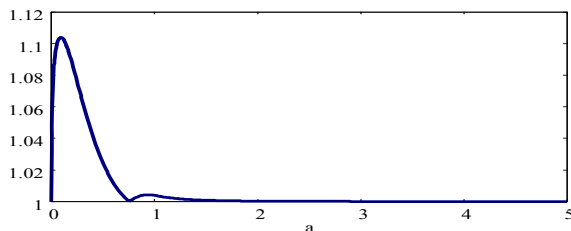


Рисунок 1 - $\text{dist_tmx}(A)$

На рис.1 видно, что для значений $A \in [0, 5]$ расстояние не превышает 1.2, что в свою очередь, не превышает $h(0.2)$. При этом $\lim_{A \rightarrow \infty} \text{dist_tmx}(A) = 1$, так

как $\lim_{A \rightarrow \infty} f(A) = \lim_{A \rightarrow \infty} g(A) = 4$. Покажем, что функция

$\text{dist_tmx}(A)$ убывает при $A > 4$. Для $A > 4$

$$f'(A) = \frac{4\sqrt{A} + 1}{(\sqrt{A} + 1)^4} > 0, \quad (6)$$

$$g'(A) = \frac{4\sqrt{A} + 1}{(\sqrt{A} + 1)^4} - \frac{1.5}{(A+1)^7} > 0,$$

Тогда при $A > 4$ $f(A) > 1$ и $g(A) > 1$, поскольку функции f и g возрастают. Тогда функция $\text{dist_tmx}(A)$ имеет следующую оценку сверху:

$$\sqrt{\frac{1}{(A+1)^6 \cdot f(A) \cdot 2\sqrt{f(A)}}} - 1, \quad (7)$$

причем (7) – убывающая функция. Таким образом, для $\forall A \geq 0$ значение функции $\text{dist_tmx}(A)$ не превышает $h(0.2)$. Значит, это расстояние не является существенным согласно порогам в схеме Шортлиффа и полученное приближение можно использовать.

Теперь зафиксируем $\alpha = 2$ и будем рассматривать различные значения k , учитывая, что $l = 1 - k$. При этом значение B будем брать таким, чтобы $B \geq h(0.2)$, тогда $k \geq 0.2$. Функция g определяется следующим образом:

$$g(A, k) = \frac{A}{((1-k)\sqrt{A} + k)^2} + \frac{f(0, k)}{(A+1)^6} \quad (8)$$

Функция расстояния $\text{dist_tmx}(A, k)$ определяется аналогично $\text{dist_tmx}(A)$ и имеет следующий вид для значений $k \in [0.2, 1]$:

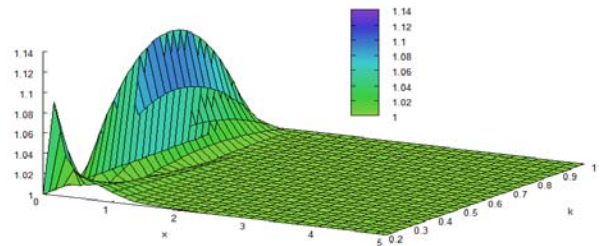


Рисунок 2 - $\text{dist_tmx}(A, k)$

Для функции $\text{dist_tmx}(A, k)$ можно аналогично соотношениям (6)-(7) показать, что при $A \rightarrow +\infty$ она убывает к единице ($h(0)=1$). Таким образом, в этом случае также расстояние не превышает $h(0.2)$. Поэтому полученное приближение функции умножения можно использовать для значения $\alpha = 2$.

Для других значений параметра α вид функции отображения h может сильно отличаться [Моросанова и др., 2011], поэтому поиск функции, приближающей $h|_{\text{tmx}}$, должен производиться для каждого α в отдельности.

2. Параметры правила вывода

Пусть с каждым правилом вывода R помимо коэффициента уверенности $b = CF(R)$ заранее связаны производные от него числа $P(R)$ и $Q(R)$:

$$P(R) = 1 - b,$$

$$Q(R) = \begin{cases} (1-b)^{-\alpha}, & 0 < b < 1, \\ +\infty, & b = 1. \end{cases} \quad (9)$$

Тогда $h|_{\text{tmx}}$ получит вид:

$$\begin{cases} (P(R) + b^{\alpha}\sqrt{A})^{+\alpha}, & 0 \leq A \leq 1, \\ (P(R) + b^{\alpha}(\sqrt{A})^{-1})^{-\alpha}, & 1 < A < +\infty, \\ Q(R), & A = +\infty. \end{cases} \quad (10)$$

После представления функции $h|_{\text{tmx}}$ в виде (10) становится видно, что

$h | \text{tmx}(A, R) =$

$h | \text{not}(h | \text{tmx}(h | \text{not}(A), R)),$

$$\text{где } h | \text{not}(A) = \begin{cases} 1/A, & 0 < A < +\infty, \\ 0, & A = +\infty, \\ +\infty, & A = 0. \end{cases} \quad (11)$$

что соответствует соотношению

$$c \cdot a = - (c \cdot (-a)). \quad (12)$$

Поэтому функция $h | \text{tmx}(A, R)$ полностью определяется своими значениями при $0 \leq A \leq 1$.

Рассмотрим один из вариантов интерпретации соотношения

$$(1 - b + b \cdot A), \quad \alpha = 1. \quad (13)$$

Связанные с правилом R числа b , $1 - b$ указывают распределение подтвержденности гипотез «правило R верно» и «достоверность R неизвестна» соответственно, то есть

$$b \cdot a = (1 - b) \cdot 0 + b \cdot a \text{ при } \alpha = 1. \quad (14)$$

Поскольку $h(0) = 1$, то соотношение (13) полностью аналогично соотношению (14).

ЗАКЛЮЧЕНИЕ

Коэффициенты уверенности правил и операция умножения в схеме Шортлиффа позволяют учитывать степень достоверности правил вывода. При изоморфных трансформациях схемы операция умножения может заметно усложняться, затрудняя ее использование и объяснение. Эту проблему можно решать несколькими способами: во-первых, использовать приближенную функцию, во-вторых, упрощать ее вид за счет заранее рассчитанных параметров правила.

Описанные способы решения проблемы усложнения операций при трансформациях схемы могут применяться и для других операций над коэффициентами уверенности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Моросанова и др., 2011] Моросанова, Н.А., Соловьев, С.Ю. Формальные свойства схемы Шортлиффа / Н.А. Моросанова, С. Ю. Соловьев // Управление большими системами. - 2011. - № 35.
- [Abraham, 2005] Rule based expert systems /In: P.S. Thorn, Richard (eds.) Handbook for Measurement Systems Design, pp. 909-919. – John Wiley and Sons Ltd., London, 2005.
- [Buchanan et al., 1984] Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project /Edited by B.G. Buchanan, E.H. Shortliffe - MA: Addison-Wesley, 1984.
- [Liu, 1986] Liu, G. S.-H. Causal and Plausible Reasoning in Expert Systems/ G. S.-H Liu // AAAI-86 Proceedings, pp. 220-225.

RULES UNCERTAINTY IN RULE-BASED EXPERT SYSTEMS

Morosanova N. A.

Lomonosov Moscow State University, Moscow, Russia

nmorosanova@gmail.com

This paper reviews main approaches to rules uncertainty representation in rule-based expert systems. The effect of the isomorphic transformations of certainty factors and associated operations on the inference process is studied.

INTRODUCTION

Knowledge in rule-based expert systems is represented by means of rules and facts, both of which may be merely hypotheses and those hypotheses may have a degree of uncertainty. For the rule there should be additionally defined a mechanism of calculating the degree of uncertainty of its consequent on the basis of the rule's degree of uncertainty.

MAIN PART

There are different methods for representing uncertainty in rules in expert systems (Bayes theory, Dempster-Shafer theory, theory of endorsements, certainty factors).

Isomorphic transformation of certainty factors using the transformation function leads to a definition of transformed operations (e.g. combination, multiplication). While some of operations are simplified after that transformation, other ones' images may be complicated functions that need to be simplified.

That can be done using approximation or pre-calculated rule parameters. Both methods are demonstrated on the example of multiplication image simplification. For the first method the measure of closeness of the approximation is defined, and the approximation functions for the multiplication image are proposed (see pic. 1 and 2). For the second case the interpretation of the simplified the multiplication image is given.

CONCLUSION

Certainty factors model proposed in [Buchanan et al., 1984] and the multiplication operation form a mechanism of representing the uncertainty in the inference rules in a rule-based expert system.

Isomorphic transformations of certainty factors and associated operations may lead to the loss of the simplicity of some operations while simplifying other ones. In particular, the multiplication operation may transform into a complicated function. Two approaches to simplify the multiplication image are proposed

Mentioned approaches to simplify operation images can be used not only for the multiplication, but for other operations in the certainty factors model as well.



УДК 004.822:514

СЕМАНТИЧЕСКИЕ МОДЕЛИ И СРЕДСТВА ИНТЕГРАЦИИ И ОТЛАДКИ БАЗ 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 отладки и интеграции баз знаний в рамках задачи разработки базы знаний для интеллектуальной системы.

База знаний рассматривается как связанная информационная конструкция, с заданной структурой [Ивашенко, 2009а], состоящая из знаков и связей отношений инцидентности между ними.

Задача отладки базы знаний заключается в построении базы знаний, удовлетворяющей некоторому конечному набору требований, на каждый которых может быть получен ответ «да», либо «нет». База знаний строится из исходного конечного множества фрагментов баз знаний. Требования формулируются на формальном логическом языке описания ошибок и описывают целостность, противоречия, неточности и полноту отлаживаемой базы знаний.

Задача интеграции баз знаний рассматривается как подзадача отладки и заключается в построении базы знаний из конечного множества исходных баз знаний и дополнительной метаинформационной конструкции.

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, TML, 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 существующим на настоящий момент средствам интеграции онтологий можно отнести: Optima, Prompt, Ontolingua, Chimaera [McGuinness et al., 2000], ONION [Gangemi et al., 1996], COMA++ [Aumuller et al., 2005] и др.

Однако все эти средства и методы имеют ограничения и не преодолевают в полной мере вышеперечисленные трудности.

1. Семантическая модель интеграции баз знаний и фрагментов баз знаний

Семантическая модель интеграции баз знаний и фрагментов баз знаний использует унифицированное представление баз знаний и обеспечивает интеграцию sc-моделей баз знаний и их фрагментов. Унифицированное представление баз знаний обеспечивается унифицированной моделью баз знаний и унифицированной моделью 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. Для описания статических предметных областей sc-языки используют математический аппарат теории множеств, для строгого описания динамических предметных областей, используются понятия нестационарной принадлежности и непринадлежности, которые задают понятия нестационарного множества. В SC-коде различаются понятия обозначения и понятия множества. Для обозначений и множеств рассмотрим функции T_{sign} и T_{set} , которые сопоставляют каждому обозначению и множеству множество моментов времени существования этого обозначения или множества. Таким образом, знак связи отношения принадлежности (непринадлежности), дуга arc , является обозначением связи нестационарной принадлежности (непринадлежности) тогда и только тогда, когда:

$$(T_{set}(begin(arc)) \cup T_{sign}(end(arc))) / T_{set}(arc) \neq \emptyset \quad (1)$$

Здесь $begin(arc)$ обозначает начало дуги arc , а $end(arc)$ – конец дуги arc . В ином случае дуга arc является обозначением стационарной принадлежности (непринадлежности).

В отличие от вышеперечисленных 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 (кодировки). 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 следующих теоретико-множественном концептов OWL 2 QL и OWL 2 EL [W3C:OWL2, 2009]: SubClassOf, EquivalentClasses, DisjointClasses, ClassAssertion, DisjointUnion, SameIndividual, SubDataPropertyOf, EquivalentDataProperties, DataPropertyAssertion, DataComplementOf, DataIntersectionOf, DataUnionOf, DataPropertyDomain, DataPropertyRange, DataSomeValuesFrom, DataMinCardinality, DataMaxCardinality, DataExactCardinality, FunctionalDataProperty, SubObjectPropertyOf, EquivalentObjectProperties, DisjointObjectProperties, ObjectPropertyAssertion, ObjectComplementOf, ObjectIntersectionOf, ObjectUnionOf, ObjectPropertyDomain, ObjectPropertyRange, ObjectMinCardinality, ObjectMaxCardinality, ObjectExactCardinality, FunctionalObjectProperty, InverseFunctionalObjectProperty, ReflexiveObjectProperty. Остальные концепты могут быть определены с помощью логического sc-языка.

Унифицированная модель баз знаний описывает базы знаний специального вида, в которых знания представлены с использованием унифицированной модели представления знаний [Ивашенко, 2009a]. В унифицированной модели баз знаний база знаний – связанная структурированная информационная

конструкция, структура которой состоит не менее чем из одного раздела, каждому из которых принадлежит свой, описываемый в этом разделе (ключевой) элемент этой конструкции, причем в описании хотя бы одного ключевого элемента присутствует его внешнее обозначение. 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, 2009a]. 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, 2011a].

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, 2009a]. 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, являющиеся 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

$$E(G) \subseteq (V(G))^2. \quad (2)$$

Для того, чтобы определить множество $I(G)$, используем специальную операцию

$$A \overset{\text{def}}{\cup} B = \bigcup_{(P,Q) \in A \times B} \{P \cup Q\}, \quad (3)$$

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

$$I(G) = \left(\bigcup_{e_{ij} \in E(G)} \{\emptyset, \{e_{ij}\}\} \right) \cap C(G). \quad (4)$$

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).

$$Q = |I(G)|. \quad (5)$$

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 принадлежностей 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 $\{\equiv, \supseteq, \subseteq, \perp, \sqcap\}$, соответственно – совпадения, обобщения, частности, строгого исключения и строгого пересечения. Например, эти отношения в проекции на теоретико-множественные соответствуют отношениям равенства множеств ($=$), надмножества (\supseteq), подмножества (\subseteq), 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 1– Пример, приводящий к противоречивому отображению

Для приведённого примера (рис. 1), 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 \wedge параллелограмм (S8), замкнутая ломаная \wedge четырёхугольник \wedge параллелограмм \wedge ромб (S9)).

Таблица 1 – Противоречивое отображение

	S1	S2	S3	S4
S5	\equiv	\supseteq	\supseteq	\supseteq
S6	\subseteq	\equiv	\supseteq	\supseteq
S7	\subseteq	\subseteq	\equiv	\equiv
S8	\subseteq	\subseteq	\equiv	\supseteq
S9	\subseteq	\subseteq	\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 [Aumuller et al., 2005], [Jean-Mary et al., 2007], [Nagy et al., 2010].

Исходя из этого, автором предложен следующий набор отношений $\{\text{не уточнены}^*, \text{связность}^*, \text{различие}^*, \text{исключение}^* (\text{не обобщает}_-, \text{не конкретизирует}_-), \text{возможное включение}^* (\text{потенциально конкретизирует}_-, \text{потенциально обобщает}_-), \text{симметричное исключение}^*, \text{строгое пересечение}^*, \text{совпадение}^*\}$ отношение потенциальной синонимии можно выразить, как $\text{возможная синонимия}^* = (\text{не уточнены}^* \cup \text{связность}^*) / (\text{различие}^* \cup \text{исключение}^* \cup \text{возможное включение}^*)$.

Следующее правило (рис. 2) 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).

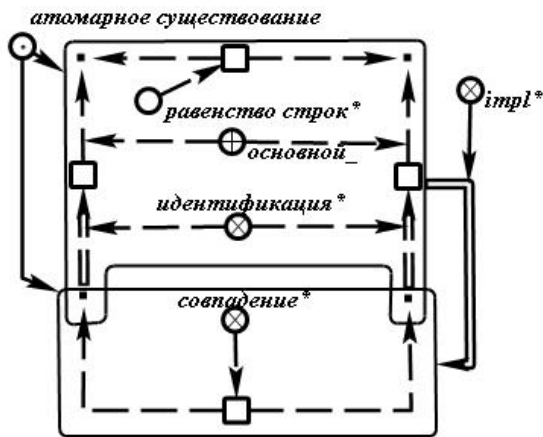


Рисунок 2 – Пример, приводящий к противоречивому отображению

Стратегии слияния при использовании методов лексико-терминологического анализа подробно рассмотрены в [Ивашенко, 2011b].

Следующие правила (рис. 3-8) описывают установление отображающих отношений на множестве знаков интегрируемых фрагментов баз знаний, после проведённых слияний в результате лексико-терминологического анализа.

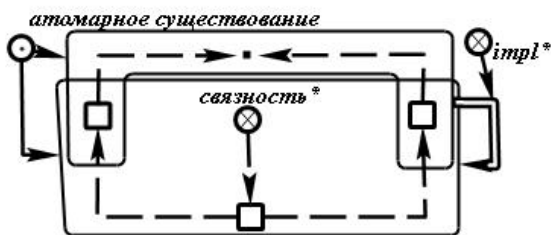


Рисунок 3 – Свойство отношения связности («два множества связны, если имеют общий элемент»)

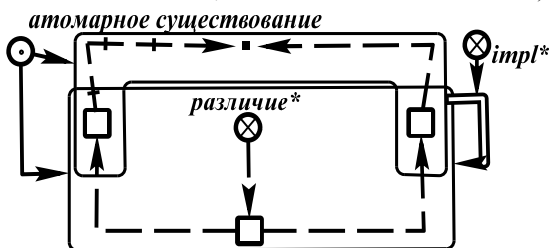


Рисунок 4 – Свойство отношения различия («два элемента различны, если один принадлежит множеству, а другой – нет»)

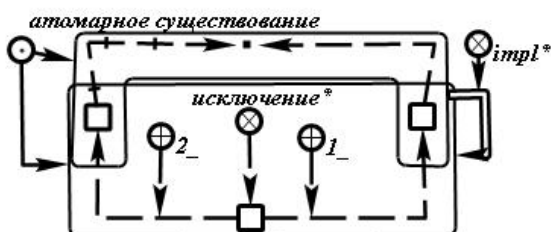


Рисунок 5 – Свойство отношения исключения (не обобщает, не конкретизирует)

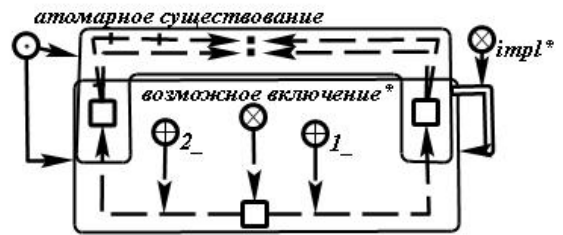


Рисунок 6 – Свойство отношения возможного (потенциального) включения

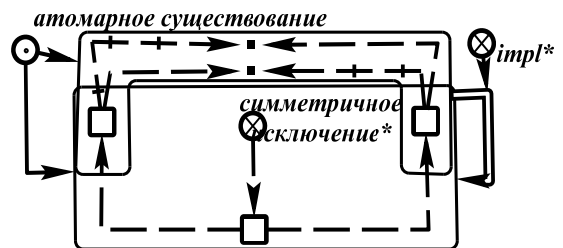


Рисунок 7 – Свойство отношения симметричного исключения

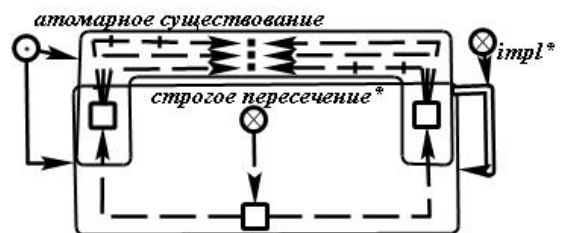


Рисунок 8 – Свойство отношения строгого пересечения

Все эти отношения можно однозначно и непротиворечно установить в результате соответствующего базовой теоретико – множественной семантике 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. Таковыми 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. 9), 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).

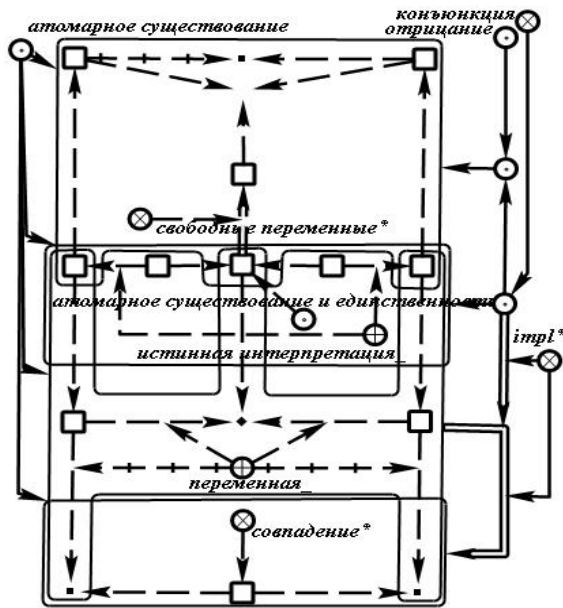


Рисунок 9 – Высказывание о слиянии значений переменных в утверждениях единственности

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. 10) приведён алгоритм выявления пар потенциальных синонимов, использующий вышеописанные правила экстенционального метода.

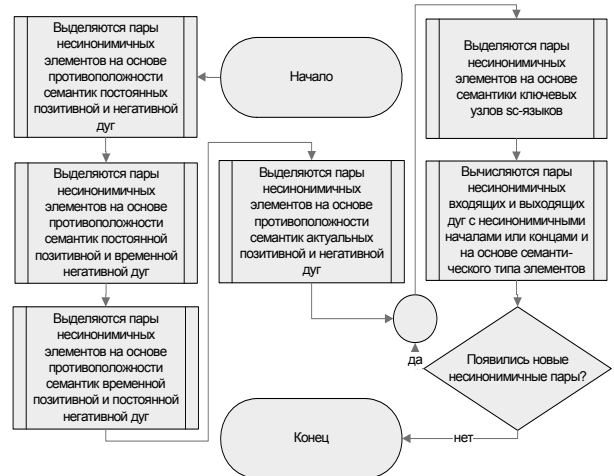


Рисунок 10 – Алгоритм вычисления множества пар различающихся (несинонимичных) элементов

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. Второй вариант предполагает формирование мета-разделов, которые описывают основные разделы исходных баз знаний и их структуру и на которые декомпозируется основной раздел интегрированной базы знаний. Второй вариант интеграции структуры баз знаний может иметь смысл, когда содержание исходных баз знаний является слабо связным.

Рассмотрим пример интеграции двух фрагментов баз знаний (см. рис. 11).

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.

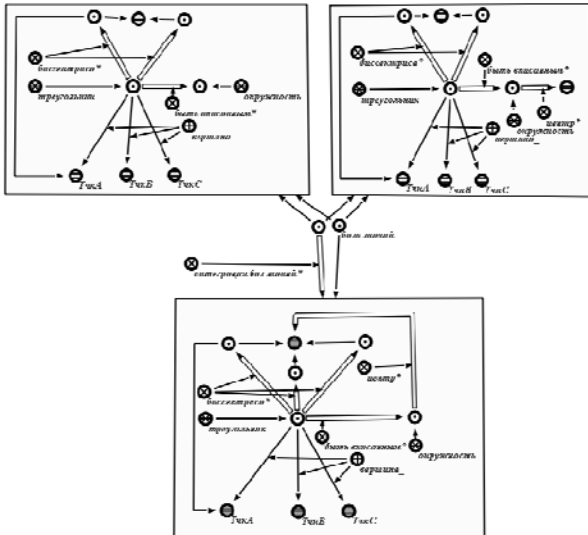


Рисунок 11 – Пример интеграции двух фрагментов баз знаний

2. Компонентное проектирование баз знаний

2.1. Модель компонентов баз знаний

Модель компонентов баз знаний описывает в соответствии с классификацией классы специфицируемых баз знаний и их фрагментов и отношения между ними и их частями. Спецификация базы знаний включает указание класса (типа) базы знаний и описание её количественных и качественных характеристик: состав, сертификат, задачно-ориентированный сборник тестовых вопросов, требования полноты и безошибочности базы знаний, информацию о разработчиках, условиях распространения и информацию для сопровождения.

2.2. Библиотека компонентов баз знаний

Библиотека компонентов баз знаний основывается на модели компонентов баз знаний, которая в свою очередь основана на унифицированной модели баз знаний, и также состоит из базы знаний, машины обработки знаний и пользовательского интерфейса. В зависимости от соотношения значений собственных характеристик [Ивашенко, 2011] и других признаков база знаний может быть классифицирована как:

- терминологический словарь
 - одно-языковой

- многоязыковой

- таксономия
 - конечного числа объектов
 - бесконечного числа объектов
- реляционная модель (онтология)
 - графовая реляционная модель
 - неграфовая реляционная модель
- тезаурус
 - с рекурсивными определениями
 - без рекурсивных определений
- теория
 - теория стационарной предметной области
 - теория нестационарной предметной области
- прикладная база знаний
 - фактографическая
 - нефактографическая
 - задачник
 - библиотека алгоритмов

Эта классификация, как и классификация фрагментов баз знаний, является основой модели компонентов баз знаний.

База знаний библиотеки компонентов баз знаний содержит каталогизированный набор компонентов, операциями машины обработки знаний библиотеки компонентов являются операции поиска и редактирования компонентов в этом каталоге, пользовательский интерфейс поддерживает соответствующие команды запуска этих операций. В структурную декомпозицию базы знаний библиотеки компонентов входят следующие разделы:

- раздел задачно-ориентированной спецификации базы знаний библиотеки компонентов,
- раздел типологии и обобщённых спецификаций компонентов различных типов,
- разделы типов компонентов, в каждом из которых представлены спецификации и описания на исходных текстах компонентов соответствующего типа,
- разделы, описывающие структуру каталога компонентов по различным признакам,
- разделы help-подсистемы.

3. Гипертекстовое представление семантических моделей баз знаний

Для представления исходных текстов баз знаний используются SCg [Голенков и др., 2001] и язык гипертекстового представления в псевдоестественном виде – SCn (Semantic Code natural). Язык SCn является некоторым аналогом языка Gellish и предназначен для упрощения записи и чтения человеком исходных текстов баз знаний.

Тексты языка SCn представляют собой набор sc.n-статей. Каждая sc.n-статья описывает некоторый sc-элемент и состоит из идентификатора этого элемента и набора sc.n-полей, каждое из

которых описывает одну или несколько связей отношения (рис. 12).

Если sc.n-статья имеет единственное sc.n-поле, то такая статья называется sc.n-предложением.

В SCn различаются следующие типы полей:

- sc.n-поле общего вида,
- sc.n-поле специального вида.

По признаку ориентированности отношения:

- sc.n-поле, описывающее связи ориентированного отношения,
- sc.n-поле, описывающее связи неориентированного отношения,

По признаку арности отношения:

- sc.n-поле, описывающее связи бинарного отношения,
 - sc.n-поле, описывающее связи ролевого отношения,
 - sc.n-поле, описывающее связи неролевого бинарного отношения,
- sc.n-поле, описывающее связи небинарного отношения.

По признаку количества описываемых связей:

- sc.n-поле, описывающее одну связь отношения,
- sc.n-поле, описывающее несколько связей отношения.

К sc.n-полям специального вида относятся:

- sc.n-поле отношения синонимии,
- sc.n-поле отношения пояснения,
- sc.n-поле отношения подмножества,
- sc.n-поле отношения надмножества,
- sc.n-поле отношения принадлежности,
- sc.n-поле отношения семантической эквивалентности,
- sc.n-поле отношения семантической близости,
- sc.n-поле отношения разбиения,
- sc.n-поле отношения определения,
- sc.n-поле отношения примера,
- и другие.

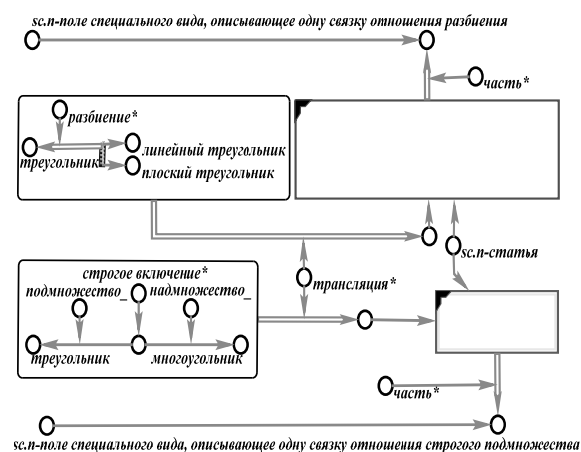


Рисунок 12 – Пример представления на SCn и эквивалентного представления на SCg

Для разметки текста в виде sc.n-статей, используется специальный язык разметки SCnML [Колб, 2012]. Некоторые sc.n-тексты могут быть транслированы в RDF.

4. Семантическая модель отладки баз знаний

4.1. Анализ качества баз знаний

Задача отладки БЗ состоит в тестировании, верификации [Ивашенко, 2011], проверке синтаксической корректности структуры баз знаний и автоматическом и ручном исправлении найденных неточностей и ошибок, основные этапы её решения представлены на рисунке 13.

В отладку базы знаний входят такие задачи как:

- верификация базы знаний,
- тестирование базы знаний,
- редактирование базы знаний.

Семантическая модель анализа качества баз знаний задаётся:

- множеством баз знаний, для которых определены наборы требований верификации в виде логических утверждений;
- множеством авторов-разработчиков фрагментов базы знаний, включая группы авторов;
- отношением авторства фрагментов базы знаний;
- конъюнкциями логических утверждений (описаний), описывающих непосредственно декларативно, либо в виде спецификации процедурных программ sc-операций:
 - нецелостность или неполноту базы знаний;
 - противоречивость или неточность базы знаний и варианты их автоматического исправления в виде:
 - изменения типа элементов;
 - удаления элементов базы знаний;
 - слияния элементов базы знаний;
 - генерации новых элементов базы знаний;

• текстами программ sc-операций редактирования и реализации проверки вышеописанных утверждений с помощью поиска и вычислений и sc-операций, обеспечивающих взаимодействие с разработчиком.

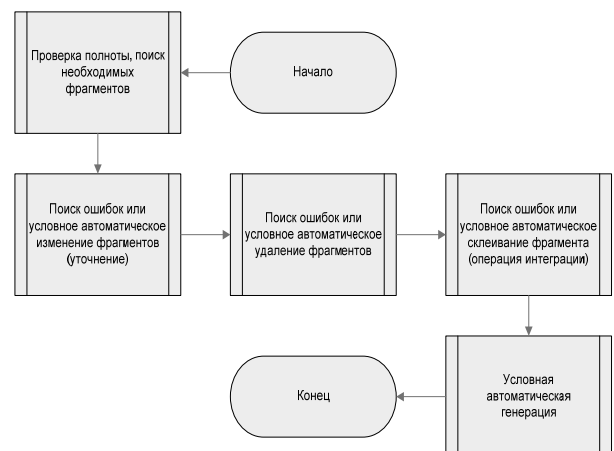


Рисунок 13 – Алгоритм верификации, основные этапы

Классификация описаний верифицируемых ошибок представлена в [Ивашенко, 2011a].

При выявлении ошибок локализуется раздел, в котором выявлена ошибка, и, при необходимости и наличии соответствующей информации, – sc.n-поле в исходном тексте, которое описывает ошибочную связку. Сообщение о выявленной ошибке отправляется разработчику (рис. 14).

Для анализа качества используются следующие sc-операции:

- sc-операция проверки, поиска и выявления ошибок нелокальной целостности и полноты
- sc-операция проверки, поиска и выявления ошибок, связанных с целостностью, полнотой, противоречивостью и неточностью, для заданного фрагмента базы знаний
- sc-операция поиска и указания раздела или поля исходного текста с ошибкой
- sc-операция формирования сообщений указанному разработчику об ошибках в указанном разделе
- sc-операции выявления и подсчёта потенциально синонимичных элементов базы знаний

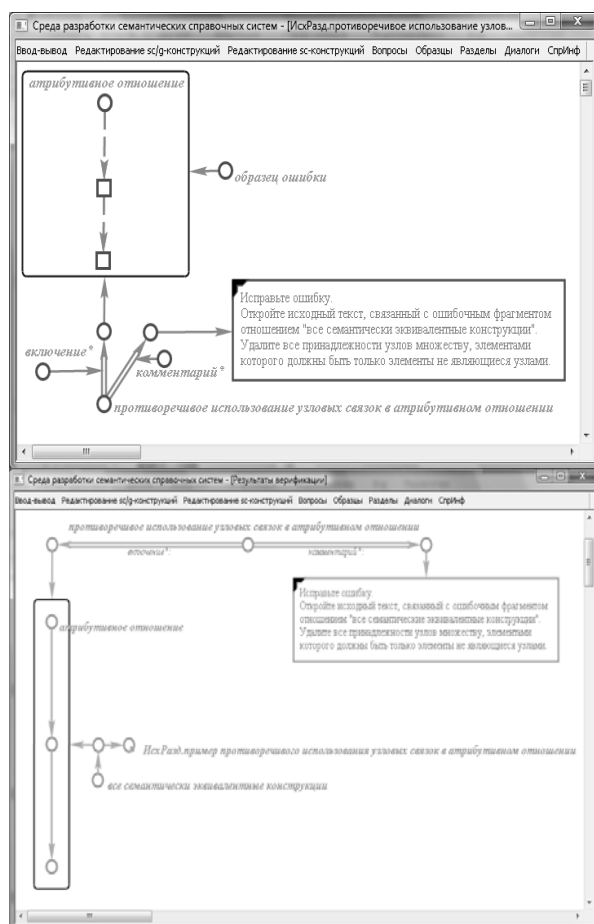


Рисунок 14 – Пример описания ошибочной ситуации противоречия и пример сообщения об обнаруженном противоречии

4.2. Тестирование баз знаний

Задача тестирования баз знаний требует условий

и средств поддержки безотказной и надёжной работы системы и возможности восстановления от сбоев. В отсутствие таких средств тестирование осуществляется большей частью вручную разработчиком на собственный риск с ручным восстановлением работоспособности в случае сбоев системы.

Семантическая модель тестирования баз знаний задаётся:

- множеством баз знаний, для которых определены наборы требований тестирования в виде логических утверждений;
- множеством авторов-разработчиков фрагментов базы знаний, включая группы авторов;
- наборами прав доступа разработчиков к фрагментам базы знаний;
- отношением авторства фрагментов базы знаний;
- конъюнкциями логических утверждений, описывающих непосредственно декларативно, либо в виде спецификации процедурных программ sc-операций тестовые команды и запросы к разрабатываемой базе знаний и ожидаемые правильные и ошибочные ответы;
- текстами программ sc-операций реализации управления правами доступа и состоянием базы знаний;
- текстами программ sc-операций редактирования и реализации проверки вышеописанных утверждений с помощью поиска и вычислений и sc-операций, обеспечивающих взаимодействие с разработчиком.

Классификация описаний верифицируемых ошибок представлена в [Ивашенко, 2011a].

Для тестирования баз знаний используются следующие sc-операции:

- sc-операция ввода произвольно-конструируемого запроса или команды к базе знаний
- sc-операция выявления несанкционированного доступа (нарушения целостности) базы знаний
- sc-операция восстановления состояния базы знаний
- sc-операция сохранения текущего состояния базы знаний
- sc-операции управления правами доступа к фрагменту базы знаний
- sc-операция формирования сообщений указанному разработчику об ошибках в указанном разделе
- sc-операции редактирования спецификации базы знаний (требований верификации)

4.3. Редактирование баз знаний

Редактирование базы знаний может быть осуществлено разработчиком с помощью классов пользовательских команд, которые задаются с помощью действий. Кроме этого разработчик может редактировать исходные тексты базы знаний.

Исправление ошибок в исходных текстах и синхронизация исходных текстов и базы знаний в памяти системы осуществляется разработчиком вручную.

Семантическая модель редактирования баз знаний задаётся:

- множеством баз знаний, для которых определены команды и операции редактирования;
- множеством авторов-разработчиков фрагментов базы знаний, включая группы авторов;
- наборами прав доступа разработчиков к фрагментам базы знаний;
- отношениями между редактируемыми разделами и фрагментами (полями) базы знаний;
- отношением авторства фрагментов базы знаний;
- текстами программ sc-операций реализации управления правами доступа и состоянием базы знаний;
- текстами программ sc-операций редактирования и поиска и формирования выборок редактируемых элементов;
- текстами программ sc-операций, обеспечивающих взаимодействие с разработчиком.

Для редактирования баз знаний используются следующие sc-операции:

- sc-операции поиска элементов базы знаний:
 - sc-операция поиска подразделов раздела базы знаний
 - sc-операция поиска подраздела базы знаний
 - sc-операция поиска элементов множества
 - sc-операция поиска по образцу
 - sc-операция экстенционального поиска
- sc-операции формирования выборки (выделения) найденных элементов базы знаний:
 - теоретико-множественные sc-операции над выборками
 - sc-операция объединения выборок
 - sc-операция пересечения выборок
 - sc-операция вычитания выборок
- sc-операция удаления элементов;
- sc-операция удаления выборки элементов;
- sc-операция изменения типа указанного элемента;
- sc-операции генерации элементов заданного типа, заданных инцидентными элементами;
- sc-операции генерации элементов заданных образцом.

ЗАКЛЮЧЕНИЕ

Средства проектирования баз знаний обеспечивают поддержку решения задачи отладки баз знаний, включая интеграцию баз знаний. В состав средств проектирования баз знаний входят:

- внешний интерфейс средств проектирования БЗ
 - пользовательский интерфейс
 - средства программного ввода\вывода и трансляторы
- отладчик баз знаний

- редактор баз знаний
- верификатор баз знаний
- средства тестирования баз знаний
- интегратор баз знаний.

Каждый из компонентов имеет свою базу знаний и систему операций. Базы знаний перечисленных компонентов образуют базу знаний средств проектирования баз знаний.

База знаний средств проектирования баз знаний содержит определяемую унифицированной моделью баз знаний метаинформацию о проектируемой базе знаний и имеет свою структурную декомпозицию, в которую входят следующие разделы:

- разделы описания процесса проектирования базы знаний и истории взаимодействия с разработчиком базы знаний;
- разделы описания методики проектирования и классов ошибок, присутствие которых выявляется при верификации базы знаний;
- разделы программ операций отладки и интеграции соответствующей машины обработки знаний и их спецификаций;
- разделы проектируемой базы знаний.

Пользовательский интерфейс средств проектирования баз знаний поддерживает необходимый набор соответствующих классов пользовательских команд и использует для общения с пользователем специальные языки (SCg [Голенков и др., 2001], SCn), которые обеспечивают наглядность, совместимость с современными web-технологиями и упрощают процесс наполнения, анализа и редактирования базы знаний.

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, А.В. Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков[и др.] – Мн.: БГУИР, 2001.
- [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.
- [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, 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, 2009а] Ивашенко В.П. Семантические модели баз знаний / В.П. Ивашенко Информационные системы и технологии (IST'2009): материалы V Международ. конф.-форума 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.
- [Кудрявцев, 2008] Кудрявцев Д. В. Практические методы отображения и интеграции онтологий. Семинар Знания и онтологии *Elsewhere*, КИИ-2008, Дубна, 2008.
- [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
- [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.
- [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.
- [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>.
- [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. Zaihrayeu (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-ODBS Workshop, September 2007, Vienna, Austria, 15-20.
- [IDEF5, 1994] Information Integration for Concurrent Engineering (IICE). 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.
- [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.
- [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.
- [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. ISBN 90-407-2597-4. <http://repository.tudelft.nl/view/ir/uuid%3Ade26132b-6f03-41b9-b882-c74b7e34a07d/>.

SEMANTIC MODELS AND TOOLS FOR KNOWLEDGE BASE INTEGRATION AND DEBUGGING

Ivashenko V.P.

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

ivashenko@bsuir.by

The work concerns structure and application of tools component knowledge base design technology (OSTIS) for knowledge base integration, verification and correction. This technology is based on homogeneous semantic network having set-theoretic semantic.



УДК 004.8

ИЗВЛЕЧЕНИЕ ЗНАНИЙ ПОЛЬЗОВАТЕЛЯ И ВЕРИФИКАЦИЯ ЗНАНИЙ САМООРГАНИЗУЮЩИХСЯ ИНФОРМАЦИОННЫХ СИСТЕМ С ОБРАТНОЙ СВЯЗЬЮ

Диковицкий В.В.*

* Учреждение Российской академии наук

Институт информатики и математического моделирования технологических процессов

Кольского научного центра РАН,

г. Анатиты, Российская Федерация

dikovitsky@iimm.kolasc.net.ru

В работе предложен метод формализации различий знаний о предметной области между пользователями и базой знаний информационной системы, и их использование для поддержки актуальности базы знаний.

Ключевые слова: база знаний, интеллектуальная система, предметная область, семантическая сеть.

ВВЕДЕНИЕ

Актуальной на сегодняшний день является задача обработки информационными системами различий знаний о предметной области как между пользователями, как и между конкретным пользователем и базой знаний (БЗ) информационной системы (ИС). В работе [Gruber, 1995], данное различие трактуется как проблема человеко-машинного и межкультурного взаимодействия. Одной из главных задач разработки онтологий является выработка единого словаря для пользователей ИС [Pinto, 1999]. Данная задача актуальна для формализации общепринятых знаний, и не может рассматриваться в рамках мультипредметных динамических предметных областей (ПрО). В реальных условиях часто даже общепринятые знания трактуются разными пользователями по-разному, и разработка единого словаря для них не только нецелесообразна, но и невозможна. Различные представления пользователей о ПрО представляется возможным использовать для получения новых и верификации имеющихся знаний, что особенно актуально для машинно-добавляемых знаний. В работе представлен метод динамического формирования БЗ на основе пользовательских знаний, позволяющий динамически в автоматическом режиме формировать пространство понятий пользователей ИС.

1. Извлечение знаний пользователя

Модель пользователя ИС – это, прежде всего его информационная модель, которая состоит из множества понятий и отношений между ними (1), основное назначение модели пользователя –

хранить область интересов конкретного пользователя.

$$UM = \langle T, L, W_T \rangle, \quad L = \langle T_i, T_j, W \rangle, \quad (1)$$

где T – множество тем для поиска, соответствующих концептам базы знаний, W_T – вес концепта в модели конкретного пользователя, L – множество отношений над концептами, W – вес отношений.

Структура модели пользователя (1) изоморфна структуре БЗ ИС, т.е. модель пользователя – это своего рода БЗ, содержащая знания пользователя, известные информационной системе. Следовательно, представляется возможным рассматривать отличные от БЗ представления пользователя о предметной области как новые знания:

$$NK = \langle T', L', W'_T \rangle, \quad (2)$$

где T' – множество тем для поиска, не соответствующих концептам базы знаний, W'_T – вес термина в модели конкретного пользователя.

Таким образом, различные представления пользователей о предметной области – это разница между информационной моделью пользователя и базой знаний ИС. Суммирование новых знаний (2) позволяет выявить синонимы концептов БЗ, новые виды в классификации понятий и т.п. Извлекать пользовательские знания представляется возможным путем обработки запросов пользователя, а так же путем учета статистики его взаимодействия с динамическими навигационными структурами (Рисунок 1).

Запрос можно представить как множество тем для поиска, задающих информационную потребность пользователя.

$$UQ = \langle T, W_T \rangle \quad (3)$$

где T - множество ключевых слов - тем для поиска, W_T - вес концепта в модели конкретного пользователя.

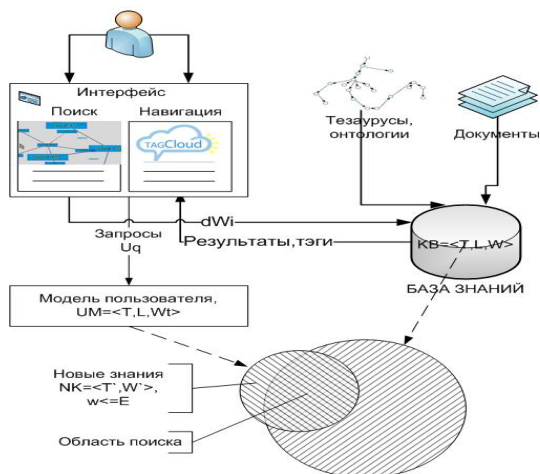


Рисунок 1 - Извлечение знаний пользователя

При пересечении множества (3) с БЗ, часть модели пользователя, непересекающаяся с БЗ, представляет новые знания(4), дополняющие БЗ.

$$UM \cap KB = \langle T', L', W' \rangle = NK, W' \leq \varepsilon \quad (4)$$

Вес связей при добавлении минимален и получает приращение с каждым обращением. Таким образом, связи являются предпосылками, и, в случае обнаружения ИС документа, имеющего пересечение с данной структурой (4) – вес связей преодолевает порог отображения ε (количественная характеристика, соответствующая весу отношений между понятиями БЗ, при котором они доступны для пользователей ИС). Это позволяет, например, выявить и добавить в БЗ отсутствующие синонимы понятий, описанных в документах, отсутствующие категории классификации и отображать их при следующих обращениях к БЗ. Модель пользователя хранит область интересов пользователя и позволяет определить контекста

2. Модификация базы знаний

Поддержка актуальности хранимых знаний представляется возможной путем выработки воздействий, направленных на модификацию БЗ - удаление и добавление объектов базы знаний. Удаление концепта семантической сети происходит при устремлении к нулю весов отношений - несвязные концепты подлежат удалению. Добавление новых знаний (4) к семантической сети происходит с весами, не превышающими порог отображения ε , т.е. новые знания, представленные в виде различий между представлений о ПрО в модели пользователя и БЗ, доступны только для компонентов приобретения знаний.

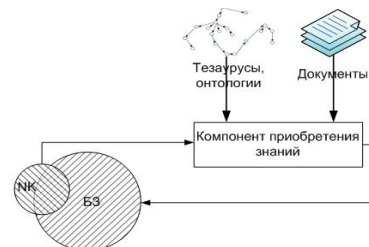


Рисунок 2 – Модификация базы знаний

Таким образом, знания NK существуют в виде предпосылок для компонента приобретения знаний, и подтверждение данных знаний является критерием добавления в БЗ.

ЗАКЛЮЧЕНИЕ

В работе предприняты попытки формализации знаний пользователя как различий его представлений о ПрО и их использование для формирования модели пользователя, основная задача которой – уменьшение неопределенности предмета поиска и верификации БЗ.

Верификация БЗ актуальна для систем, использующих автоматизированные компоненты приобретения знаний. Обработка естественного языка влечет за собой неопределенность, снизить которую представляется возможным с помощью обратной связи с пользователем. Кроме того, данный метод позволяет рассматривать различия модели пользователя и базы знаний как новые знания, что является одним из аспектов развития человеко-машинного взаимодействия.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Gruber, 1995] Gruber T. R. «Toward principles for the design of ontologies used for knowledge sharing». Presented at the Padua workshop on Formal Ontology, March 1993

[Pinto, 1999] Pinto HS, Gómez-Pérez A, Martins JP (1999) Some Issues on Ontology Integration. / In Proc. of IJCAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends. Vol. 18, pp. 1 - 12. Stockholm, Sweden, 1999.

EXTRACTION OF USER KNOWLEDGE AND VERIFICATION OF KNOWLEDGE SELF-ORGANIZING INFORMATION SYSTEMS WITH FEEDBACK

Dikovitsky V.V. *

* Establishment of Russian Academy of Sciences Institute for Informatics and Mathematical Modelling of Technological Processes of the Kola Science Center RAS, Apatity, Russian Federation

dikovitsky@iimm.kolasc.net.ru

In this paper we propose a method of formalizing differences between knowledge of users and knowledge base of information systems, and using this differences for verification knowledge base.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822, 004.891

РАЗРАБОТКА ЭКСПЕРТНЫХ СИСТЕМ В СРЕДЕ MULTI STUDIO

Катаев В.А.

*ОАО «Пермская научно-производственная приборостроительная компания»,
г. Пермь, Российская Федерация*

Bravo55555@yandex.ru

В работе содержится краткое описание разработки статических экспертных и информационных систем в программной среде Multi Studio российской разработки. Среда базируется на универсальном языке Multi и универсальных Multi-сетях, являющихся комбинацией синтаксических и семантических сетей. Представлены результаты сравнения Multi Studio с известной средой CLIPS.

Ключевые слова: семантические сети, экспертные системы, CLIPS, Multi Studio.

ВВЕДЕНИЕ

Экспертные системы (ЭС) относятся к классу интеллектуальных систем (ИС). Проектирование ИС в настоящее время имеет несколько принципиальных проблем. В статье [Голенков, 2011] обозначены следующие недостатки современных технологий проектирования ИС:

- Технологии искусственного интеллекта (ИИ) не ориентированы на широкий круг разработчиков ИС.
- Велики сроки разработки и трудоемкость сопровождения ИС.
- Высока степень зависимости технологии ИИ от платформ их реализации.
- Низкий уровень интеграции научных и практических результатов в области ИИ.
- Современные компьютеры плохо приспособлены для решения задач ИИ.

В этой же статье предлагаются пути устранения указанных недостатков:

- Ориентация на семантическое представление знаний.
- Унификация моделей ИС.
- Блочное проектирование ИС с использованием типовых компонентов.
- Поэтапное эволюционное проектирование на основе прототипов.
- Использование одинаковых принципов при проектировании ИС и инструментальных средств их разработки.
- Разработка подсистем помощи для разработчиков ИС и для их пользователей.
- Включение в ИС подсистемы автооптимизации.

- Обеспечение максимальной независимости эволюции баз знаний (БЗ) от эволюции решателей задач.
- Разработка языка унифицированного кодирования семантических сетей.
- Добавление в машину обработки знаний (МОЗ) новых агентов, не требующих внесения изменений в действующие.
- Трактовка пользовательского интерфейса ИС как специализированной ИС, имеющей свою БЗ и свою МОЗ.

При решении указанных проблем мы исходим из того, что главным направлением развития ИС является совершенствование языка проектирования ИС и разработки эффективной среды реализации этого языка. Необходимые свойства языка:

- Высокий уровень (принципиально приближенный к естественному языку).
- Простота.
- Универсальность.
- Блочность.
- Итеративность.
- Рекурсивность.
- Синонимичность.
- Омнимичность.
- Контекстность.
- Возможность манипулирования объектами сетевых структур, как наиболее универсальных.

Необходимые свойства среды разработки ИС:

- Простота.
- Универсальность.
- Мультиплатформенность.

- Среда должна позволять эффективно обрабатывать сетевые информационные структуры (базы знаний).
- Наличие интерфейсов с основными информационными системами.

Нами предпринята попытка реализации указанных свойств в среде Multi Studio, которая базируется на языке Multi и универсальной информационной структуре, называемой Multi-сетью.

1. Среда Multi Studio

Multi Studio (MS) – это универсальный многооконный диалоговый интерпретатор команд языка Multi, работающий под управлением грамматики, написанной на самом языке Multi.

Меню и системные сообщения (например, об ошибках) – на базовом языке, которым является русский. Имеется возможность использования в качестве базового языка других естественных языков. Система позволяет обработку текстов на русском, английском и на других языках.

Multi – это высокоуровневый командный (алгоритмический) язык с функциональной парадигмой и префиксной формой записи. Его принципиальный (базовый) синтаксис:

- Multi-текст – это непустой список предложений.
- Предложение – это непустой список выражений, разделенных пробелами. В конце предложения стоит точка с запятой.
- Выражение – это объект и необязательный аргум.
- Объект может быть командой, числом, литеральной строкой, именем, датой и т.п.
- Аргум – это список выражений, заключенный в круглые скобки.
- В конце предложения закрывающие круглые скобки аргумов могут отсутствовать.

Пример:

$P\# (\text{"сумма="} + (X\ 123 ; X (5000)); \quad (1)$

где $P\#$ – вывод на экран объектов аргума через пробел.

Результат: сумма = 5123

Универсальность (функциональная, технологическая, предметная) системы определяется универсальностью командного языка.

Функциональная универсальность предполагает наличие в языке средств (команд, функций) для обработки данных различной логической структуры (включая базы данных, базы знаний регулярной и произвольной структуры, тексты искусственных и естественных языков). Данные и программы их обработки в системе имеют одинаковый унифицированный формат и хранятся совместно в однородной электронной памяти, в так называемом бункере.

Технологическая универсальность предполагает возможность использования одного языка на всех

этапах обработки информации (включая ввод, контроль, сохранение данных, выборку их из хранилища, передачу данных и вывод пользователю).

MS можно применять для решения различных задач: при обучении информатике школьников и студентов, для вычислений, для разработки информационных и экспертных систем.

Сеанс работы с MS называется беседой и состоит из серии заданий (Z). Задание на языке Multi записывается в окно ввода. Результат выполнения Z, как правило, выдается в окно вывода.

В системе также имеется возможность включать "ассистентов", которые синтезируют речь на естественном языке и озвучивают работу MS.

2. Multi-сеть представления знаний в Multi Studio

На логическом уровне Multi-сеть (МС) является синтаксической сетью (СинС), на которую накладываются семантические сети (СемС). СинС состоит из синтаксических узлов и синтаксических связей (ребер) между ними в соответствии с базовым синтаксисом языка. В каждом логическом узле находится один логический объект. Синтаксические связи (отношения) между узлами (объектами) являются бинарными направленными и в исходном тексте представлены разделителями:

Открывающая скобка – "родитель→ребенок"

Пробел между объектами – "брат→брат"

Закрывающая скобка – "брат→пусто"

Часть СинС, состоящая из узла-имени и его потомков всех уровней, называется "семья" (S). Если у имени нет потомков, то S называется одиночной. В составе S могут находиться другие S. Если в числе потомков содержится имя, совпадающее с именем главы семьи, такая S называется рекурсивной. У нескольких S могут быть общие потомки.

Пример. В задании (1) X(5000) является S.

На физическом уровне основной вид памяти системы – Бункер, файл, состоящий из ячеек ("нейро"). СинС представлена в Бункере узлами-объектами, между которыми имеются ссылки. Каждый узел занимает одно нейро.

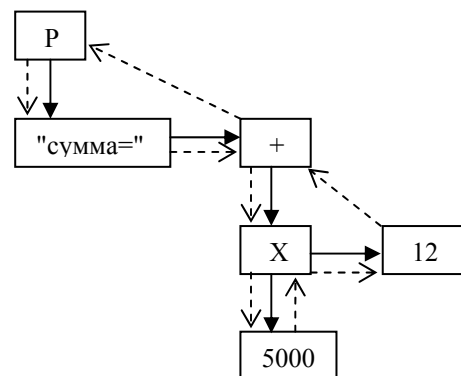


Рис.1 СинС задания (1)

Пример. Задание (1) в Бункере представляется в виде следующей СинС (рис.1).

МОЗ производит выполнение Z, используя ссылки (→) и стек. Последовательность обхода узлов Z (по принципу "сверху вниз и слева направо") указан пунктирными линиями.

Бункер имеет множество каталогов, каждый из которых представляет независимую виртуальную область Бункера – "сферу" (КС), которую также можно назвать семейным архивом, так как в каталоге сохраняются S. В каждый момент времени одна из сфер является текущей (ТС). При выполнении задания между Z и КС производится семейный S-обмен:

- Если одна сторона имеет одиночную S, а другая одноименную полную S, то потомки становятся общими.
- Если в КС нет одноименной S, то эта S помещается в КС.
- При наличии у обеих сторон одноименных полных S замена или не замена потомков S решается в зависимости от контекста.

По времени сохранения семей в каталогах сферы делятся на 3 типа:

- Каталог очищается после выполнения каждого задания.
- Каталог очищается в конце беседы.
- Каталог очищается по программе пользователя.

Семейные имена подразделяются на 2 типа: глобальные и локальные.

Соответственно S подразделяются на глобальные и локальные. Область действия глобальных имен – сфера, локальных – задание. Возможно вхождение одних S в состав других S в произвольной комбинации. Указанные свойства S позволяют использовать в сетях омонимы.

Переход из одной сферы в другую производится программно. При создании нового Бункера каталоги сфер пустые.

У каждой сферы своя СинС, на верхнем уровне которой находятся семейные имена. На одну СинС может "накладываться" множество независимых СемС. В отличие от СинС структура СемС может быть весьма разнообразной. Даже для одной СемС в общем случае возможно множество вариантов. Так в примере (1) возможна следующая эквивалентная СемС:

$P\# ("сумма =" + (123 X (5000)) ;$ (2)

Здесь P# является синтаксическим родителем команды '+'. На семантическом уровне P# является функцией, а '+' – ее аргументом. Аналогично синтаксически X является отцом числа 5000, а семантически он является подстановочным идентификатором для числа 5000.

В общем случае семантические связи между объектами представляются объектами-связями. Например, предложение «Николай является отцом Маши и мужем Зои» на Multi может выглядеть так:

Николай(являться(отец(Маши) муж(Зоя); (3)

Еще пример:

Нож(состоять_из (рукоять лезвие; (4)

В частности, семантические связи между объектами в Бункере могут быть наложены по умолчанию на синтаксические связи без наличия объектов-связок.

Дети (Иван(2000 169) Маши(1998 162); (5)

В соответствии с внешней семантикой пользователя 2000 – год рождения, 169 – рост в сантиметрах.

Язык также позволяет записать это предложение с явной пользовательской семантикой:

*Дети(Иван(2000:год_рождения 169:рост_см) (6)
Маши(1998:год_рождения 162:рост_см);*

При этом можно писать параметры детей в произвольной последовательности и даже в неполном составе:

*Дети(Иван(2000:год_рождения) (7)
Маши(162:рост_см 1998:год_рождения;*

Исходные тексты программ оформляются на Multi аналогично описанию БЗ и точно также записываются в Бункер.

Таким образом, в Multi Studio моделью представления знаний является универсальная Multi-сеть.

В составе Multi-сети могут находиться фреймы и производные правила.

Используя команду ## [структурный вывод] пользователь может распечатать структурно (по уровням) всю сеть объектов заданной сферы или любой ее части (если он имеет доступ к ним). При этом информация выводится в синтаксисе Multi, что позволяет читать программы и данные (и обмениваться ими) при отсутствии исходных текстов.

Наличие в КС S позволяет вызывать из Бункера на исполнение поименованные программы или их части.

Так если поименовать первое предложение примера (1) как ZZ и выполнить задание

$ZZ(P\# ("сумма =" + (X 123 ;$ (8)

то в следующем задании можно запустить программу-семью ZZ:

$\{ (ZZ;$ (9)

Можно также запустить ZZ с заменой X:

$\{ (ZZ; X(20.55;$ (10)

В системе MS применяется словарь команд.

Пользователь может менять некоторые параметры словаря: заменять обозначения команд или вводить дополнительные обозначения – синонимы. Например, команде '+' можно назначить русский синоним 'сумма' или английский 'add'. Таким образом, в Multi-сети могут находиться

семантические синонимы, синтаксически обозначаемые различно.

Большинство команд Multi имеют глубокую контекстность (область действия) и вкладываются друг в друга, что существенно упрощает, сокращает Multi-текст и увеличивает быстрдействие системы.

3. Разработка экспертных систем в среде Multi Studio

Имеющийся набор команд языка Multi позволяет весьма просто разрабатывать в среде Multi Studio статические экспертные системы (ЭС). Для этого помимо общих команд (например, арифметических) имеется несколько специальных команд.

Основная команда называется «меню» и имеет следующую структуру:

*menu.(заголовок” вариант1 (аргум1)
вариант2 (аргум2) ...)* (11)

Команда меню открывает окно меню (рис.2), в котором указывается заголовок, поля вариантов и 1 или 2 дополнительных поля. Заголовок может быть наименованием проблемы или вопросом.

Поле варианта может быть разделом проблемы (ответом на вопрос в заголовке) или наименованием справки. Наименование справки – это литерал, первым символом которого является знак '!'. Поле других вариантов можно задавать именем, числом, текстом, символами + - = >= < <= !=. Символы + - могут интерпретироваться как «ДА» и «НЕТ».

При выборе варианта происходит переход на аргум соответствующего варианта. В частности, аргум может содержать переход на блок команд, в котором содержится новое меню.

Если выбрано поле справки, то после окончания работы аргума происходит автоматический возврат на текущее меню.

Дополнительные поля: «назад» и «конец». Поле «назад» открывает предыдущее меню (если оно имеется). Поле «конец» открывает дополнительное меню, в котором имеется несколько вариантов возврата, вплоть до выхода из экспертной системы.

Пример фрагмента ЭС поиска неисправностей двигателя:

*R#(«ЭКСПЕРТНАЯ СИСТЕМА
ПОИСКА НЕИСПРАВНОСТЕЙ ДВИГАТЕЛЯ»
Состояние(menu.(“Состояние двигателя?” (12)
“НОРМА” («хорошо» Решение1)
“НЕ ЗАВОДИТСЯ” (Вал)
«ПЕРЕБОИ» (Мощность) ;
Решение1(menu.rgb(“255..” “255..”
типи.(«РЕШЕНИЕ 1 :»
«РЕМОНТ НЕ НУЖЕН»;
Мощность(menu. (“Мощность двигателя?”
“ВЫСОКАЯ” (инерция)
“НИЗКАЯ” (контакты) ;*

При запуске этого задания на экране системы появляется диалоговое окно следующего вида (поля «назад» нет, так как меню первого уровня.)

При выборе варианта Пользователь может "кликнуть" на нужное поле выбора или нажать на

клавиатуре цифру – порядковый номер поля. Номера имеют только первые 9 полей (1-9). После выбора ответа команда запускает соответствующее действие.

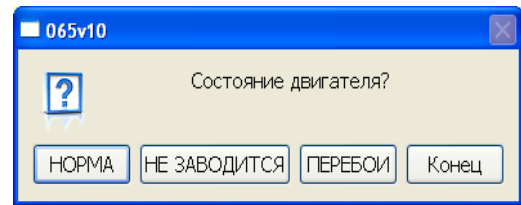


Рисунок 2. Пример меню первого уровня

При выборе варианта 1 (НОРМА), открывается новое меню «РЕШЕНИЕ 1 :», в котором тексты вопроса и первого (и единственного) ответа – красные. После выбора ответа Пользователем, система завершает работу с ЭС.

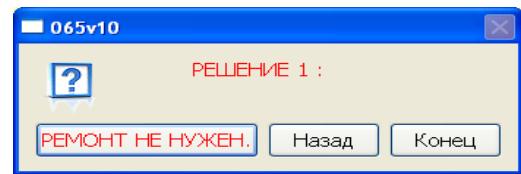


Рисунок 3. Пример цветного меню

При выборе варианта 3 (ПЕРЕБОИ) появляется окно с меню «Мощность двигателя?» и так далее до получения очередного решения (рекомендации) системы.

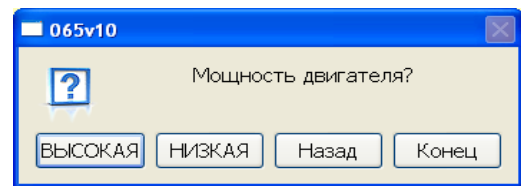


Рисунок 4. Пример меню второго уровня

Если несколько вариантов требуют одни и те же действия, их можно объединить в один составной текст-литерал, в котором тексты вариантов разделяются вертикальной чертой '|’.

*menu. (“цвет яблока?” (13)
“красный|желтый”(“ешь”)
“серый|с пятнами”(“выкинь”);*

Пример справки: (14)
*menu. («Куда пойдешь?» «Налево» «Направо»
«! Справка»(«Налево пойдешь – коня потеряешь.
Направо пойдешь – Змея-Горыныча
повстречаешь.») ;*

Максимально возможное количество вариантов в одном меню – 40. В общем случае они располагаются в виде матрицы G (горизонталь) x V (вертикаль). По умолчанию в каждой строке по горизонтали размещается 6 вариантов. Дополнительная команда menu.g позволяет изменять размер G от 1 до 10.

К дополнительным командам относятся теги – это группа команд редактирования текста заголовка, вариантов (а также текста протокола работы ЭС). Например, окраска текста заголовка и

вариантов регулируется командой-тегом menu.rgb (“К.З.С”), где К указывает насыщенность красного (0-255) в общем цвете, З – зеленого и С – синего. Окраска текста протокола регулируется аналогичной командой rgb.

Теги могут располагаться в тексте задания:

- до команды «меню»
- в аргуме меню
- или отсутствовать.

В последнем случае цвета, размеры и другие свойства текстов назначаются по умолчанию.

Команды menu.sv и menu.nv позволяют запоминать текст и/или номер варианта с заданными именами, чтобы использовать их далее в текущем или следующих заданиях.

В процессе работы с окнами меню может формироваться протокол, текст которого в режиме реального времени выводится в окно вывода. Пользователь имеет возможность следить за ходом логического вывода решения и своевременно исправлять допущенные ошибки путем возврата на предыдущие меню. В конце работы протокол может выводиться на печать. Имеется 3 режима протоколирования (полное, частичное и блокировка).

Пример протокола запуска фрагмента ЭС (12):

```

«ЭКСПЕРТНАЯ СИСТЕМА
  ПОИСКА НЕИСПРАВНОСТЕЙ
  ДВИГАТЕЛЯ
  Состояние двигателя? ПЕРЕБОИ
  Мощность двигателя? Назад
  ВОЗВРАТ на
  предыдущий вопрос
  Состояние двигателя? НОРМА
  хорошо
РЕШЕНИЕ 1 : РЕМОНТ НЕ НУЖЕН.»

```

ЭС в Multi Studio – это сеть команд menu. и их составляющих. В приведенных примерах формируется ЭС с прямым логическим выводом. Команда меню позволяет также формировать ЭС с обратным логическим выводом.

4. Сравнение системы Multi Studio и среды CLIPS

Для сравнения Multi Studio с другими средами была выбрана система CLIPS по причине ее доступности. К тому же эта среда является базовой в российском вузовском образовании по ЭС [Сидоркина, 2011, Частиков и др., 2003].

Сравнение систем проводилось на примере тестовой ЭС поиска неисправностей двигателя, заимствованного у разработчиков системы CLIPS [Джарратано, 2007].

Пример: фрагменты экспертной системы поиска неисправностей на языке CLIPS:

```

(deffunction ask-question (?question $?allowed-values)
  (printout t ?question)
  (bind ?answer (read))

```

```

  (if (lexemep ?answer)
      then (bind ?answer (lowercase ?answer)))
  (while (not (member ?answer ?allowed-values))
    (printout t ?question)
    (bind ?answer (read))
    (if (lexemep ?answer)
        then (bind ?answer (lowercase ?answer))))
  ?answer )
  (defrule determine-engine-state ""
    (not (working-state engine ?))
    (not (repair ?))
    =>
    (if (yes-or-no-p "Does the engine start (yes/no)? ")
        then
        (if (yes-or-no-p "Does the engine run normally
        (yes/no)? ")
            then
            (assert (working-state engine normal))
            else
            (assert (working-state engine unsatisfactory)))
            else
            (assert (working-state engine does-not-start)))
        )

```

Предварительные результаты сравнения систем в части разработки статических ЭС:

- Система Multi Studio – универсальна (на ней кроме разработки ЭС возможно проводить другие вычислительные работы). CLIPS – специализированная оболочка ЭС.
- Диалог в Multi оконный с параллельным протоколированием каждого шага логического вывода (по усмотрению Пользователя). В CLIPS совмещено то и другое в формате командной строки.
- В CLIPS-программе нет возвратов к предыдущим меню.
- Языком интерфейса является русский (возможен английский и другие), на CLIPS – кириллица не поддерживается.
- В Multi Studio можно модифицировать программу «на ходу», заменяя отдельные именованные части, не перекомпилируя всю программу.
- Скорость работы Multi-программы значительно выше за счет того, что программа ходит по готовым путям семантической сети. CLIPS-программа по классическому алгоритму многократно обходит базу знаний в последовательном поиске нужных правил.
- Текст программы на Multi в 4 раза короче.
- в Multi-программе используются пять операторов, вместо тридцати в системе CLIPS.
- Трудоемкость разработки статической ЭС в среде CLIPS на порядок выше, чем в Multi Studio.
- Multi-программы после короткого обучения может разрабатывать эксперт, не прибегая к услугам программиста. CLIPS -программы может писать только программист.

ЗАКЛЮЧЕНИЕ

В докладе предложен вариант решения языковой проблемы разработки статических экспертных систем с использованием командного языка Multi в среде Multi Studio. Система Multi Studio развивается. Предполагается включение в ее состав группы команд, позволяющих моделировать работу устройств и систем, что позволит уменьшить трудоемкость разработки динамических экспертных систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Голенков, 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): материалы Междунар. научн-техн. конф. (Минск, 10–12 февраля 2011 г.); – Минск: БГУИР, 2011, С. 21–58.

[Джарратано, 2007] Джарратано, Джозеф. Экспертные системы: принципы разработки и программирование, 4-е издание. Пер. с англ. / Джарратано, Джозеф, Райли, Гари; – М.: ООО "И.Д. Вильямс", 2007.

[Сидоркина, 2011] Сидоркина, И.Г. Системы искусственного интеллекта: учебное пособие / И. Г. Сидоркина; – М.: КНОРУС, 2011.

[Частиков и др., 2003] Разработка экспертных систем. Среда CLIPS / Частиков А.П. [и др.]; - СПб.: БХВ-Петербург, 2003.

EXPERT SYSTEMS DEVELOPMENT IN THE MULTI STUDIO ENVIRONMENT

Kataev V.A.

*LtD "Perm Scientific Industrial Instrument-Making
Company", Perm, Russian Federation*
Bravo55555@yandex.ru

This paper includes short summary of static expert systems development in Multi Studio (MS) Environment. Multi Studio is Russian development. Environment is based on universal language Multi and semantic Multi-networks. Multi-network is combination (symbiosis) of syntactic and semantic networks. Results of the comparison Multi Studio with known environment CLIPS are presented.

Key words: CLIPS, expert system, Multi Studio, semantic network.

INTRODUCTION

Expert systems (ES) belong to class of intelligence systems (IS). IS designing is labor intensive and long. We consider that the main line of IS development is IS designing language improvement and development of effective environment of this language implementation.

Multi Studio system attempts to eliminate this problem by implementing a simple universal high-level language Multi.

Multi main properties are: high-level (Multi language is close to natural language), simplicity, universality, modularity, context using, ability to

manipulate network structures objects (network structures are the most universal).

MAIN PART

The main part contains:

1. Description of Multi Studio Environment. MS is a universal multiple-windows interactive Multi language interpreter. Multi language is a high-level command language. Multi belongs to the functional paradigm and has prefix notation.

2. The MS knowledge representation model. System memory ("electronic brain") consists of a cells ("neuro") set. Each "neuro" contains an object (e.g., a number) and its syntax connection to other "neuros". In general, "neuro" is semantic network node. Knowledge representation frame model and production model is a special case of a semantic network. Knowledge bases and programs have similar description in Multi language. They are in the same network.

3. Expert systems development in the Multi Studio Environment. Set of language commands enables to develop in MS Environment static ES. The core command is called "menu" and has the structure: menu("Question" "answer 1" (action 1) "answer 2" (action 2) ...). The command opens a window, which shows the question, answers (basic buttons) and two additional buttons: the "back" and the "end". When you click on the basic button, the relevant action is executed, e.g. moving to the next command "menu". The "back" returns to the previous menu command. The "end" finishes ES.

ES in Multi Studio is a network of commands "menu" and its components. The command enables forward and backward chaining.

4. Comparison of Multi Studio and CLIPS. Comparison of MS and CLIPS was performed by the example of test ES engine troubleshooting. This ES was taken from developers of system CLIPS.

Results:

The program text on Multi is four times shorter than the program text on CLIPS.

The Multi program uses five operators instead of thirty in the CLIPS program.

The program in CLIPS doesn't have returns to previous questions.

ES development in CLIPS is times more difficult than in Multi Studio.

Multi Studio system is a universal system. It enables other calculations. CLIPS is a specialized system for ES development.

CONCLUSION

Multi Studio system enables to reduce static ES development time and complicity. The additional group of commands will be included in Multi language. This enables simulating devices and systems work and developing dynamic ES.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОНТОЛОГИЧЕСКАЯ ПАРАДИГМА ПРОГРАММИРОВАНИЯ

Грибова В.В., Клещев А.С.

*Учреждение Российской академии наук Институт автоматизации и процессов управления
Дальневосточного отделения РАН (ИАПУ ДВО РАН), Владивосток, Россия*

`gribova@iacp.dvo.ru,`

`kleshev@iacp.dvo.ru`

Данная работа посвящена описанию новой парадигмы программирования – онтологической, которая является дальнейшим развитием парадигмы декларативного программирования. В работе описана постановка задачи, модель данных, лежащая в основе парадигмы, основные конструкции языка.

Ключевые слова: парадигмы программирования, модель данных, семантические сети, онтология.

ВВЕДЕНИЕ

Трудоемкость разработки программных систем сопоставима (а порой и превышает) сложность разработки крупных материальных объектов [Обзор Microsoft, 2005]. При этом сопровождение таких систем является еще более дорогим процессом [Промыслов и др., 2005], что требует от исследователей поиска новых подходов, направленных на решение этих проблем. В качестве одного из таких подходов была предложена парадигма декларативного программирования и разработанные в ее рамках языки декларативного программирования.

Программа на языке декларативного программирования в общем случае представляет собой описание абстрактной модели решаемой задачи или, согласно [Дехтяренко, 2003], исполнимую спецификацию результата вычислений по программе. Программисту не требуется описывать процесс управления вычислениями, он полностью возлагается на процессор языка. Среди множества преимуществ парадигмы декларативного программирования, в качестве основных называются: более простое написание программы, более легкое ее понимание по сравнению с соответствующими императивными программами, упрощение ее сопровождения и модификации [Lloyd, 1994].

К языкам декларативного программирования относят, как правило, языки функционального и логического программирования. Однако, как отмечено в [Lloyd, 1994], провозглашенная идея декларативного программирования в современных языках логического и функционального

программирования в полной мере не реализована; в результате практические программы на Прологе или Лиспе ненамного легче разрабатывать, понимать и сопровождать, чем программы на императивных языках. Среди других недостатков языков декларативного программирования называют скудные средства взаимодействия с пользователем и сложность внесения в программы императивных операторов, если их использование более удобно по сравнению с декларативными конструкциями.

Цель данной работы – предложить новую парадигму программирования, названную парадигмой онтологического программирования, как дальнейшее развитие парадигмы декларативного программирования, а также предложить язык программирования OPL (Ontological Programming Language), удовлетворяющий определению декларативного программирования, в котором программа является онтологией результата вычислений; для этого языка также вводятся механизмы организации интерфейса с пользователем и включения в язык императивных конструкций.

1. Постановка задачи

При решении любой задачи на компьютере результат может быть получен лишь как итог некоторого процесса вычислений, который будем называть процессом получения результата. Сложность написания программы для решения некоторой задачи в значительной степени состоит в том, что разработчик этой программы должен представить себе все множество процессов получения результатов решения задачи (экстенционал задачи) при различных допустимых исходных данных и определить это множество

средствами языка программирования в виде программы [Успенский и др., 1987]. Сложность понимания программы сопровождающим программистом в значительной степени состоит в том, что этот программист должен по программе представить себе экстенционал задачи и понять, почему процессы получения результата ведут именно к результату решения этой задачи. Сложность модификации программы в значительной степени состоит в том, что сопровождающий программист, понимая, как должны измениться результаты решения задачи, должен понять, как должны измениться процессы получения результатов, чтобы они приводили именно к этим изменениям результатов, а затем понять, как эти изменения в процессах получения результатов могут быть выражены через изменения в программе. Рассмотрим с этой точки зрения, как связаны процессы получения результата и программы с результатами вычислений в парадигмах императивного, функционального и логического программирования.

В основе парадигмы императивного программирования лежат вычислительные модели [Себеста, 2001, Floyd, 1979]. В этих моделях процесс получения результата представляет собой последовательность состояний, первое из которых получается из исходных данных с помощью входной процедуры, каждое следующее состояние получается из предыдущего с помощью оператора непосредственной переработки, а из последнего состояния извлекается результат вычислений с помощью выходной процедуры. Все состояния вычислительного процесса, кроме последнего, лишь косвенно связаны с результатом вычислений (решения задачи). В современных императивных языках состоянием вычислительного процесса является множество значений переменных, а оператором непосредственной переработки – оператор присваивания, с помощью которого следующее состояние процесса получается изменением значения одной переменной, а все остальные переменные сохраняют свои значения. В программе на императивном языке последовательности выполнения операторов присваивания определяются с помощью линейных участков, условных операторов и операторов цикла (а также вызовов процедур).

В основе парадигмы функционального программирования лежит лямбда-исчисление [Себеста, 2001, Floyd, 1979]. Процесс получения результата может быть представлен ориентированной размеченной сетью вызова функций, в которой меткой каждой терминальной вершины является некоторое исходное данное, а меткой каждой нетерминальной вершины – значение некоторой функции, аргументами которой являются метки концов дуг, выходящих из этой вершины (ориентация дуг – от результата к аргументу). Результатом вычислений является метка корня сети (вершины, в которую не входит ни одна дуга). Все промежуточные значения (метки

нетерминальных вершин), кроме метки корня, лишь косвенным образом связаны с результатом вычислений. В современных функциональных языках имеется некоторые наборы базовых функций, а также средства определения (конструирования) новых функций из базовых и уже определенных (в том числе условные термы и рекурсия).

В основе парадигмы логического программирования лежит исчисление предикатов первого порядка [Себеста, 2001, Floyd, 1979]. Процесс получения результата может быть представлен ориентированной размеченной сетью вывода результата, в которой меткой каждой терминальной вершины является кортеж отношения, представляющий некоторое исходное данное, а меткой каждой нетерминальной вершины – кортеж отношения, являющийся результатом применения некоторого правила к посылкам, которыми суть метки концов дуг, выходящих из этой вершины (ориентация дуг – от заключения к посылкам). Результатом вычислений (вывода) является метка корня сети. Все промежуточные значения (метки нетерминальных вершин), кроме метки корня, лишь косвенным образом связаны с результатом вычислений. В современных логических языках программа представляет собой некоторый запрос и множество правил (импликаций) и фактов (кортежей отношений).

В каждой из трех рассмотренных парадигм результат вычислений получается лишь на последнем шаге процесса получения результата, а все остальные шаги этого процесса имеют к результату вычислений лишь косвенное отношение, на них получаются лишь некоторые промежуточные значения. Такие процессы получения результата можно назвать косвенными по отношению к результату.

Таким образом, для снижения сложности разработки программ, их понимания и модификации необходимо предложить такую парадигму программирования, в которой процессы получения структурных результатов вычислений, представляемых в виде ориентированных графов, были бы прямыми, т.е. состояли из шагов, на каждом из которых формируется некоторая часть результата. В этом случае, сам такой процесс получения результата может быть представлен в виде графа, совпадающего с этим результатом, а значит, программа на языке является (исполнимой) спецификацией множества результатов вычислений (а не множества косвенных процессов их получения). Разрабатывая такую программу, программист должен лишь специфицировать множество результатов вычислений; анализируя такую программу, программист должен представить себе лишь множество результатов вычислений, специфицируемых этой программой; модифицируя такую программу, программист должен лишь понять, как следует изменить спецификацию множества результатов вычислений, чтобы получить

требуемые изменения этих результатов, причем каждое такое изменение в программе является локальным, связанным с изменением соответствующих частей результатов вычислений. Такая спецификация множества результатов вычислений, в соответствие с современными представлениями в области искусственного интеллекта, называется онтологией результата вычислений. Учитывая это, назовем предлагаемую парадигму *парадигмой онтологического программирования*. Ее можно рассматривать в качестве более полного воплощения парадигмы декларативного программирования по сравнению с парадигмами функционального и логического программирования.

2. Модель данных, лежащая в основе парадигмы

Одно из положений парадигмы онтологического программирования состоит в том, что все данные имеют вид семантических сетей. Для определенности рассматриваются иерархические семантические сети (что не является принципиальным), определяемые следующим образом. Иерархическая семантическая сеть есть связный ориентированный граф без циклов, в котором каждая дуга имеет метку, а вершины могут быть одного из двух типов - простые и структурные; вершина сети, в которую не входит ни одна дуга, называется ее корнем. Каждая простая терминальная вершина сети имеет в качестве метки константу некоторого сорта. Корень сети имеет две метки-термина, из которых первая метка есть метка класса (имя функции, в результате выполнения которой была создана эта семантическая сеть), а вторая - индивидуальная метка этой сети. Каждая структурная вершина сети является контейнером, содержащим упорядоченное конечное множество иерархических семантических сетей с одной и той же меткой класса и попарно различными индивидуальными метками.

Семантическая сеть может быть постоянного хранения (храняемая независимо от приложения) для обработки всеми приложениями, реализованными в рамках парадигмы онтологического программирования, так и временная, создаваемая при запуске приложения и прекращающая свое существование по окончании этого запуска.

Использование семантических сетей в качестве модели данных означает, что обрабатываются только целостные информационные структуры, которые выделяются к качестве объектов обработки, в отличие от других парадигм, поддерживающих обработку информационных объектов, хотя и произвольной структуры, но не обязательно объединенных в целостную информационную структуру (как правило, целостные информационные структуры разделяются на различные фрагменты, являющиеся значениями различных переменных, связь между которыми известна и понятна только разработчику

программы). Например, в экспертной системе медицинской диагностики такими целостными информационными структурами являются история болезни, база знаний и объяснение, представленные в форме семантических сетей.

3. Приложение

В общем случае приложение имеет одну или несколько выходных семантических сетей (результатов), может иметь (или не иметь) одну или несколько входных семантических сетей (входных данных), а также иметь (или не иметь) промежуточные семантические сети (промежуточные результаты). Каждая промежуточная или выходная семантическая сеть приложения вычисляется с помощью функции (название которой является меткой класса корня этой семантической сети).

Каждая функция может иметь (или не иметь) входные семантические сети, в точности одну выходную семантическую сеть и не может иметь промежуточных семантических сетей

Входные семантические сети, как приложения, так и функций, могут быть постоянного хранения (созданными и существующими независимо от приложения), так и временными созданными во время запуска приложения (как результат одной из его функций). Соответственно, выходные семантические сети, как приложения, так и функций, могут быть как постоянного хранения, так временными.

Между всеми семантическими сетями приложения (входными, промежуточными и выходными) задается частичный порядок, в котором вычисляются промежуточные или выходные семантические сети, как результаты соответствующих функций. Поэтому программу приложения можно рассматривать как суперпозицию этих функций, связанных с вычислением всех семантических сетей приложения, не являющихся входными семантическими сетями постоянного хранения.

Таким образом, выполнение приложения состоит в вычислении выходных семантических сетей по входным семантическим сетям постоянного хранения, созданным до запуска приложения (их может и не быть) через промежуточные семантические сети, создаваемые при запуске приложения.

Каждая функция приложения представляет собой онтологию (описание структуры) выходной семантической сети этой функции на языке OPL.

4. Язык онтологического программирования

Язык OPL является визуальным логическим языком программирования. Описание функции – онтология выходной семантической сети,

представляет собой ориентированный граф с размеченными вершинами и дугами. Метками дуг являются термины, которые в ходе выполнения программы становятся метками дуг выходной семантической сети функции. Метками вершин являются логические формулы.

Процесс вычисления выходной семантической сети функции сводится к построению этой сети, начиная с корня. В процессе построения семантической сети устанавливается однозначное соответствие между вершинами и дугами этой сети и вершинами и дугами ее онтологии (описанию функции). Выходная семантическая сеть, в зависимости от описания функции, может формироваться как автоматически, без участия пользователя, так и интерактивно, с участием пользователя, который в процессе диалога управляет процессом построения этой сети.

Логическими формулами языка являются: простая формула, простая кванторная формула, унарная формула, пропозициональная формула, структурная кванторная формула, множество импликаций.

Простая формула есть граф, состоящий из единственной вершины с меткой s . Метка s является константой одного из сортов, либо переменной v , либо значением переменной v^* .

Унарная формула есть граф, состоящий из начальной вершины и дуги с меткой T (термином), выходящей из начальной вершины и входящей в начальную вершину некоторой логической формулы F .

Пропозициональная формула есть граф, состоящий из начальной вершины с пропозициональной меткой P и выходящих из нее n дуг (не менее двух), каждая из которых имеет метку T_i (термин; i от 1 до n и все эти метки должны быть попарно различны) и входит в начальную вершину некоторой логической формулы F_i . Пропозициональными метками P являются: $\&$ — конъюнкция, \vee — дизъюнкция, $|$ — исключающее или. Множество пропозициональных меток должно быть расширяемым.

Простая кванторная формула есть граф, состоящий из единственной вершины с меткой QMT , где Q — знак квантора, M — описание множества, а T — термин. Знаком квантора может быть: \forall (для всех), \exists (существует), $\exists 2$ (существует не менее двух), $\exists ?$ (существует, но не для всех), $\exists !$ (существует и единственен), $\exists []$ (существует подынтервал).

Описание множества может явно задаваться либо значениями своих элементов, которыми являются попарно различные константы, либо определяться сортом возможных значений элементов множества, либо целым или вещественным интервалом. В качестве описания множества также может выступать переменная. Названием сорта может быть: "строка", "целый",

"вещественный", "целый интервал", "вещественный интервал", "дата-время". Множество сортов и кванторов является расширяемым.

В зависимости от описания множества возможны различные кванторы, предшествующие его описанию. Описание множества и кванторы задают условия и ограничения при порождении очередного фрагмента результата.

Структурная кванторная формула состоит из начальной вершины с меткой, имеющей вид QMT , где Q — знак квантора, M — описание множества, а T — термин, и логической формулы F , начальная вершина которой изображается внутри начальной вершины структурной кванторной формулы

Множество импликаций есть граф, состоящий из одной вершины с меткой $\{P_1 \Rightarrow F_1, \dots, P_m \Rightarrow F_m\}$, где P_1, \dots, P_m — антецеденты, а F_1, \dots, F_m — консеквенты импликаций. Антецедент импликации есть конечное множество компонент — логических формул, каждая из которых может иметь префикс. Префикс есть название некоторой программы на языке. Консеквент импликации есть логическая формула.

Переменные, которые объявлены в логических формулах, могут входить только в антецеденты и консеквенты импликаций, а специальные метки вершин в унарных формулах — только в антецеденты импликаций. Если переменная входит в консеквент импликации, то она должна входить и в ее антецедент.

Переменные используются в импликации, если очередной фрагмент результата должен быть сформирован в зависимости от значений других меток (терминов) уже сформированных фрагментов графа, либо внешних по отношению к данному (таким образом могут использоваться внешние данные и эти данные также представлены в форме графа.). В этом случае в качестве метки логической формулы либо описания множества (зависит от типа логической формулы) выступает переменная. При этом если при формировании графа используются внешние данные по отношению к данному графу, то эта формула содержит префикс, совпадающий с именем файла, из которого будут вычисляться значения переменных.

В общем случае при порождении результата вычислений, когда активной вершине является множество импликаций, то вычисляется истинность антецедентов этих импликаций. Если у некоторой импликации истинен ее антецедент, то активной вершине соответствует начальная вершина консеквента этой импликации.

Если в антецеденте импликации в логических формулах присутствует переменная, то при порождении вместо имени переменной осуществляется подстановка ее значений. Импликация будет выполняться если при подстановке некоторого значения переменной все компоненты антецедента истинны.

Для организации сложных эффективных вычислений средств логического языка может оказаться недостаточно. Для решения этой проблемы язык OPL дополнен вычисляемыми предикатами. Вычисляемые предикаты предназначены для описания вычислительных алгоритмов, описание которых средствами декларативного логического языка слишком громоздко. Сами вычисляемые предикаты описываются средствами императивного языка в соответствующей среде создания приложений. Вычисляемый предикат имеет имя и множество формальных параметров. Вызов вычисляемого предиката с фактическими параметрами может быть лишь компонентой антецедента импликации. При поиске по образцу, когда в антецеденте импликации встречается вызов вычисляемого предиката, происходит его выполнение: в тело предиката подставляются значения, указанные в его аргументах, и производятся вычисления.

Команды абстрактного интерфейса определяют функциональность пользовательского интерфейса. Их можно разделить на четыре класса:

- команда вывода (процессором языка) сформированного фрагмента семантической сети (выполняются при построении семантической сети в интерактивном режиме или по запросу пользователя);

- команды непосредственного ввода с клавиатуры одного значения заданного типа и изменение этого значения (входит в операционную семантику кванторных формул с квантором существования и единственности, если множество, по которому пробегает квантор, бесконечно);

- команды непосредственного ввода с клавиатуры нескольких значений заданного типа и изменения этих значений (входит в операционную семантику кванторных формул с различными вариантами квантора существования, если множество, по которому пробегает квантор, бесконечно);

- команды выбора значения либо подмножества значений из предлагаемого множеств и изменение этого выбора (входит в операционную семантику дизъюнкции и кванторных формул с различными вариантами квантора существования, если множество, по которому пробегает квантор, конечно).

Команды абстрактного интерфейса ввода и выбора значений совпадают для кванторных формул с различными вариантами квантора существования, если квантор пробегает конечное множество, и пропозициональных формул дизъюнкции и исключающего или.

С использованием языка описана экспертная система медицинской диагностики. Входными данными для нее являются «Простая база наблюдений» и «История болезни». Само приложение (программа на языке) представляет

собой семантическую сеть «Объяснение» (формирование объяснения). Объяснение состоит из двух частей – объяснения гипотезы о том, что пациент здоров, и объяснения гипотез о том, что пациент болен одним из заболеваний, представленных в базе заболеваний (см. рис. 1).

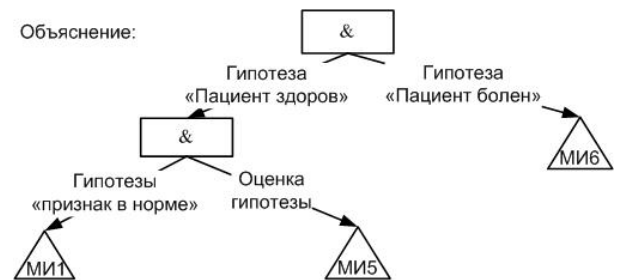


Рисунок 1 – Объяснение экспертной системы медицинской диагностики

Объяснение гипотезы о том, что пациент здоров, состоит из объяснения гипотез о том, что все наблюдаемые признаки пациента находятся в норме, и из оценки гипотезы о том, что пациент здоров.

Импликация MI1 (рис.2) формирует значение переменной $v1$, которым является множество всех наблюдаемых признаков в истории болезни; для каждого из этих признаков объяснение гипотезы о том, что он находится в норме, состоит из объяснения гипотез о том, что все наблюдавшиеся значения этого признака являются нормальными, и из оценки гипотезы о том, что этот признак находится в норме.



Рисунок 2 – Описание гипотезы "признак в норме"

Импликация MI2 (рис. 3) для каждого значения $v1^*$ переменной $v1$ формирует значение переменной $v2$, которым является множество моментов наблюдения признака $v1^*$ в истории болезни; для каждого из этих моментов наблюдения формируется оценка гипотезы о том, что значение признака $v1^*$, наблюдавшееся в этот момент, является нормальным.

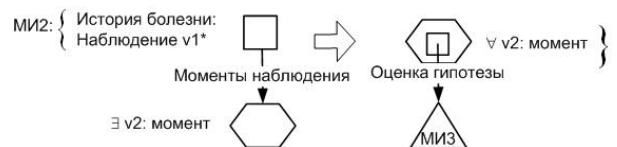


Рисунок 3 – Описание гипотезы "значение нормальное"

Первая импликация (рис. 4) из множества MI3 для каждого значения $v1^*$ переменной $v1$ и для каждого значения $v2^*$ переменной $v2$ формирует значение переменной $v3$, которым является значение, наблюдавшееся в момент $v2^*$ у признака $v1^*$ в истории болезни, а также формирует значение переменной $v4$, которым является множество

нормальных значений признака $v1^*$ в базе заболеваний, а затем выполняется вычисляемый предикат «принадлежность» (принадлежность элемента $v3$ множеству $v4$); если значение этого предиката есть «истина», то формируется оценка «подтверждена» гипотезы о том, что значение признака $v1^*$, наблюдавшееся в момент $v2^*$, является нормальным. Вторая импликация, если значение этого предиката есть «ложь», формирует оценку «опровергнута» гипотезы о том, что значение признака $v1^*$, наблюдавшееся в момент $v2^*$, является нормальным.

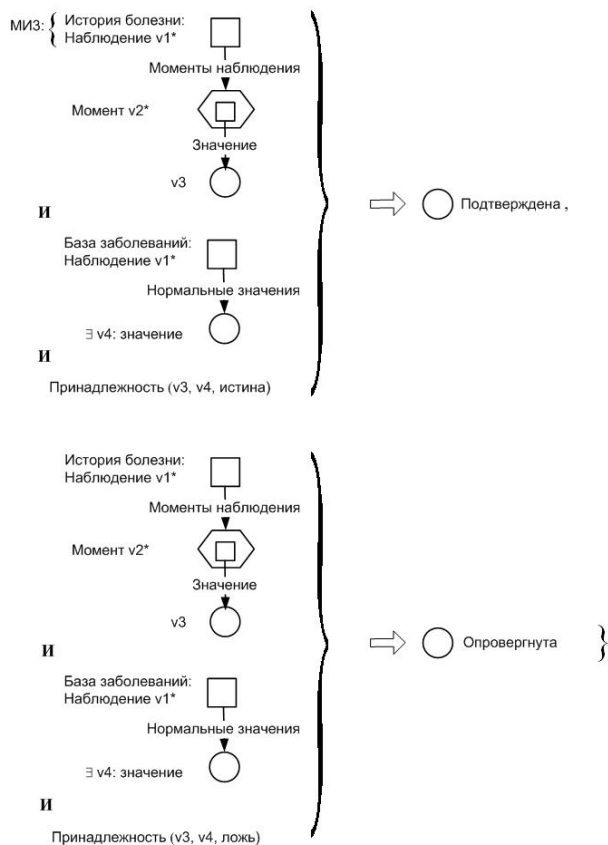


Рисунок 4 – Описание оценки гипотезы "значение нормальное"

Первая импликация (Рис. 5) из множества МИ4 для значения $v1^*$ переменной $v1$ формирует оценку «подтверждена» гипотезы о том, что признак $v1^*$ находится в норме, если в объяснении оценки гипотез о том, что значения признака $v1^*$, наблюдавшееся в моменты, равные всем значениям переменной $v2$, являются нормальными, «подтверждена». Вторая импликация, для значения $v1^*$ переменной $v1$ формирует оценку «опровергнута» гипотезы о том, что признак $v1^*$ находится в норме, если в объяснении оценки гипотез о том, что значения признака $v1^*$, наблюдавшееся в моменты, равные некоторым элементам множества - значения переменной $v2$, являются нормальными, «опровергнута».

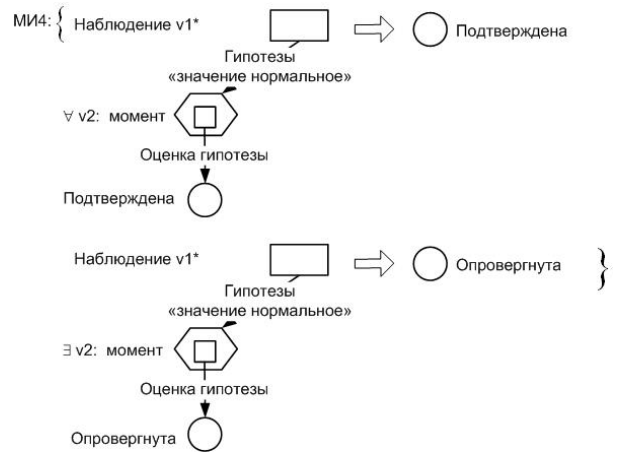


Рисунок 5 – Описание оценки гипотезы "признак в норме"

Первая импликация из множества МИ5 (рис. 6) формирует оценку «подтверждена» гипотезы о том, что пациент здоров, если в объяснении оценка гипотезы о том, что признаки, равные всем значениям переменной $v1$, есть «подтверждена». Вторая импликация формирует оценку «опровергнута» гипотезы о том, что пациент здоров, если в объяснении оценка гипотезы о том, что признаки, равные некоторым элементам множества - значения переменной $v1$, есть «опровергнута».

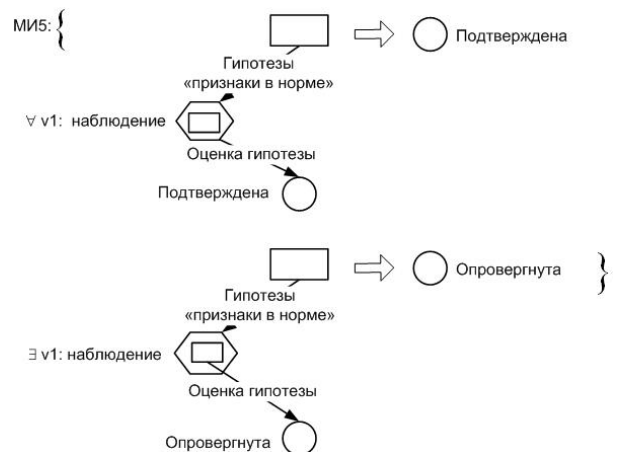


Рисунок 6 – Оценка гипотезы "пациент здоров"

Гипотеза «пациент болен» описывается аналогичным образом.

5. Приложение

Основные задачи, которые были поставлены при разработке новой онтологической парадигмы программирования – это снижение сложности разработки и сопровождения программ. Анализ существующих парадигм и языков программирования показал, что основными сложностями, с которыми сталкиваются программисты, разрабатывающие программы и программисты, которые их сопровождают, являются:

- сложность программисту, разрабатывающему программу, понять, формализовать и описать

средствами языка программирования множество процессов получения результатов решения задачи при различных исходных данных; эта сложность имеет место, независимо от того, что представляет собой этот процесс – последовательность состояний, сеть вызова функций либо сеть вывода результата;

- сложность программисту, сопровождающему программу, понять, как устроено описанное программистом множество решений, а в случае, если его необходимо изменить, понять, как внести все необходимые изменения в программу;

- сложность программисту реализовать пользовательский интерфейс, обеспечивающий ввод информации от пользователя и вывод результатов и встроить его в процесс получения результата; в результате программа становится еще более сложной для понимания и сопровождения, а изменение процесса вычислений приводит к необходимости внесения изменения в интерфейс и наоборот.

Парадигма онтологического программирования предлагает следующие решения указанных выше проблем.

1. Использование семантических сетей в качестве модели данных означает, что обрабатываются только целостные информационные структуры. Это упрощает сопровождение приложений – программисту легче понять целостную информационную структуру, нежели множество разрозненных структур, связь между которыми известна и понятна только разработчику программы.

2. Приложение в рамках предложенной парадигмы может быть представлено как частично упорядоченное множество вызовов функций; каждая функция формирует в качестве результата и, соответственно, обрабатывает в качестве входных данных только целостные информационные структуры, которых, как правило, с приложением связано не слишком много. Соответственно, основная «нагрузка» ложится на описание каждой функции. Описание функции представляет собой онтологию результата функции. Таким образом, при разработке приложения необходимо: выделить целостные информационные структуры, определить частичный порядок между задачами их формирования; описать функции для вычисления каждой такой информационной структуры. Сопровождение приложения сводится к добавлению новых целостных информационных структур и функций, изменению частичного порядка вызовов функций приложения и сопровождению самих функций - онтологий их результатов. Описание онтологии результата, как правило, более компактно и наглядно, чем описание процесса его получения.

3. Онтология ансамбля семантических сетей представляется на визуальном языке в виде ориентированного графа. Графовое (визуальное)

представление согласно многим исследованиям, например [Касьянов и др., 2003], также упрощает понимание программы. Таким образом, программа на языке OPL, в общем случае, является более понятной, чем программы, представленные на языках других парадигм.

4. Модель данных, поддерживаемая парадигмой, включает данные (семантические сети) постоянного хранения (подобно базам данных). Это, во-первых, упрощает их повторное использование; во-вторых, упрощает разработку программ, поскольку не требуется дополнительного программирования для чтения информации и ее записи в поддерживаемый формат хранения.

5. Семантические сети образуют ансамбль, для которого программа функции является онтологией. Тем самым, многие функции оказываются повторно-используемыми. При этом если процесс формирования информационной структуры является интерактивным, то каждая функция может рассматриваться как редактор формирования такой структуры. Время разработки приложения за счет повторного использования функций может быть сокращено. Например, если функция представляет собой редактор историй болезни, то она может быть использована для ввода историй болезни в экспертных системах медицинской диагностики, в медицинских диагностических тренажерах, во многих других медицинских информационных системах. При этом для каждой такой системы не надо разрабатывать редактор для формирования историй болезни, что, в общем случае, является трудоемкой работой.

6. Абстрактный пользовательский интерфейс входит в операционную семантику логических формул языка. Таким образом, не требуется программирование интерфейса функций.

7. Процесс получения результата в предлагаемой парадигме до некоторой степени аналогичен такому же процессу в логической парадигме. Однако в логической парадигме на каждом шаге процесса вывода должно определяться множество применимых правил и множество значений их посылок. В онтологической парадигме процесс порождения является детерминированным, для каждого шага процесса построения семантической сети определена единственная формула, операционная семантика которой определяет действия на этом шаге.

В настоящее время сложно обозначить весь спектр применений парадигмы онтологического программирования. На сегодня видны два основных применения онтологической парадигмы программирования - создание интеллектуальных систем и редакторов информационных ресурсов любого типа – онтологий, знаний, данных.

Работа выполнена при финансовой поддержке РФФИ, проект 10-07-00090

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Обзор Microsoft, 2005] Обзор Microsoft // 2005. [Электронный ресурс] http://www.ict.edu.ru/ft/005126/intro_net.pdf.

[Промыслов и др., 2005] Промыслов В.Г., Жарко Е.Ф., Промыслова О.А. Практические аспекты сопровождения и модификации сложных программных систем // Труды IV Международной конференции "Идентификация систем и задачи управления" SICPRO'05. Москва, 25-28 января 2005 г. М.: ИПУ РАН, 2005. С. 1151.

[Дехтяренко, 2003] Дехтяренко И.А. Программирование в повествовательном наклонении // SoftCraft. Декларативное программирование. 04.02.2003. [Электронный ресурс] - <http://www.softcraft.ru/paradigm/dp/dp01.shtml>

[Lloyd, 1994] Lloyd J.W. Practical Advantages of Declarative Programming // Proceedings of the 1994 Joint Conference on Declarative Programming.

[Успенский и др., 1987] Успенский В.А., Семенов А.Л. Теория алгоритмов. Основные открытия и приложения - М.: Наука, 1987. - 288 с.

[Себеста, 2001] Себеста Роберт В. Основные концепции языков программирования = Concepts of Programming Languages. - 5-е изд. - М.: «Вильямс», 2001. — С. 672.

[Floyd, 1979] Floyd R.W. The Paradigms of Programming // Communications of the ACM, 1979. 22(8). Pp. 455-460.

[Касьянов и др., 2003] Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. – Изд-во: БХВ-Петербург, 2003. - 1104 с.

ONTOLOGICAL PROGRAMMING PARADIGM

Gribova V., Kleschev A.

*Institute of Automation and Control Processes,
Far Eastern Branch of RAS*

gribova@iacp.dvo.ru, kleschev@iacp.dvo.ru

This report is devoted to a new ontological programming paradigm. It is an evolution of the declarative programming paradigm. The data model, basic structures of language, and examples are described.

Key words: programming paradigm, data model, semantic network, ontology

INTRODUCTION

Software development is time consuming process. Even more difficult is software maintenance. They both requires new approaches from developers to resolve these problems. The declarative programming paradigm and languages realized this paradigm were proposed as an approach to solve the problem mentioned above.

In general a program in a declarative language is a description of an abstract model of the task to be solved, or, in accordance with [3], an executing specification of a result of computing. The programmer does not have to describe a process to control computation, it is the function of the language processor. Among other advantages is the fact that it is easier to write, to understand, and to maintain programs using languages of the paradigm in comparison with those written in imperative languages.

Declarative languages comprise logical and functional languages. However, the main idea of declarative programming in the modern logical and functional languages has not been realized yet

completely. As a result programs written by Prolog and Lisp are not considerably easier to develop, understand, and maintain than programs written in imperative languages. Among other drawbacks of declarative languages are poor facilities for user interface realization, difficulty of including imperative operators, if necessary.

The aim of this report is to suggest a new programming paradigm, called the ontological programming paradigm as a further evolution of the declarative programming paradigm and to suggest an ontological programming language OPL, satisfying the declarative programming definition where a program is an ontology of results of computing. Mechanisms for user interface realization and for including imperative structures are proposed.

MAIN PART

According to the suggested paradigm the process of obtaining results is a graph, a program in the OPL is an executive specification of the set of results of computation. This specification is an ontology of results.

All data in the ontological paradigm are semantic networks. The semantic network may be stored constantly (out of the programs) or temporarily (within the program). Using semantic networks as a data model means that objects of processing are complete information structures.

The OPL is a visual logical language of programming. Logical formulas of the language are: a simple formula, a simple quantifier formula, a unary formula, a propositional formula, a structural quantifier formula, a set of implications.

For realization of complicated calculations computed predicates are added in the OPL. Their purpose is to describe calculations using operators of an imperative language (for example, Java). A computed predicate has name and a set of formal parameters. Call of a computed predicate with a set of actual parameters can be only from an antecedent of an implication.

Abstract interface commands define functional of a user interface. They are divided into four classes: output commands, input (edit) commands of a value of a type, input (edit) commands of a set of values of a type, choice (edit) a subset from the set of values.

CONCLUSION

The new paradigm is intended for reducing labour-intensiveness of development and maintenance of intelligent systems. The main idea is to describe an ontology of results using the visual logical language of programming. The programmer does not have to describe a process to control computation, it is the function of the language processor. All data in the ontological paradigm are semantic networks. The language has facilities for user interface realization and for including imperative structures.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ, ОРИЕНТИРОВАННЫЕ НА ОБРАБОТКУ СЕМАНТИЧЕСКИХ СЕТЕЙ

Гулякина Н.А., Пивоварчик О.В., Лазуркин Д.А.

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

guliakina@bsuir.by

pivovarchyk@tut.by

lazurkin@bsuir.by

В работе рассматриваются языки и технологии программирования, ориентированные на обработку семантических сетей.

Ключевые слова: интеллектуальная help-система, семантическая теория программ, язык программирования, база знаний, графовые языки программирования.

ВВЕДЕНИЕ

Интеллектуализация является одним из основных направлений развития современных компьютерных систем. Для эффективного использования интеллектуальных технологий разработчики должны владеть методами представления знаний, методами моделирования баз знаний, технологиями проектирования программ, обрабатывающих знания. Высокая сложность и уникальность разработки интеллектуальных систем предъявляет высокие требования к квалификации их разработчиков. Поэтому важной задачей является повышение качества и сокращение сроков освоения этих технологий. Одним из способов решения этой задачи является создание программных средств поддержки разработки интеллектуальных систем, которые решают задачи информационного обслуживания по проектированию программного обеспечения и аккумулируют опыт квалифицированных программистов и преподавателей. Такие программные средства будем называть интеллектуальными help-системами. Интеллектуальные help-системы позволяют повысить уровень интеллектуальной информационной поддержки разработчиков программ. Наличие help-системы в составе инструментального средства проектирования программ позволит снизить начальные квалификационные требования к разработкам программ и повысить количество проектируемых интеллектуальных информационных систем.

В данной статье описана интеллектуальная help-система по разработке программ, ориентированных на обработку семантических сетей. В качестве языка программирования, ориентированного на обработку семантических сетей, в данной статье рассматривается язык SCP (Semantic Code Programming) [Голенков, В.В. и др., 2001b].

1. Языки программирования, ориентированные на обработку знаний

Языки программирования, ориентированные на обработку знаний, существенно отличаются от языков, ориентированных на обработку традиционных структур данных. Особенностью таких языков является:

- Использование специальных языков для представления обрабатываемых знаний;
- наличие гибких средств управления знаниями или базами знаний.

Среди языков, ориентированных на обработку знаний, выделяются языки программирования, ориентированные на обработку семантических сетей. Под семантической сетью понимается графовая структура, вершины и дуги которой обладают определенной семантической интерпретацией. [Голенков, В.В. и др., 2001a] Переработка знаний, представленных в виде семантических сетей, сводится к изменению конфигурации сети. Языки, тексты которых являются семантическими сетями, называются семантическими графовыми языками.

Примерами графовых языков представления и обработки знаний являются:

- сетевой язык представления фреймов [Поспелов Д.А. 1990];
- язык растущих пирамидальных сетей [Гладун В.П., 1987];
- графовый язык представления информации в памяти вегетативной машины [Борщев В.Б., 1990];
- графовый язык представления информации в памяти абстрактной сетевой машины [Степанов А.М., 1981];
- графовые логические языки [Вагин В.Н., 1989];
- волновые языки, обеспечивающие решение задач навигации на семантических сетях;
- графовые языки представления различным образом устроенных продукционных программ, ориентированных на переработку семантических сетей [Кузнецов В.Е., 1989];
- язык передачи маркеров и др.

К достоинствам семантических графовых языков представления знаний, по сравнению с другими способами представления знаний, относятся:

- компактность, обусловленная тем, что в отличие от символического текста в семантической сети знак каждого объекта или понятия описываемой предметной области присутствует только в одном экземпляре и состоит из одного элемента;
- ассоциативность, заключающаяся в существовании простых процедур поиска элементов семантической сети, связанных заданным образом с заданными элементами;
- наличие простой возможности введения метаинформации в семантическую сеть путем простого наращивания исходной семантической сети метасетью без какого-либо изменения исходной семантической сети;
- возможность рассмотрения описываемых предметных областей одновременно на неограниченном числе уровней детализации;
- приспособленность к поддержке структур любого вида, и в частности к поддержке сложноструктурированных знаний;
- приспособленность к интеграции самых различных специализированных языков и самых различных моделей представления знаний;
- приспособленность к представлению различного рода лингвистических знаний (о синтаксисе, о семантике, о прагматике естественных языков), что делает эффективным использование семантических графовых языков для создания естественно-языковых интерфейсных подсистем в интеллектуальных системах;
- приспособленность к параллельной асинхронной переработке знаний, что делает эффективным использование семантических графовых языков для создания интеллектуальных систем, поддерживающих

сложноструктурированные знания и сложные логические операции.

Для представления и обработки семантических сетей в проекте OSTIS (Open Semantic Technology for Intelligent Systems) [Электронный ресурс] используется семейство графовых языков, текстами которых являются sc-конструкции. Такие языки называются sc-языками. Базовым языком для построения sc-языков является SC-код. [Голенков В.В. и др., 2001a] С помощью SC-кода можно представлять любую информацию, в том числе и нечисловую, описывать сложноструктурируемые предметные области. На базе SC-кода разработано множество языков представления знаний, ориентированных на описание определенных предметных областей.

Хранимые в памяти интеллектуальной системы процедурные знания осуществляют описание методов решения различных классов задач и, следовательно, обеспечивают, с одной стороны, существенное повышение производительности интеллектуальной системы (если в состав процедурных знаний будут включены описания методов решения наиболее часто решаемых задач) и, с другой стороны, обеспечивают неограниченное расширение и модифицируемость системы операций переработки знаний.

К числу процедурных знаний относятся:

- всевозможные команды, каждая из которых описывает некоторое действие, считающееся элементарным в рамках соответствующей абстрактной машины и направленное либо на преобразование информации, хранимой в памяти этой машины, либо на преобразование ее внешней среды;
- всевозможные инициированные команды (операторы, инструкции), представляющие собой описание элементарных действий, подлежащих выполнению;
- всевозможные процедурные программы, каждая из которых представляет собой систему команд, иницируемых по тем или иным правилам;
- информационные конструкции, каждая из которых представляет собой описание текущего состояния процесса реализации некоторой процедурной программы.

Без активного использования в интеллектуальных системах различных программ (т.е. программных знаний о методах решения конкретных классов задач) практически невозможно создавать эффективные прикладные интеллектуальные системы. Таким образом, возникает проблема создания таких моделей переработки знаний и таких моделей реализации программ, которые бы легко интегрировались в одну модель, в рамках которой некоторые фрагменты знаний одновременно рассматривались бы как данные для соответствующих программ.

Интеграция формальных моделей реализации хранимых программ с формальными моделями

переработки знаний обеспечивает повышение производительности интеллектуальных систем (если в базу знаний включить программы, соответствующие часто решаемым задачам), а также обеспечивает открытый характер и легкую модифицируемость системы операций переработки знаний (для этого в базу знаний необходимо включить программы, которые описывают указанные операции).

Реализацию произвольной графодинамической формальной модели условно можно разбить на три этапа:

- приведение реализуемой графодинамической модели к каноническому виду, т.е. построение эквивалентной sc-модели;
- запись микропрограмм операций (правил вывода) для канонического вида реализуемой формальной модели (sc-модели) на некотором языке программирования;
- техническая разработка интерпретатора или компилятора указанного языка программирования, на котором записываются микропрограммы операций реализуемых формальных моделей.

Для обработки знаний, представленных с помощью sc-языков, существуют языки программирования:

- SCP (Semantic Code Programming) – базовый процедурный язык программирования;
- SCPH (Semantic Code Programming High-level) – процедурный язык программирования высокого уровня;
- функциональный язык программирования;
- логические языки программирования на базе языка SCL.

Данные языки программирования разработаны на базе SC-кода. Таким образом, они также являются sc-языками. Следовательно, программы и перерабатываемые ими данные, являются sc-конструкциями. Построение языков представления знаний и языков обработки знаний по одному принципу обеспечивает представление данных и программ в виде sc-конструкций, что позволяет легко интегрировать различные виды информации и хранить в одной базе знаний декларативные и процедурные знания.

2. Технология проектирования программ, ориентированных на обработку семантических сетей

Центральной проблемой любой эффективной технологии проектирования систем является то, как должна быть устроена система, чтобы ее легче было разрабатывать, модифицировать, пополнять и сопровождать. Такая технология должна быть ориентирована не столько на разработку определенного класса систем, сколько на их постоянное совершенствование.

Технология должна включать в себя:

- теорию проектируемых программ – уточнение того, как устроены проектируемые программы того класса, на которые ориентирована данная технология;
- структурированную библиотеку типовых многократно используемых компонентов проектируемых программ;
- инструментальные средства (средства автоматизации) проектирования;
- методику проектирования;
- методику обучения проектированию;
- интеллектуальную help-систему для информационного обслуживания и обучения проектировщиков;
- инфраструктуру, обеспечивающую организацию проектирования.

3. Язык программирования SCP

SCP (Semantic Code Programming) [Голенков, В.В. и др., 2001b] – это базовый язык программирования, ориентированный на обработку однородных семантических сетей, имеющих базовую теоретико-множественную интерпретацию. Для представления таких семантических сетей используется SC-код. Язык SCP является sc-языком, т.е. тексты языка (scp-программы) представляются в виде sc-конструкций. Программы языка SCP и перерабатываемые ими информационные структуры (sc-конструкции) могут храниться в одной базе знаний. Тест языка SCP с формальной точки зрения есть определенным образом устроенная sc-конструкция, хранящаяся в sc-памяти и описывающая некоторую систему параллельных взаимодействующих процессов переработки другой sc-конструкции, хранимой в той же sc-памяти. Особенности языка SCP являются:

- ориентация на использование структурно перестраиваемой (графодинамической) ассоциативной памяти;
- хорошая приспособленность к переработке нечисловых структур, что обеспечивает описание сложно структурированных предметных областей;
- ориентация на переработку непосредственно семантических сетей;
- возможность описания механизмов решения задач различного уровня сложности;
- возможность через понятие sc-узла интеграции scp-программ с программами, написанными на других языках программирования и описывающими переработку информационных структур;
- высокий потенциал распараллеливания процессов переработки информации в графовой структурно перестраиваемой ассоциативной памяти [Голенков, В.В. и др., 2001b].

Язык SCP обладает мощной типологией операторов, позволяющей производить различные виды операций с информационными структурами, представленными в виде семантических сетей.

Типология операторов языка SCP:

- операторы генерации sc-конструкций;
- операторы удаления sc-конструкций;
- операторы ассоциативного поиска sc-конструкций;
- операторы выборки;
- операторы проверки условий;
- операторы изменения свойств sc-элемента;
- операторы управления scr-процессом;
- операторы вывода.

Операционная семантика языка программирования SCP задается соответствующей абстрактной машиной. Абстрактная машина задается абстрактной памятью, в которой хранятся перерабатываемые конструкции, и множеством операций. Все вспомогательные данные, необходимые абстрактной scr-машине для реализации хранимых в памяти scr-программ, так же представляются в виде sc-конструкций. Абстрактная память является динамической информационной конструкцией. Структура памяти определяется синтаксисом языка, принципы изменения памяти определяются операциями.

Язык SCP разрабатывается и реализуется в нескольких модификациях, имеющих одинаковый набор непосредственно выполняемых операций, но различный синтаксис. Система операций языка SCP описана в [Голенков В.В. и др., 1994а], [Голенков В.В. и др., 1994б]. Реализован интерпретатор, обеспечивающий выполнение всех операций, включенных в текущую версию языка SCP. Для каждой из модификаций языка SCP дополнительно к указанному интерпретатору требуется реализовать конвертор из соответствующей формы представления scr-процедур в форму, поддерживаемую интерпретатором. Средства разработки программ, ориентированных на обработку семантических сетей

Инструментальное средство проектирования программ базового языка программирования, ориентированного на обработку унифицированных семантических сетей, поддерживают представление scr-программ в графической форме (SCg) и в линейной форме (SCs, M4SCP). Инструментальное средство, используемое в настоящее время, включает: редактор scr-программ, средство сборки репозитория исходных текстов, scr-интерпретатор, scr-отладчик с возможностью визуального отображения процесса отладки scr-программ [Лазуркин Д.А. и др., 2010].

Так как SCP является процедурным языком, то основная схема традиционного подхода отладки программ подходит и для создания отладчика scr-программ. Однако следует отметить, что тексты scr-программ представлены в виде семантических сетей и хранятся в ассоциативной памяти, что расширяет возможности отладки таких программ и упрощает сам отладчик по следующим причинам:

- структуру программ, представленных в виде sc-конструкций, легче модифицировать, чем их линейные аналоги;

- ассоциативный доступ к sc-памяти и явное выделение связей между частями scr-программы позволяет отладчику проявлять интеллектуальность без анализа низкоуровневой структуры отлаживаемой программы;

- часть символьной информации (имена программ, имена переменных) хранится в sc-памяти в виде идентификаторов соответствующих sc-элементов;

- упрощается возможность добавления к программе символьной метаинформации.

Процесс, интерпретирующий scr-программу, должен сообщать отладчику о ходе своей работы, поэтому для поддержки возможности отладки scr-программ необходимо расширить модель scr-процесса [Голенков, В.В. и др., 2001], [Голенков, В.В. и др., 2001b], [Электронный ресурс].

Трассировщик scr-процесса – это программа, которая сопоставлена данному scr-процессу и получает сведения о ходе его выполнения. На данном этапе разработки будем считать, что трассировщик является микропрограммой, т.е. его текст написан на языке внешнем по отношению к sc-памяти. Расширим связку, которая обозначает scr-процесс дополнительным компонентом с атрибутом *ptrace_*, который является знаком трассировщика. Таким образом, каждому scr-процессу может быть поставлен в соответствие только один трассировщик. Управление и всю необходимую информацию трассировщику передает scr-процесс при возникновении следующих ситуаций:

- осуществилось выполнение одного scr-оператора (только в случае пошаговой трассировки);
- в результате интерпретации scr-оператора произошла ошибка;
- scr-процесс создал процесс-потомок;
- scr-процесс завершился.

Отладка является частным случаем трассировки scr-процесса, поэтому создана специальная микропрограмма трассировщика. Она имеет доступ ко всем структурам scr-процесса и при необходимости может приостановить scr-процесс, исключив его из очереди планировщика. Эта же микропрограмма получает следующие сообщения управления от пользовательского интерфейса:

- продолжить выполнение текущего отлаживаемого scr-процесса, который приостановлен («продолжить исполнение»);
- включить режим пошаговой трассировки и перейти к следующему scr-оператору текущей scr-программы («шаг трассировки»);
- перейти к начальному scr-оператору scr-программы, которую вызывает текущий scr-оператор («шаг во внутрь»).

При создании дочернего scr-процесса для него устанавливается трассировщик родительского scr-процесса, если родительский scr-процесс находится в режиме трассировки.

Предложенный механизм низкоуровневой отладки *scr*-программ обеспечивает основу для организации визуальной отладки с отображением графодинамического процесса преобразования *sc*-памяти. Инструментальное средство является интеллектуальной системой и построено с использованием открытой семантической технологии.

4. Интеллектуальная help-система по проектированию программ, ориентированных на обработку семантических сетей

Интеллектуальная help-система является консультантом в области технологии проектирования программ, ориентированных на обработку семантических сетей. Для проектирования интеллектуальной help-системы используется открытая семантическая технология компонентного проектирования интеллектуальных систем. В основе семантической технологии проектирования интеллектуальных систем лежит семантическая модель представления знаний, которая использует семантическую сеть с базовой теоретико-множественной интерпретацией и представление знаний в *sc*-коде. Для представления знаний в базе знаний help-системы используется *sc*-код. [Голенков В.В. и др., 2001a] Таким образом, help-система является *sc*-системой.

Help-система состоит из следующих компонентов:

- интеллектуальная help-система по семантической теории программ;
- интеллектуальная help-система по библиотеке программ, ориентированных на обработку семантических сетей;
- интеллектуальная help-система по комплексу инструментальных средств проектирования программ, ориентированных на обработку семантических сетей;
- интеллектуальная help-система по методике проектирования программ, ориентированных на обработку семантических сетей;
- интеллектуальная help-система по методике обучения проектированию программ, ориентированных на обработку семантических сетей.

Каждый компонент содержит знания из соответствующей области технологии проектирования программ. В соответствии с открытой семантической технологией каждый компонент включает:

- справочную подсистему;
- подсистему мониторинга и анализа деятельности разработчика программ;
- подсистему управления обучением.

Каждая из подсистем взаимодействует с другими подсистемами, а также может функционировать автономно. Разработка каждой подсистемы состоит

из проектирования:

- базы знаний;
- машины обработки знаний;
- пользовательского интерфейса.

Одним из основных принципов открытой семантической технологии является унификация интеллектуальных систем. Унификация позволяет не только упростить разработку интеллектуальных систем, но и интегрировать как отдельные компоненты, так и целые системы. Это позволяет реализовывать каждый компонент help-системы и каждую подсистему отдельно и легко интегрировать их. В настоящее время разрабатывается справочная подсистема help-системы по семантической теории программ.

Справочная подсистема является консультантом-экспертом в области семантической теории программ, который может ответить на любой вопрос новичка или опытного пользователя. К функциям справочной подсистемы относятся:

- поиск информации по запросу пользователя, в том числе свободно-конструируемому;
- отображение найденной информации с учетом уровня квалификации пользователя;
- анализ программных текстов и внесение предложений по улучшению их эффективности;
- генерация программных текстов по запросу пользователя.

При проектировании интеллектуальной системы семантическая технология вначале предполагает разработку семантического языка представления вопросов. Разработанный язык определяет способы формулирования запросов к хранящейся в базе знаний информации и формирования ответов по каждому выделенному классу вопросов. Классы вопросов в разрабатываемой справочной системе определяются в соответствии с процедурами (операциями), которые требуются при ответе на них. В справочной подсистеме используются базовые классы семантического языка вопросов, а также выделены специальные классы. Основными классами вопросов являются:

- стандартные вопросы (например, запросы всех известных элементов заданного множества; запросы всех известных множеств, элементом которых является указываемый *sc*-элемент);
- запросы высказываний, связанных с заданными объектами, обобщенными структурами или высказываниями (например, относится ли *scr*-оператор *genEl* к классу *scr*-операторов поиска);
- запросы, связанные с классификацией множеств (например, какие классы *scr*-операторов являются подклассами операторов поиска);
- запросы, связанные с отношениями (например, привести отношения, заданные на множестве *scr*-программ);
- запросы, связанные с внешней идентификацией элементов базы знаний и с внешним представлением различных ее фрагментов;

- запросы, связанные с определениями и пояснениями понятий (например, привести определение понятия *scp-переменная*);
- запросы фрагментов семантических окрестностей элементов баз знаний (например, вывести полную семантическую окрестность понятия *scp-оператор*);
- запросы, связанные с доказательствами (например, истинным ли является данное высказывание «*scp-оператор genEl генерирует одноэлементную sc-конструкцию в базе знаний*»);
- запросы, связанные с программами (например, каким способом можно посчитать мощность множества);
- запросы фрагментов логических спецификаций формальных теорий (например, как выглядит теоретико-множественная система основных классов объектов семантической теории программ).

При анализе ответов на вопросы выделены ключевые понятия, которые легли в основу базы знаний, и определены отношения, для установления связей между ними. Также, для каждого понятия в базе знаний заданы его теоретико-множественные свойства. К теоретико-множественным свойствам можно отнести: пояснение и определение понятия, разбиение на более частные понятия, включение в более общие понятия, синонимия понятия, пересечение и объединение с другими понятиями, основные утверждения о понятии, примеры понятия и др.

В базе знаний справочной подсистемы интеллектуальной help-системы по семантической теории программ представлена информация о семантической теории программ, описаны синтаксис и денотационная семантика базового языка программирования, ориентированного на обработку унифицированных семантических сетей (языка программирования SCP), абстрактная машина, интерпретирующая базовый язык программирования.

Для описания семантической теории программ в виде семантической сети представлены:

- общие понятия теории программ;
- языки представления знаний;
- языки обработки знаний.

Выделены следующие ключевые понятия и отношения: *язык программирования, технология программирования, программа, алгоритм, оператор, операнд, операция, переменная, константа, множество, кортеж, решаемый класс задач*, описание операции** и др.

Для представления синтаксиса и денотационной семантики базового языка программирования в базе знаний используется семантическая сеть. В виде семантической сети представлены:

- структуры программы базового языка программирования, ориентированного на обработку унифицированных семантических сетей

- структуры обрабатываемых информационных конструкций;
- структуры оператора базового языка программирования, ориентированного на обработку унифицированных семантических сетей;
- типов операторов генерации унифицированных семантических сетей;
- типов операторов удаления унифицированных семантических сетей;
- типов операторов ассоциативного поиска унифицированных семантических сетей;
- типов операторов проверки условий;
- типов операторов изменения свойств элементов унифицированных семантических сетей;
- типов операторов управления процессами обработки унифицированных семантических сетей;
- типов операторов вывода на консоль;
- типов операторов ассоциативной выборки;
- примеров программ базового языка программирования, ориентированного на обработку унифицированных семантических сетей.

Выделены следующие ключевые понятия и отношения: *scp-программа, scp-константа, scp-переменная, scp-параметр, трехэлементная sc-конструкция, пятиэлементная sc-конструкция, одноэлементная sc-конструкция, scp-оператор, тип scp-оператора, scp-оператор генерации sc-конструкций, scp-оператор ассоциативного поиска sc-конструкций, scp-оператор удаления sc-конструкций, scp-оператор проверки условия, scp-оператор изменения свойств, scp-оператор управления scp-процессом, scp-оператор ассоциативной выборки sc-конструкций, значение*, атрибутивное отношение assign_, атрибутивное отношение fixed_ и др.* Также, в базу знаний включены утверждения и правила, описывающие способы проектирования программных текстов и задающие семантику языка.

Для каждого понятия выделены его свойства. Например, для понятия *scp-программа* выделены:

- синонимичные идентификаторы;
- пояснение;
- примеры (рисунок 1);
- структура программы.

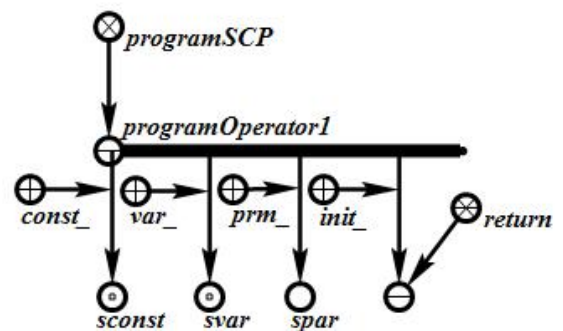


Рисунок 1 – SCP-программа с одним *scp-оператором*

Для понятия *тип scp-оператора* выделены

следующие свойства:

- синонимичные идентификаторы;
- определение;
- пояснение;
- разбиения по различным признакам (например, разбиение соответствующее типу выполняемого действия);
- структура (обобщенная для всех операторов);
- примеры.

Для понятия *genElStr3-оператор* выделены следующие свойства:

- синонимичные идентификаторы;
- пояснение;
- принадлежность;
- включение;
- компоненты;
- семантика;
- примеры с семантическими комментариями (рисунок 2).

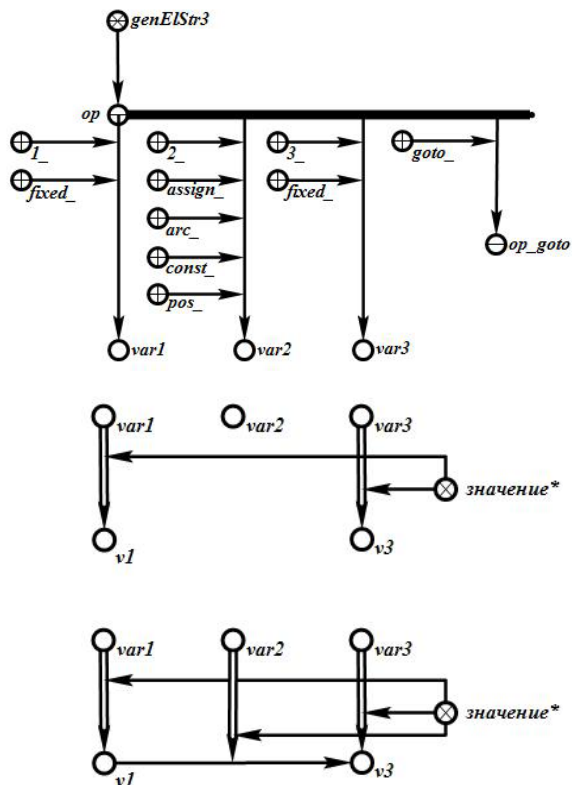


Рисунок 2 – Пример записи *scp*-оператора *genElStr3*, состояние памяти до и после выполнения оператора

При работе с системой пользователь может посмотреть синтаксис в виде семантической сети, в виде, представленном традиционно применяемыми средствами (форма Бэкуса-Наура и др.), в виде естественно-языковых текстов.

Для описания абстрактной машины, интерпретирующей базовый язык программирования, в базу знаний были включены понятия необходимые для представления в виде семантической сети:

- языковых средств, описывающих текущее состояние процесса интерпретации *scp*-программ;
- операций машины, отражающих специфику языка SCP (операционных семантик *scp*-операторов разных типов);
- используемых общих подпрограмм;
- модели *sc*-памяти;
- способов реализаций машины.

Выделены ключевые понятия: *scp-процесс*, *тип состояния scp-процесса*, *then-состояние scp-процесса*, *else-состояние scp-процесса*, *repeat-состояние scp-процесса*, *run-состояние scp-процесса*, *error-состояние scp-процесса*, *dead-состояние scp-процесса*, *sc-память*, *сегмент*, *операционная семантика genEl-оператора*, *операционная семантика genElStr3-оператора* и др. Для понятий выделены теоретико-множественные свойства. Например, для понятия *scp-процесс* определены синонимичные идентификаторы, пояснение, компоненты, примеры. Понятия для описания общих подпрограмм представляют собой имена программ, использующихся для реализации операционных семантик *scp*-операторов. Для каждой программы задана спецификация: пояснение, условие применения, класс задач, решаемых с помощью этой программы, алгоритм, примеры выполнения программы. Для описания понятий *операционная семантика* определенного *scp*-оператора выделены свойства: пояснение, алгоритм, примеры выполнения заданного оператора с семантическими комментариями.

Семантическая технология содержит множество типовых операций, которые можно использовать для реализации машины обработки знаний интеллектуальной *help*-системы по семантической теории программ. Множество операций машины обработки знаний *help*-системы было выделено на основании анализа спецификации вопросов. Каждому классу вопросов соответствует операция (типичная операция или специальная операция).

ЗАКЛЮЧЕНИЕ

Язык SCP в графодинамических ассоциативных машинах занимает особое место, поскольку его удобство и эффективность во многом определяют эффективность реализуемых с его помощью графодинамических моделей. Язык SCP является достаточно хорошо приспособленным к реализации преобразований, связанных с изоморфизмом *sc*-конструкций, благодаря ориентации языка SCP на переработку графовых конструкций, наличие в языке SCP операторов ассоциативного поиска, параллелизму.

Принципиальным отличием языка SCP от известных языков программирования является ориентация на графодинамическую (структурно перестраиваемую) ассоциативную память.

При разработке интеллектуальной *help*-системы для разработчиков программ, ориентированных на

обработку семантических систем, используется та же технология, что и использовалась для проектирования инструментальных средств. Это позволяет легко интегрировать систему с инструментальным средством и обеспечить информационное обслуживание разработчика scp-программ как при обучении, так и в процессе проектирования программного обеспечения. Язык программирования и язык представления знаний разработаны на базе той же технологии. Это позволяет унифицировать и интегрировать все необходимые знания о семантической теории программ.

Результаты, приведенные в работе, апробируются в рамках открытого проекта OSTIS (<http://ostis.net>).

Работа поддержана грантами БРФФИ-РРФФИ Ф10P-149, Ф10P-148, Ф10P-175

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Голенков В.В. 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.

[Электронный ресурс] Открытая семантическая технология компонентного проектирования интеллектуальных систем [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://ostis.net/mediawiki/index.php/>.

[Голенков В.В. и др., 2001b] Программирование в ассоциативных машинах / В.В.Голенков, Г.С.Осипов, Н.А.Гулякина и др. – Мн.: БГУИР, 2001. – 276 с.

[Поспелов Д.А., 1990] Искусственный интеллект: В 3 кн. Кн. 2. Модели и методы: Справочник /Под ред. Д.А. Поспелова. - М.: Радио и связь, 1990.

[Гладун В.П., 1987] Гладун В.П. Планирование решений. - Киев: Наукова думка, 1987.

[Борщев В.Б., 1990] Борщев В.Б. Параллельные асинхронные вычисления и моделирование параллельного выполнения логических программ (вегетативная модель вычислений и её применения) // Итоги науки и техники. Сер. Вычислительные науки. Том 4. - М.: ВИНТИ, 1990. - с. 114-197.

[Степанов А.М., 1981] Степанов А.М. Фреймы и параллельные смешанные вычисления. - Новосибирск, 1981. (Препринт / СО ВЦ АН СССР; N 297).

[Вагин В.Н., 1989] Вагин В.Н. Дедукция и обобщение в системах принятия решений. - М.: Наука, 1989.

[Кузнецов В.Е., 1989] Кузнецов В.Е. Представление в ЭВМ неформальных процедур. М.: Наука, 1989.

[Лазуркин Д.А. и др., 2010] Лазуркин Д.А., Пивоварчик О.В. Средства разработки программ, ориентированных на обработку семантических сетей // Информационные системы и технологии (IST'2010): материалы Y1 междунар. конф. (Минск, 24-25 нояб. 2010.- Минск: А.Н. Вараксин, 2010. - с. 501-504.

[Голенков В.В. и др., 1994a] Голенков В.В., Гулякина Н.А., Королев В.Г., Татаренко В.А., Золотой С.А.. Описание семантики языка SCPas. Операторы преобразования состояния SC-графа (Материалы по математическому обеспечению ЭВМ). – Мн: Ин-т техн. кибернетики АН Беларуси, 1994.

[Голенков В.В. и др., 1994b] Голенков В.В., Гулякина Н.А., Татаренко В.А., Гапонов П.А., Кузьмицкий В.М.. Описание семантики языка SCPas. Операторы поиска и проверки условий. Операторы управления вычислительным процессом (Материалы по математическому обеспечению ЭВМ). – Мн: Ин-т техн. кибернетики АН Беларуси, 1994.

LANGUAGES And TECHNOLOGIES PROGRAMMING, ORIENTED TO TREATMENT of SEMANTIC NETWORKS

Guliakina N.A., Pivovarchyk O.V.,
Lazurkin D.A.

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

**guliakina@bsuir.by
pivovarchyk@tut.by
lazurkin@bsuir.by**

The article is devoted to the intellectual help-system for programmes developers which are oriented on handling semantic net. Semantic programming languages and programming technologies are described. The components of help-system are presented in the article. The intellectual help-system of semantic programming theory is described in details.

INTRODUCTION

Programmers use programming technologies which are oriented on handling knowledge base for engineering intellectual information systems. The use of this technologies requires special knowledge and a high level qualification. The use of intellectual help-systems will permit to decrease the basic requirements to the programmer's qualification and to increase the quantity of the engineered intellectual information systems.

MAIN PART

The article is devoted to the programming technology which are oriented on handling semantic nets and to the intellectual help-system. Sc-code is a knowledge representation language. SCP is a programming language.

The intellectual help-system is composed of components. Components contain formal models of technology's areas: a semantic programming theory, programmes' libraries, compute-aided software engineering, a methodology of software engineering, a training methodology of software engineering.

Now we engineer intellectual help-system of semantic programming theory. For help-system's development we use the open semantic technology for intelligent systems (OSTIS).

CONCLUSION

The knowledge representation language, the programming language, the compute-aided software engineering, the intellectual help-system develop using one technology. It will permit to integrate the help-system and the compute-aided software engineering, to make sure information service of scp-programmes' developer in the time of learning and in the time of engineering



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

Онтологический подход к интеграции компонентов имитационной модели в Triad.Net

Волегов И.С., Замятина Е.Б.

** Пермский государственный национальный исследовательский университет
г. Пермь, Российская Федерация*

e_zamyatina@mail.ru

volegovis@gmail.com

В работе рассматриваются вопросы, связанные с автоматизацией и оптимизацией этапов имитационного эксперимента. Известно, что имитационное моделирование применяется как метод исследования в различных областях науки и производства. По этой причине сокращение времени на разработку имитационной модели и имитационного эксперимента является актуальным. В работе приводится опыт разработки системы Triad.Net и применение онтологического подхода для автоматического доопределения имитационной модели, добавления нового компонента, а также использования компонентов, подготовленных в других системах имитационного моделирования.

Ключевые слова: имитационное моделирование, онтологический подход, интеграция компонентов, переиспользование кода, оптимизация, автоматизация разработки модели.

ВВЕДЕНИЕ

Имитационное моделирование находит все большее применение как метод исследования сложных динамических систем. Этот метод актуален в том случае, если другие методы исследования, например, аналитические или численные, не могут быть применены, поскольку объект исследования сложно формализовать. Примерами сложных динамических систем могут служить вычислительные системы, системы телекоммуникации, бизнес-процессы, цепочки поставок и т.д.

Существует достаточно большое количество систем моделирования, которые позволяют выполнять исследования методом имитационного моделирования. Некоторые системы имитационного моделирования (СИМ) являются универсальными (GPSS WORLD[Sargent R.G.,2004], AnyLogic[Borshev A.,2007], Extend[Krahl D.,2002], , другие же предназначены для исследований в конкретной предметной области, т.е. являются специализированными (здравоохранение, логистика и т.д.).

Так или иначе, при выборе СИМ пользователи руководствуются определенными критериями, в числе которых можно назвать:

- простоту создания модели;

- валидность созданной модели и легкость проверки валидности;
- скорость выполнения имитационного прогноза;
- наглядность представления результатов моделирования;
- возможность получения нерегламентированных запросов при проведении имитационного прогноза и т.д.

Одним словом, все этапы имитационного эксперимента должны быть в той или иной мере автоматизированы, удобны пользователю и нацелены на оптимизацию времени проведения эксперимента, включая этап создания модели.

В настоящей работе рассматриваются вопросы применения онтологического подхода для автоматизации этапа создания модели, а более точно, интеграции компонентов в имитационную модель (ИМ), в том числе и тех компонентов, которые были созданы в других системах моделирования.

Онтологии находят широкое применение в имитационном моделировании.

Известно, что онтология - это описание типов сущностей, существующих в предметной области, их свойств и отношений между ними [Benjamin P. Et al, 1995]. Каждая предметная область – обычно, в этом контексте, это некая часть реального мира, например, производственная система, университет,

предприятие, - имеет собственную онтологию, которую называют онтологией предметной области. В онтологии предметной области определяются различные виды объектов (например, инструменты и работники), свойства (например, «быть сделанным из металла» или «иметь троих детей»), и отношения между видами и их экземплярами (например, «часть» или «женат на»).

Онтологии создаются во множестве областей знаний. Например, в последнее время было создано несколько онтологий для биологического домена. Создание онтологий для имитационного моделирования осложняется рядом причин [Miller J.A. et al, 2005], о которых речь пойдет далее.

Моделирование не ограничено конкретным доменом, поскольку модели могут представлять биологические, химические, физические, транспортные, военные и т.п. системы;

Моделирование и его методы основаны на математике, вероятностных и статистических расчетах, и, таким образом должно их придерживаться, поэтому онтологии для этих областей должны служить основой для всех остальных (т.н. онтологии среднего уровня)

В работе [Benjamin P. et al, 2006] рассматривается возможность использования онтологий на различных этапах имитационного моделирования начиная с этапа сбора информации о моделируемой системе и заканчивая этапом валидации модели. В работе определяются также цели и задачи использования онтологий на каждом этапе.

Примерами использования онтологий практически на всех этапах моделирования могут служить управляемые онтологиями среды моделирования (например, [Benjamin P. et al, 2005]), а так же подходы к объединению различных федератов, разрабатываемые для HLA [Rathnam T. et al, 2004].

Подход, разрабатываемый для HLA, использует онтологии для описания требований, которым должны удовлетворять интерфейсы федератов для успешного взаимодействия в федерации, а так же для разработки этих требований, с учётом знаний о моделируемой предметной области.

В работе [Liang V.-C. et al, 2003] представлена онтология портов, рассматриваемая как средство автоматизации компоновки моделей из компонентов.

Порты описывают интерфейс, определяющий границы компонентов или подсистем в конфигурации системы. Система представляется как конфигурация подсистем или компонентов, соединенных друг с другом через четко определенные интерфейсы. Конфигурация интерфейса компонента состоит из портов, которые определяют предполагаемое взаимодействие объекта с окружающей средой. Взаимодействия могут проходить в виде обмена энергией, материей или сигналами (т.е. информацией). При построении модели взаимодействие двух подсистем друг с другом обозначается наличием связи между соответствующими портами подсистем.

Модель представляется в виде графа, в вершинах которого находятся компоненты модели (модели подсистем). Вершины имеют порты – именно через них компонент взаимодействует со своим окружением. Наличие взаимодействия между компонентами представляется на графе в виде наличия связи между соответствующими портами.

То, что взаимодействия описываются при помощи портов, вовсе не означает, что использованы могут быть исключительно компоненты со строго стандартизированными интерфейсами. В том случае, когда описание некоторого типа соединения частей моделируемой системы отсутствует в онтологии портов (например, описание сварочного шва между деталями механизма), взаимодействие всё равно может быть представлено одним из относительно небольшого множества обобщённых взаимодействий (например, жёсткое механическое соединение).

Использование онтологии портов для описания возможных типов соединения частей системы позволяет не только автоматизировать процесс сборки моделей из компонентов путём использования знаний о том, какие виды портов могут быть соединены и каким образом, но и упрощают процесс построения компонентов, разрешая противоречия и многозначность терминологии.

1. Система имитационного моделирования Triad.Net

Система моделирования Triad.Net представляет собой совокупность лингвистических и программных средств имитационного моделирования.

1.1. Архитектура системы имитационного моделирования Triad.Net

СИМ Triad.Net включает следующие компоненты [Миков А.И. и др., 2010]: компилятор, ядро, графический редактор, подсистему отладки, подсистему валидации, подсистему синхронизации распределенных объектов модели, подсистему балансировки (распределенная версия), подсистему организации удаленного доступа, подсистему защиты от внешних и внутренних угроз, подсистему автоматического доопределения модели. Назначение каждого из компонентов представлено ниже: TriadCompile (компилятор языка Triad, переводит описание имитационной модели с языка Triad во внутреннее представление); TriadDebugger (отладчик, использует механизм информационных процедур алгоритма исследования, локализует ошибки и вырабатывает рекомендации для их устранения на основании правил из базы данных, для каждого класса ошибок осуществляется поиск по онтологии соответствующего обработчика ошибок); TriadCore (ядро системы, включает библиотеки классов основных элементов модели), TriadEditor (редактор моделей, предназначен для работы с моделью как в удаленном, так и локальном режимах, локальный режим предполагает работу с системой в том слу-

чае, если нет удаленного доступа), TriadBalance (подсистема балансировки), TriadSecurity (подсистема безопасности, этот компонент используют при удаленном доступе к системе моделирования), TriadBuilder (подсистема автоматического доопределения частично описанной модели), база данных, где хранятся экземпляры элементов модели, TriadMining-набор процедур для исследования результатов модели методами DataMining, TriadRule – алгоритм синхронизации объектов распределенной модели, использующей для вычислительного эксперимента ресурсы нескольких вычислительных узлов.

1.2. Представление имитационной модели в Triad.Net

Описание модели в системе Triad [Mikov, 1995] можно определить как $M = (STR, ROUT, MES)$, где STR – слой структур, ROUT – слой рутин, MES – слой сообщений.

Слой структур представляет собой совокупность объектов, взаимодействующих друг с другом посредством посылки сообщений. Каждый объект имеет полюсы (входные и выходные), которые служат соответственно для приёма и передачи сообщений. Основа представления слоя структур – графы. В качестве вершин графа следует рассматривать отдельные объекты. Дуги графа определяют связи между объектами.

Объекты действуют по определённому алгоритму поведения, который описывают с помощью рутины. Рутинa представляет собой последовательность событий e_i , планирующих друг друга (E – множество событий; множество событий рутины является частично упорядоченным в модельном времени). Выполнение события сопровождается изменением состояния объекта. Состояние объекта определяется значениями переменных рутины. Таким образом, система имитации является событийно-ориентированной. Рутинa так же, как и объект, имеет входные и выходные полюса. Входные полюса служат соответственно для приёма сообщений, выходные полюса – для их передачи. В множестве событий рутины выделено входное событие e_{in} . Все входные полюса рутины обрабатываются входным событием. Обработка выходных полюсов осуществляется остальными событиями рутины. Для передачи сообщения служит специальный оператор *out* (*out* <сообщение> *through* <имя полюса>). Совокупность рутин определяет слой рутин ROUT.

Слой сообщений (MES) предназначен для описания сообщений сложной структуры.

Система Triad реализована таким образом, что пользователю необязательно описывать все слои. Так, если возникает необходимость в исследовании структурных особенностей модели, то можно описать только слой структур. В слое структур определены стандартные процедуры, с помощью которых можно определить множество вершин графа, множество ребер, дуг и т.д., найти кратчайшее расстояние между двумя вершинами, компоненты связности GetStronglyConnectedComponents(G), выделение

слоя из структур модели GetGraphWithoutRoutines(M).

Ниже приведено описание слоя структур модели, которая представляет собой фрагмент компьютерной сети, состоящей из рабочих станций, передающих сообщения друг другу, и маршрутизаторов, отвечающих за нахождение пути передачи данных.

```
Type Router,Host; integer i;
M:=dStar(Rout[5]<Pol[4]>);
M:=M+node Hst[8]<Pol>;
M.Rout[0]=>Router;
for i:=1 by 1 to 4 do
  M.Rout[i]=>Router;
  M:=M+edge(Rout[i].Pol[1] — Hst[2*i-2]);
  M:=M+edge(Rout[i].Pol[2] — Hst[2*i-1]);
endf;
for i:=0 by 1 to 7 do
  M.Hst[i]=>Host;
endf;
```

Следует обратить внимание, что в Triad модель рассматривается как переменная. Она может быть построена с помощью операций над моделью. При построении модели используют графовые константы, которые соответствуют основным топологиям компьютерных сетей. В приведенном выше описании модели использовали графовую константу «звезда» (dstar). Кроме того, в приведенном выше примере были использованы «семантические» типы (Type Router,Host). В данном случае, это семантические типы «маршрутизатор», и «хост». Семантические типы используют для того, чтобы можно было доопределить модель с помощью рутины, извлеченной из библиотеки экземпляров рутин. Для поиска соответствующего экземпляра рутины.

Система Triad реализована таким образом, что пользователю необязательно описывать все слои. Так, если возникает необходимость в исследовании структурных особенностей модели, то можно описать в модели только слой структур.

Алгоритмом имитации будем называть совокупность объектов, функционирующих по определённому сценарию, и синхронизирующий их алгоритм.

1.3. Алгоритм исследования

Для сбора, обработки и анализа имитационных моделей в системе Triad.Net существуют специальные объекты – информационные процедуры и условия моделирования. Информационные процедуры и условия моделирования реализуют алгоритм исследования модели.

Информационные процедуры ведут наблюдение за элементами модели (событиями, переменными, входными и выходными полюсами), указанными пользователем. Если в какой-либо момент времени имитационного эксперимента пользователь решит, что следует установить наблюдение за другими элементами или выполнять иную обработку собираемой информации, он может сделать соответствующие указания, подключив к модели другой набор информационных процедур. Условия моделирования анализируют результат работы информационных процедур и определяют, выполнены ли

условия завершения моделирования. Кроме того, именно условия моделирования позволяют выполнить операции над моделью.

В системе Triad.Net для анализа функционирования компьютерной сети можно использовать стандартные и пользовательские информационные процедуры. Пользовательские информационные процедуры описываются на языке Triad [Миков, 1995]. Для каждого элемента сети можно указать список необходимых информационных процедур, которые будут вести наблюдение во время моделирования за переменными, событиями и полюсами элемента. Система также предоставляет лингвистические средства для создания собственных условий моделирования, в которых можно описывать оригинальные алгоритмы сбора статистики и алгоритмы преобразования модели в динамике.

2. Онтологический подход к интеграции компонентов имитационной модели

При описании проектов по имитационному моделированию, в которых использовались онтологии, приведен пример проекта PortOntology, в котором знания о портах компонентов были использованы для автоматической сборки модели из компонентов.

В СИМ Triad.Net онтологии используются для доопределения частично описанной модели [Mikov A. et al, 2007], а именно, для доопределения поведения некоторого устройства, например, маршрутизатора в компьютерной сети. Это бывает необходимо на начальных этапах моделирования, когда исследователю или проектировщику необходимо получить оценки моделируемого объекта или системы, даже если они являются достаточно грубыми. При этом поиск компонента модели (экземпляра рутин) в базе данных осуществляется с помощью онтологий автоматически.

Для оперативного построения имитационной модели в конкретной предметной области используются графический интерфейс [Замятина Е.Б. и др., 2011], добавление нового компонента модели (экземпляра рутин) и настройка на конкретную предметную область выполняется с помощью онтологий.

Кроме того, в СИМ Triad.Net разрабатываются программные механизмы, позволяющие интегрировать в Triad-модель компоненты, разработанные в других СИМ. Это позволит не только воспользоваться уже готовыми моделями и провести эксперименты с ними в среде Triad, но и организовать распределенный имитационный эксперимент.

2.1. Автоматическое доопределение модели

В процессе проектирования модели, модель может быть определена частично. В настоящее время реализована работа с частично определёнными моделями вида:

$$M = \{STR; ROUT^*; MES\}$$

При этом:

$$ROUT^* \subseteq ROUT$$

т.е. не установлено поведение ряда объектов модели. Иными словами, в модели присутствуют вершины, не расшифрованные структурой (терминальные), не имеющие наложенной на них рутин.

Таким образом, задача системы автоматического доопределения модели состоит в том, чтобы для каждой такой вершины подобрать подходящую рутину. Для этого, очевидно, необходима информация о существующих рутинках, а так же некоторые знания о том, что собственно моделируется. Для этих целей используется понятие семантического типа.

Семантический тип – специальное понятие. Оно вводится для того, чтобы сгруппировать ряд объектов по некоторому смысловому, структурному, поведенческому типам [20]. Так, для обозначения множества процессорных устройств при моделировании вычислительных систем, может быть введён семантический тип Процессор, для обозначения элементов памяти – тип МодульПамяти. При моделировании систем массового обслуживания уместны будут семантические типы **Очередь**, **ГенераторЗаявок** и т.п.

Для того, чтобы причислить объект к тому или иному семантическому типу, в тексте программы употребляется специальный оператор:

<имя объекта> => <имя типа>

Сами семантические типы объявляются специальным оператором **type** <имя типа>

Фрагмент программы, использующей семантические типы, приведен выше.

В системе Triad.Net выделены условия доопределения терминальной вершины экземпляром рутин:

- Условия специализации;
- Условия конфигурации;
- Условия декомпозиции.

Условие специализации предполагает, что накладываемый на вершину экземпляр рутин, найденный в базе данных, имеет тот же семантический тип, что и терминальная вершина.

Условие конфигурации налагает ограничение на количество полюсов экземпляра рутин. Количество полюсов должно соответствовать количеству полюсов вершины.

Условие декомпозиции предназначено для проверки «графа окружения». Это означает, что необходимо проверить соответствие топологии графа, который образован вершинами, смежными с терминальной, и «графом окружения» из базы данных.

Если условия выполняются, то соответствующий экземпляр рутин накладывается на терминальную

вершину. В результате исследователь получает полностью определенную модель, готовую к выполнению.

3. Добавление нового компонента модели

3.1. Представление знаний

Для настройки на конкретную предметную область в Triad используют онтологии. В Triad разработана базовая онтология. Основу ее составляют следующие классы:

- TriadEntity – любая логическая сущность языка Triad, имеющая имя. Подклассами TriadEntity являются все классы базовой онтологии, кроме вспомогательных (например, тип полюса), а именно:
- Model – имитационная модель.
- ModelElement – составная часть имитационной модели, а также все, чем может быть специализирована вершина структуры имитационной модели.

Подклассами ModelElement являются классы:

- Structure – структура имитационной модели.
- Routine – рутина.
- Message – сообщение и т.д.

Основными свойствами в базовой онтологии являются следующие свойства:

- Свойства владения чем-либо: модель имеет структуру, структура имеет вершину, вершина имеет полюс и т.д.
- Свойства принадлежности к чему-либо – inverse properties по отношению к соответствующим свойствам владения – структура принадлежит модели, вершина принадлежит структуре, полюс принадлежит вершине и т.д.
- Свойства, связывающие полюс с присоединенной к нему дугой: connectsWithArc (Pole, Arc), connectsWithPole (Arc, Pole).
- putsOn (Routine, Node) – свойство, связывающее вершину с наложенной на нее рутинной.
- Свойства, связывающие вершину с расшифровывающей ее структурой: explicatesNode (Structure, Node), explicatedByStructure (Node, Structure).
- modelingToCondition (Model, ModelingCondition) – свойство, связывающее модель с условием моделирования.

На основании СИМ Triad.Net было разработано программное средство для проектирования компьютерных сетей TRIADNS[Замятина Е.Б. и др.].

Онтология системы TriadNS дополняет базовую онтологию. Введены специализированные для области компьютерных сетей подклассы основных классов базовой онтологии (см. рис.1):

- ComputerNetworkModel – модель компьютерной сети;
- ComputerNetworkStructure – структура модели компьютерной сети;
- ComputerNetworkNode – элемент компьютерной сети, изначально содержит подклассы WorkStation, Server, Router;
- ComputerNetworkRoutine – рутинная элемента компьютерной сети, и т.д.

В онтологии есть два специальных свойства для полюса:

- isRequired(ComputerNetworkRoutinePole, Boolean) - обязательно ли полюс должен быть соединен с другим полюсом
- canConnectedWith(ComputerNetworkRoutinePole, ComputerNetworkRoutine) – определяет семантический тип элемента, с которым полюс можно соединить.

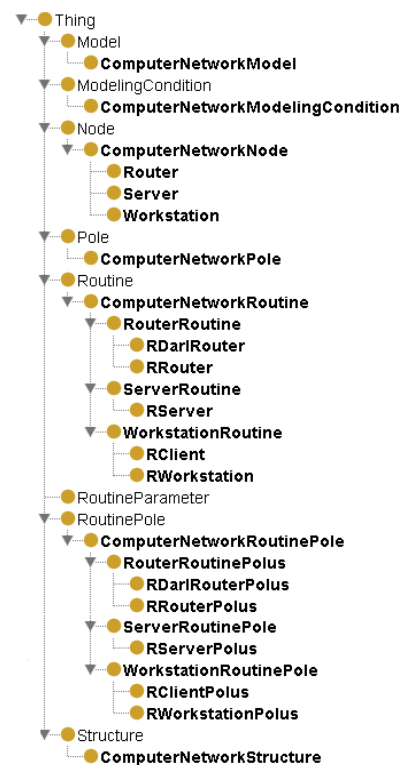


Рисунок 1 Иерархия классов онтологии

Имитационная модель компьютерной сети может быть представлена графически или описана на языке Triad (входной язык CAD TRIADNS). Рассмотрим более подробно особенности работы с графическим интерфейсом.

3.2. Графический интерфейс

Графический редактор позволяет оперативно

проектировать компьютерные сети. Структуру сети формируют путем перемещения пиктограмм из панели элементов, обозначающих конкретные элементы сети, в рабочую область, затем добавляют связи между ними (см.рис.2).

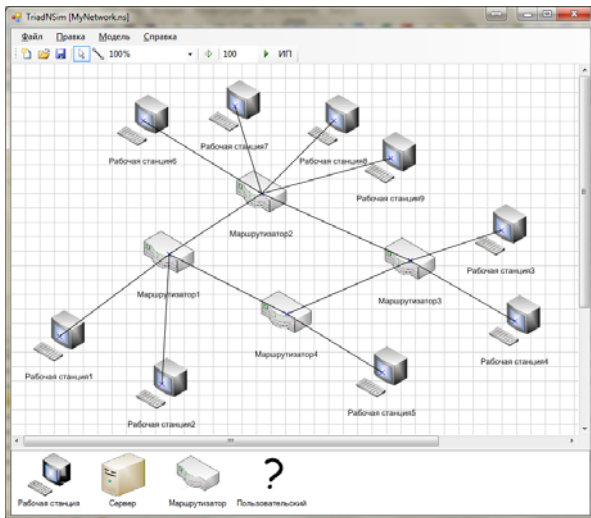


Рисунок 2 – Окно графического редактора

Как уже говорилось ранее, имитационная модель имеет графовое представление. Элементы сети являются вершинами этого графа. Элементы сети загружаются из онтологии предметной области (подклассы класса `ComputerNetworkNode`, изначально он содержит 3 подкласса: «Рабочая станция», «Сервер», «Маршрутизатор»). Пользователь может также добавлять собственные элементы с помощью диалоговых окон системы, для этого нужно указать имя элемента, описание, суперкласс, изображение. При этом элементы программно добавляются в онтологию с использованием библиотеки OWL API.

Для функционирования сети необходимо определить поведение каждой из вершин графа. Поведение вершин графа представлено рутинами. Все возможные рутинные элементы также описывают в онтологии. Стандартные рутинные элементы имеют некоторое количество параметров, которые пользователь может изменять, после наложения рутин на элемент сети. Для каждого элемента сети можно определить произвольное количество рутин, описав их на языке Triad и добавив в онтологию. При добавлении рутин следует указать необходимость соединения каждого полюса рутин и семантический тип возможной смежной вершины, значения по умолчанию для указанных в рутине параметров.

4. Средства поддержки интероперабельности имитационных моделей в Triad.Net

За последние годы было создано множество систем имитационного моделирования, среди которых четко прослеживается деление на классы: дискретно-событийные системы, процессно-ориентированные, специализированные системы по моделированию физических процессов и другие.

Системы, находящиеся в рамках одного класса очень похожи по своей структуре, при этом каждая обладает рядом преимуществ или недостатков перед конкурентами, но, зачастую, не имеет никаких средств взаимодействия с другими имитационными моделями.

При работе с такими системами пользователь жестко ограничен ее функциональными возможностями, не имея средств переноса своей модели в другую систему имитационного моделирования, расширить язык введением или модификацией существующей конструкции или настройки визуальной среды моделирования на конкретную предметную область. Эти недостатки можно устранить, введя дополнительный уровень в систему имитации: уровень метамodelей, который сможет описать:

- Целевой язык имитационного моделирования, при помощи которого будет производиться имитационный эксперимент, в данном случае это язык Triad – входной язык СИМ Triad.Net.
- Язык-источник, представляющий уже существующий язык имитационного моделирования, модели которого нам хотелось бы перенести на платформу целевого языка для дальнейшей модификации и исследования.
- Конкретные предметные области, т.к. зачастую пользователю бывает сложно работать с абстрактными концепциями систем имитационного моделирования.
- Взаимосвязи между вышеперечисленными пунктами

Реализация системы Triad.Net предполагает создание метамodelей, которые будут описывать все важнейшие этапы функционирования: настройку транслятора и конвертера на определенную систему имитационного моделирования и настройку среды разработки на конкретную обобщенную модель, при этом при преобразовании моделей будут использоваться отображения одной метамodelи на другую.

Структура информационной системы при таком подходе будет иметь вид, представленный на рис.3. Такой подход позволит создать открытую систему имитационного моделирования, предоставляющую пользователю гибкий набор механизмов взаимодействия с ней и расширения ее функционала:

- Описание дополнительных метамodelей для систем имитационного моделирования, позволяющее расширить круг доступных для работы моделей.
- Расширяемости метамodelей.

Расширяемость метамodelей предполагает:

- Изначально пользователь может создавать не полноценные метамodelи систем имитационного моделирования, а неко-

торые их приближения, отражающие лишь необходимые на данном этапе элементы и связи.

- При переходе сторонней системы на более новую версию или добавление в нее функционала метамодель может быть доработана (а не создана с нуля).
- При недостаточных описательных возможностях обобщенной метамодели, используемой в визуальном редакторе, ее так же можно расширить.

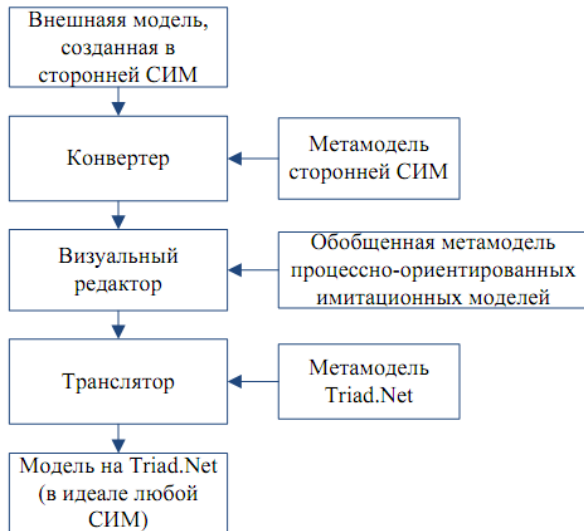


Рисунок 3 Архитектура системы

Важным этапом в создании любой информационной системы является этап определения формализма представления модели. С точки зрения открытых систем решение должно быть масштабируемым и обладать свойством интероперабельности, т.е. в данном случае, с точки зрения моделей, формализм должен предоставлять возможности по переводу модели из одного представления в другое, будь то графовые трансформации или наборы генераторов элементов модели.

Рассмотрим основные особенности языков имитационного моделирования, на которые стоит обратить внимание при выборе формализма:

- В основе модели лежит граф, описывающий основные элементы модели и связи между ними.
- Большинство языков предоставляет возможности по иерархическому описанию моделей и ее элементов.

С точки зрения метамоделирования формализм должен предоставлять возможности по отображению семантических связей, таких как наследование, агрегация и декомпозиция. Отдельно можно заметить, что предпочтение стоит отдать формализму, обладающему общедоступной спецификацией, чтобы по возможности обеспечить и перенос самих метамodelей.

Всем этим требованиям соответствуют онтоло-

гии. Использование онтологий в качестве метамodelей имитационного моделирования позволяет абстрагироваться от конкретной реализации и создать систему, которая будет способна работать с большинством современных СИМ. Пользователи легко смогут переходить к другим системам имитационного моделирования, объединять модели, написанные в разное время на разных языках имитационного моделирования или разрабатывать модели для исследования сразу в нескольких СИМ. Это позволит использовать совокупный потенциал различных СИМ без необходимости многократного описания одной и той же модели.

Главным недостатком сред распределенного имитационного моделирования, помимо их громоздкости и большого ряда ограничений и правил, накладываемых на структуру имитационной модели, является отсутствие семантики в передаваемых сообщениях. При использовании таких систем появляется необходимость в дополнительной обработке сообщений, определении его структуры и содержания, в то время, как в терминах конкретной парадигмы имитационного моделирования каждое сообщение можно соотнести с тем или иным концептом, понятным как источнику сообщения, так и приемнику.

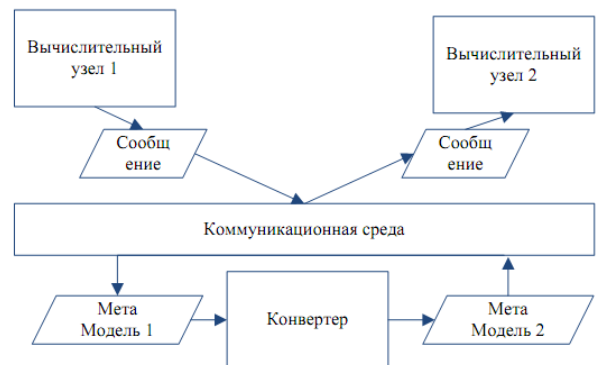


Рисунок 4 Архитектура распределенного имитационного эксперимента

Как мы видим на рисунке 4, этот недостаток можно устранить, добавив уровень метамodelей в коммуникационную прослойку, позволив узлам отсылать непосредственно элементы моделей. В коммуникационной прослойке выполняется конвертация сообщений, основанная на отображении метамodelей СИМ-источника и СИМ-приемника.

В настоящий момент ведутся работы по реализации этого проекта, разработана базовая онтология СИМ Triad.Net, онтология процессо-ориентированных СИМ, выполнены работы по конвертации онтологии процессо-ориентированной модели в Triad-модель.

ЗАКЛЮЧЕНИЕ

Итак, в работе сделан краткий обзор и представлены примеры применения онтологического подхода для интеграции компонентов имитационной мо-

дели: автоматическое добавление недостающих компонентов в частично определенной модели, добавление в Triad модель компонентов, разработанных в других СИМ. Онтологический подход позволяет сделать систему открытой и адаптируемой, позволяет автоматизировать этап создания модели, а также оптимизировать этот процесс.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Sargeant R.G., 2004] Sargent R.G. Some Recent Advances In The Process World View. // Proceedings of the 2005 Winter Simulation Conference / M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.,— pp 293-298

[Borshev A., 2007] Borshev A. AnyLogic Professional. Winter Simulation Conference (WSC'07), Washington, (December 2007), Final Program. Abstracts. p. 1.

[Krahl D.] Krahl D. The Extend Simulation Environment. Proceedings of the 2001 Winter Simulation Conference (WSC'01), Crystal Gateway Marriott, Arlington, VA, (December 2001). pp. 217-225.

[Benjamin P.,1995], Benjamin P., Menzel C, Mayer R. J. Towards a method for acquiring CIM ontologies. // International Journal of Computer Integrated Manufacturing, 8 (3) 1995, pp. 225-234.

[Miller J.A., 2005] Miller J.A., Baramidze G. Simulation and the Semantic Web // Proceedings of the 2005 Winter Simulation Conference / M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds..— pp. 2371-2377

[Benjamin P., 2006] Benjamin P., Patki M., Mayer R. J. Using Ontologies For Simulation Modeling // Proceedings of the 2006 Winter Simulation Conference/ L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds. —pp.1161-1167

[Benjamin P., 2005] Benjamin P., Akella K.V., Malek K., Fernandes R. An Ontology-Driven Framework for Process-Oriented Applications // Proceedings of the 2005 Winter Simulation Conference / M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.,— pp 2355-2363

[Rathnam T.,2004], Rathnam T., Paredis C.J.J. Developung Federation Object Models Using Ontologies // Proceedings of the 2004 Winter Simulation Conference / R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.,— pp 1054-1062

[Liang V.-C.,2003], Liang V.-C, Paredis C.J.J. A Port Ontology for Automated Model Composition // Proceedings of the 2003 Winter Simulation Conference / S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds., - pp 613-622

[Mikov, 1995] Mikov A.I. Formal Method for Design of Dynamic Objects and Its Implementation in CAD Systems // Gero J.S. and F.Sudweeks F.(eds), Advances in Formal Design Methods for CAD, Preprints of the IFIP WG 5.2 Workshop on Formal Design Methods for Computer-Aided Design, Mexico, Mexico, 1995. pp.105-127.

[Mikov A.,2007], Mikov A., Zamyatina E., Kubrak E. Implementation of simulation process under incomplete knowledge using domain ontology //Proceedings of the 6-th EUROSIM Congress on Modeling and Simulation. University of Ljubljana, Slovenia, 2007, Vol.2. Book of full papers. p.1-7.

[Замятина Е.Б., 2011], Замятина Е.Б., Миков А.И., Михеев Р.А. Применение онтологий в системе проектировании и моделирования компьютерных сетей TRIADNS. Труды Конгресса по интеллектуальным системам и информационным технологиям. «IS&IT'11», Научное издание в 4-х.томах, М. Физматлит, 2011, Т.1, стр. 253-260

[Миков А.И., 2010] Миков А.И., Замятина Е.Б. Проблемы повышения эффективности и гибкости систем имитационного моделирования. Проблемы информатики, №4(8), Новосибирск, Институт вычислительной математики и математической геофизики СО РАН, 2010, стр.49-64

THE INTEGRATION OF SIMULATION MODEL COMPONENTS INTO TRIAD.NET USING ONTOLOGICAL APPROACH

Volegov I.S. , Zamyatina E.B.

*Perm State Natonal Research University, Perm,
Russion Federations*

Volegovis@gmail.com, e_zamyatina@mail.ru

INTRODUCTION

The paper discusses the problems of the use of ontology in various stages of simulation. More precisely, it focuses on the problem of automatic simulation model пугтукфешшт using ontological approach. The following sections consider simulation system Triad.Net and some issues of simulation components integration into simulation model Triad: the completion of partly defined simulation model (the behavior of some objects is not defined), the definition of the behavior of new object in graphical interface, the integration of the simulation components made in another simulation systems.

MAIN PART

Triad.Net – is a simulation system for the investigation of the complex computing systems. A simulation model $\mu=\{STR, ROUT, MES\}$ consists of three layers, where STR is a layer of structures, ROUT – a layer of routines and MES – a layer of messages.

The layer of structures is dedicated to describe the physical units and their interconnections, but the layer of routines presents its behavior. Each physical unit can send a signal (or message) to another unit. Many objects being simulated have a hierarchical structure. Thus their description has a hierarchical structure too. One level of the system structure is presented by graph $P = \{U, V, W\}$.

As it was described above in a completely described model each terminal node $v_i \in V$ has an elementary routine $r_i \in ROUT$. An elementary routine is represented by a procedure. This procedure has to be called if one of poles of node v_i receives a message.

But some of the terminal nodes v_i of partly described model have no routines. The routine has to be found in special library using ontology being defined for Triad model.

Moreover the behavior of new object in graphical interface can be defined using this ontology. And the next example: the simulation component being designed in another simulation system may be integrated in Triad simulation model using this ontology too.

CONCLUSION

Thus the suggested approach allows to automate the process of simulation model generation. This approach supposes use of ontologies. Ontologies are the convenient path to domain describing. So a simulation system using ontologies becomes a powerful tool for scientific research because it complies with principles of open source software.

СЕКЦИЯ № 3

СЕМАНТИЧЕСКИЕ МОДЕЛИ ИНФОРМАЦИОННОГО ПОИСКА И РЕШЕНИЯ ЗАДАЧ. СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННО-ПОИСКОВЫХ МАШИН И РЕШАТЕЛЕЙ ЗАДАЧ

SECTION № 3

SEMANTIC MODELS OF INFORMATION SEARCH AND PROBLEM SOLVING. SEMANTIC TECHNOLOGIES OF DESIGN OF INTELLIGENT INFORMATION RETRIEVAL ENGINES AND INTELLIGENT PROBLEM SOLVERS

Данная секция посвящена семантическим моделям решения задач, простейшим видом которых являются семантические модели решения информационно-поисковых задач. Особенность информационно-поисковых задач заключается в том, что результат их решения уже содержится в текущем состоянии базы знаний, его надо только локализовать (найти, выделить). В отличие от этого в ходе решения задач других видов осуществляется генерация (порождение, построение) знаний, являющихся результатами решения этих задач.

Постоянное увеличение объемов информации, доступной в Интернете, затрудняет нахождение нужной информации конечными пользователями. Искомая информация может храниться на сайтах не только в текстовом виде, а также в виде картинок, мультимедиа и т.д.

Ведущие поисковые системы, такие как Google, Yandex, Bing пытаются помочь пользователю в процессе нахождения информации, путем локализации сервисов и выстроить так называемые вертикали, которые помогут сузить область поиска и тем самым повысить вероятность получения нужного ответа. Однако, в качестве ответа все равно будет предложен набор документов и ссылок на сайты, которые, по мнению поисковой системы, являются наиболее релевантными запросу.

Отсутствие, как правило, семантической разметки на сайтах с использованием существующих языков для описания онтологий, например, RDF и OWL, затрудняет осуществление интеллектуального поиска с использованием программных агентов.

Возникает вопрос, поможет ли внедрение в описание всех сайтов семантики полностью решить информационную задачу поиска? Сможет ли тогда конечный пользователь получать ответ на свой вопрос в удобном для него виде без необходимости изучения нескольких десятков страниц поисковой выдачи?

На эти вопросы на данный момент нет однозначного ответа. Если в результате поиска необходимо ответить на простые вопросы, которые не требуют сложных логических рассуждений, то существующие поисковые машины могут эффективно с ними справиться, тем более, что внедрение на всех Интернет-ресурсах онтологического описания является практически неосуществимой задачей.

Семантический поиск, в свою очередь, поможет ответить на сложные логические вопросы пользователя. Также стоит отметить, что реализация семантического поиска в сети Интернет для обработки запросов на естественном языке, будет, скорее всего, реализовываться с использованием существующих поисковых систем, в которых уже решены такие проблемы как, например, морфологический анализ.

Использование онтологий как основы для решения информационных задач поиска существенно при необходимости нахождения информации в рамках одной хорошо структурированной предметной области. Также известны примеры применения онтологий для объединения баз знаний, баз данных, при построении информационных порталов, где на их основе строится единая точка взаимодействия с гетерогенными источниками. Данные направления будут преобладать в будущем для практического применения онтологий в решении информационных задач поиска.

Важнейшими проблемами, связанными с проектированием интеллектуальных моделей решения задач являются:

- обеспечение семантической совместимости всех этих моделей;
- интеграция различных моделей и стратегий решения задач;
- создание библиотеки многократно используемых компонентов, из которых можно синтезировать различные интеллектуальные решатели задач, использующие самые различные модели и стратегии решения задач в самых различных сочетаниях.

ВОПРОСЫ, ВЫНОСИМЫЕ НА ОБСУЖДЕНИЕ КРУГЛОГО СТОЛА СЕКЦИИ № 3 :

- *Уточнение понятия вопроса и понятия информационной задачи в рамках семантической модели обработки информации.*
- *Типология и многообразие вопросов.*
- *Каковы направления интеллектуализации информационного поиска.*
- *В чем специфика навигации и информационного поиска в семантическом пространстве.*
- *Многообразие стратегий решения задач.*
- *Универсальные стратегии решения задач и реализующие их программные агенты.*
- *Что такое интеллектуальный решатель задач и каковы направления интеллектуализации решателей задач.*



УДК 004.822:514

СЕМАНТИЧЕСКИЙ ПОИСК КАК СОСТАВЛЯЮЩАЯ УПРАВЛЕНИЯ ЗНАНИЯМИ В SEMANTIC WEB

Рогушина Ю.В.

Институт программных систем НАНУ, г.Киев, Украина

ladamandraka2010@gmail.com

Проанализированы проблемы, возникающие в процессе управления онтологическими знаниями в Web, в частности, связанные с интеграцией знаний из различных источников, извлечением новых знаний из доступной информации и поиском тех знаний, которые нужны пользователю для решения конкретной задачи. Предложены методы автоматизации создания метаописаний информационных ресурсов и персонализации поиска на основе тезаурусов и онтологий, характеризующих предметную область, интересующую пользователя. Предложенные методы реализованы при разработке информационно-поисковой системы МАИПС, в которой поиск персонафицируется на основе агентного подхода и онтологического анализа.

Ключевые слова: онтология, Semantic Web.

ВВЕДЕНИЕ

Сегодня значительная часть современных Web-приложений являются в определенной степени интеллектуальными, то есть каким-то способом используют знание относительно соответствующей предметной области (PrO) и способны сами создавать новые знания.

На современном этапе развития ИТ большинство интеллектуальных Web-приложений используют технологии и стандарты, разработанные в рамках проекта Semantic Web [Хорошевский, 2008]. Управление знаниями в среде Semantic Web требует разработки соответствующих средств получения, сохранения, поиска и использования знаний с учетом таких свойств среды Web, как динамичность и гетерогенность.

Центральным компонентом концепции Semantic Web является использование онтологий, которые позволяют формализовать знания о PrO и, в отличие от XML Schema, являются представлением знаний, а не форматом сообщений. Над онтологиями можно выполнять операции логического вывода. Инструментальные средства обеспечивают создание онтологий и их связывание с различными ИР; проверку онтологий на непротиворечивость, усовершенствование онтологий; выполнение операций логического вывода над онтологиями. В существовавшем ранее программном обеспечении для создания, обработки и использования знаний эти знания не были интероперабельными, и невозможно было перенести базу знаний из одного приложения в

другое без значительных переработок. Но в современных Web-приложениях для представления знаний широко используют онтологии, которые имеют существенный теоретический базис (в частности, дескриптивные логики) и обеспечивают повторное использование знаний в различных ИС.

Сегодня в рамках проекта Semantic Web уже разработан ряд стандартов и языков для управления знаниями (рис.1.).

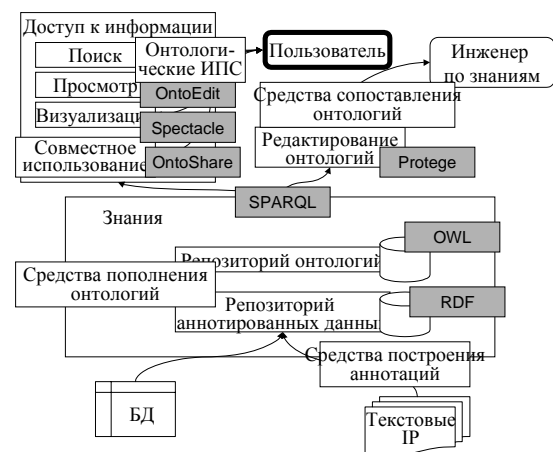


Рисунок 1 – Архитектура управления знаниями на основе Semantic Web

ПРОБЛЕМЫ УПРАВЛЕНИЯ ЗНАНИЯМИ В WEB

Основные проблемы управления знаниями в Web связаны с: *интеграцией знаний*, полученных от разных ИР (например, интеграция онтологий

нескольких разных Про или ИР в одной Про); *поиском противоречий* между знаниями, которые отображены в контенте разных ИР, оценкой их достоверности и надежности; *получением новых знаний* из уже имеющихся и их представлением в форме, понятной пользователю; *поиском знаний, нужных конкретному пользователю* для решения определенных задач; *автоматизацией создания метаданных*, корректно описывающих контент ИР (как текстовых, так и мультимедийных) на уровне содержания, и поиском в таких метаописаниях.

Проблемы, возникающие в процессе управления знаниями в Web, сводятся к следующим четырем.

Проблема 1. Выбор средств представления знаний, достаточно мощных для удовлетворения потребностей пользователей, но пригодных как для быстрой автоматизированной обработки, так и для понимания человеком. Сегодня для представления знаний Про широко применяют онтологии, но до сих пор не предложено общепринятого определения понятия "онтология". Под онтологией Про обычно понимают ту часть знаний Про, которая предполагает ее относительную неизменность и ограничивает значение терминов Про.

Проблема 2. Создание новых знаний по имеющимся ИР (например, создание метаописаний и онтологии ИР, логический вывод, выполнение запросов к БЗ). В современном Web ИР содержат неявные нечеткие и противоречивые знания, а количество ИР вызывают необходимость в автоматизации их извлечения. Наличие языка создания метаописаний (RDF) – необходимое, но не достаточное условие формирования таких метаописаний. Чтобы автоматизировать, например, создание метаописания полнотекстового документа, нужно, во-первых, использовать методы лингвистического анализа, а во-вторых, применять формализованные знания соответствующей Про в интероперабельном представлении (онтологии Про). Кроме того, необходимо разработать специализированные методы индуктивного, традуктивного, дедуктивного вывода, ориентированные на обработку именно таких структур знаний (например, индуктивное обобщение знаний, представленных тройками RDF).

Проблема 3. Сопоставление различных информационных объектов на семантическом уровне (например, интеграция или поиск отличий двух онтологий, сопоставление информационного запроса и ИР, соответствующих этому запросу, определение Про ИР по его контенту). Эта проблема достаточно нетривиальна и не сводится к обычному поиску. Она требует анализа закономерностей Про, наличия средств их формального представления и разработки алгоритма сравнения таких описаний.

Проблема 4. Оценка качества новых знаний (достоверности, непротиворечивости, актуальности, полноты). Это требует анализа разных моделей представления знаний, использования

соответствующего математического аппарата (например, теории высказываний первого порядка) и оценки качества онтологий, рассматривая для этого не только пару «онтология - реальный мир», для которой устанавливается соответствие, но и тройку - «реальный мир - неформализованное знание о мире (существующее соглашение в некотором сообществе) - формальное представление знания о мире».

ОНТОЛОГИИ КАК СРЕДСТВО ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Анализ публикаций показывает, что именно онтологии являются адекватным и эффективным средством для моделирования представлений о разнообразных Про, объектах и информационных ресурсах.

В различных источниках предлагаются разные формальные модели представления онтологий. Во всех них присутствует: 1) множество *терминов* (понятий, концептов), которое может подразделяться на множество классов и множество экземпляров; 2) множество *отношений* между понятиями, в котором могут явным образом выделяться отношения «класс-подкласс», иерархические (таксономические) отношения и отношения синонимии (подобия), а также функции - специальный случай отношений, для которых n-й элемент отношения однозначно определяется n-1 предшествующими элементами; 3) *аксиомы и функции интерпретации* понятий и отношений.

Формально онтология представляется тройкой (X, R, F) , где X - множество концептов, R - множество отношений между концептами, F - функции интерпретации концептов из множества X и отношений с R . Данная модель носит общий характер, в то время, как на практике пользуются более точными моделями. Например, в [Cimiano, 2006] онтология определяется как структура, которая включает идентификаторы концептов, идентификаторы отношений, идентификаторы атрибутов и типы данных, а также иерархию концептов и иерархию отношений.

В [Euzenat, 2007] онтология определяется как кортеж, который, кроме множеств классов, экземпляров, отношений и типов данных, содержит множество значений и ряд отношений (специализацию, исключение, создание экземпляра и присваивание). Для онтологии ее моделью является интерпретация, которая удовлетворяет всем утверждениям онтологии.

Проанализировав выразительные возможности разных средств представления онтологий и формальные модели онтологий, можно утверждать, что существующие технологии Semantic Web предлагают разные средства описания онтологий, которые отличаются по своим выразительным возможностям и по своей сложности: RDF Schemas предоставляет простейший уровень для

представления онтологий, а OWL Full – наиболее сложный.

Выбор средства представления онтологии зависит от специфики проблемы, для которой она разрабатывается.

Таким образом, можно говорить о целесообразности управления знаниями в Web на основе онтологического анализа.

СЕМАНТИЧЕСКИЙ ПОИСК КАК ОСНОВНАЯ СОСТАВНАЯ УПРАВЛЕНИЕ ЗНАНИЯМИ

При современном уровне развития ИТ и Web-технологий задача поиска информации трансформировалась из выявления документов, которые содержат определенные ключевые слова, в поиск знаний, необходимых для решения определенной задачи. Семантический поиск – это процесс поиска документов по их смыслу. В дальнейшем будем рассматривать семантический поиск как вид автоматизированного полнотекстового информационного поиска с учетом смыслового содержания слов и словосочетаний запроса пользователя и предложений текстов проиндексированных информационных ресурсов [Gladun, 2009].

При семантическом поиске, в отличие от обычного, предметом поиска может быть не просто ИР (документ или его фрагмент), а информационный объект определенного класса, то есть пользователь может (явно или неявно) указать класс искомого объекта. Это может быть довольно простой и распространенный класс, например, «человек» или «мультимедийный объект», либо класс, специфический для данной ПрО, например, «научная публикация».

Нередко задача семантического поиска состоит не просто в нахождении объекта определенного класса, который удовлетворяет ряду условий (например, «научный сотрудник», который работает в «организации ХХХ» возрастом до 30 лет), а в выявлении набора объектов разных классов, которые находятся в заданных отношениях и удовлетворяющих условия поискового запроса. При этом нередко надо использовать правила и закономерности ПрО поиска. Например, относительно простую задачу «найти группы людей разного возраста, которые проживают по одному адресу» довольно легко трансформировать в запрос «найти семьи», но чтобы запрос «разработать за 2 месяца программу автоматизации работы отдела №13, которая поддерживает одновременное обслуживание до 100 клиентов» был трансформирован в «создать группу из 3 человек в составе Иванова, Петрова и Сидорова для написания программы на С++ со следующими функциями ...», нужно применить большое количество правил, которые описывают и компетенции исполнителей, и функции отдела, и

свойства программирования. Тем не менее в перспективе такие запросы тоже должны выполняться автоматизированно.

Такие сложные запросы пользователя (следует отметить, что под пользователем следует понимать как человека, так и некую программную сущность, которая имеет цели и намерения) намного труднее формализовать: 1) пользователь должен описать ту проблему, для решения которой ему нужны искомые сведения; 2) надо знать, какие именно сведения уже есть у пользователя в наличии; 3) необходимо понимать, какую информацию (в какой форме, какого уровня сложности и т.п.) пользователь способен воспринять и обработать; 4) информация может содержаться в доступных при поиске документах неявно (то есть необходимо сначала выполнить над ней операции логического вывода, обобщение, сравнение и т.п.).

Таблица 1 – Сравнение традиционных и семантических ИПС

	Традиционные ИПС	ИПС, использующие семантический поиск
Запрос	набор ключевых слов	Информационная потребность в сведениях определенной ПрО
Персонализации поиска	История запросов пользователя	Модель пользователя и его информационных потребностей
Результат поиска	Документ, содержащий ключевые слова запроса	Знания, извлеченные из документов, релевантных запросу и описывающих интересующий пользователя объект
Источник сведений об ИР	Индекс ИПС	Индекс ИПС и метаданные о доступных ресурсах
Описание ПрО	-	Онтология ПрО

Рассмотрев средства представления метаданных о гетерогенных (в том числе и мультимедийных) ИР, которые публикуются в распределенной динамической среде Web, можно сделать вывод о том, что, несмотря на многообразие подходов к отображению семантики информационных ресурсов, на современном уровне развития информационных технологий в большинстве случаев наиболее релевантным остается информационный поиск по ключевым словам. Эффективность такого поиска довольно низка, однако ее можно значительно повысить за счет использования контекста запроса и сведений о конкретном пользователе, который предоставляет запрос, и об его специфических информационных интересах, и учета предыстории его обращений с запросами к этой ИПС [Гладун, 2009]. Для

определения контекста запроса представляется целесообразным использовать онтологический подход к описанию предметной области поиска.

Как показывает анализ публикаций, один из перспективных подходов к задаче контекста поиска базируется на онтологиях, которые содержат перечень основных терминов ПрО, связи между ними и правила вывода.

ИСТОЧНИКИ ОНТОЛОГИЧЕСКИХ ЗНАНИЙ ДЛЯ СЕМАНТИЧЕСКОГО ПОИСКА

На сегодня задача поиска извлечения из ресурсов необходимых пользователю знаний остается одной из наиболее актуальных в ИТ. При этом уже разработан ряд средств и методов, которые могут использоваться для решения этой задачи. Средствами пополнения онтологических знаний являются: непосредственное (не автоматизированное) построение онтологии или тезауруса специалистом предметной области, автоматизированная обработка метаданных об ИР, получение онтологических знаний из естественных языковых текстов, применение методов индуктивного вывода и логические операции над существующими онтологиями.

Пользователю нужно создать онтологию той области, к которой относятся его информационные интересы, чтобы потом использовать ее при поиске наиболее пригодных ИР. Эта довольно сложная задача. Конечно, пользователь может использовать какую-то общую онтологию, которая была создана ранее другими исследователями и покрывает область его интересов. Но из-за сложности структуры и большого объема таких онтологий пользователю тяжело вносить в них изменения и дополнения. Кроме того, общие онтологии могут не соответствовать убеждениям и знаниям пользователя и не отображать его персональные предпочтения. С другой стороны, для самостоятельного создания онтологий пользователь должен не только четко представлять себе структуру интересующей его ПрО, основные ее понятия и связи между ними, но и обладать знаниями и привычками инженера по знаниям.

Поэтому, кроме редактора онтологий, который позволяет пользователю непосредственно строить онтологию, целесообразно предоставить ему определенные программные средства, способные помочь пользователю в формализации его знаний в виде онтологий и тезаурусов. Методы индуктивного вывода и лингвистический анализ позволяют извлекать термины и связи между ними из документов ПрО. Оба подхода дополняют друг друга.

Тезаурус является частным случаем онтологии, его проще формировать и обрабатывать. Тезаурус – это полный систематизированный набор данных о любой области знаний, который позволяет человеку или компьютеру в ней ориентироваться.

В связи с необходимостью анализа большого количества ИР предлагается использовать упрощенный алгоритм построения их тезауруса: по полному перечню слов, используемых в ИР, строится словарь терминов, из которого удаляются слова, помещенные в специально разработанному пользователем список («стоп»-слова). Этот алгоритм применяется только для тех ИР, которые не сопровождаются метаописаниями. В противном случае из метаописаний (в формате RDF или OWL) приобретаются термины тезауруса и связи между ними, которые дополняют построенный по контенту ИР словарь. Потом тезаурус ИР сравнивают с тезаурусом пользователя.

Пополнение онтологии ПрО может осуществляться также и в результате лингвистического анализа текстов, выбранных пользователем в соответствии с представлениями о своих информационных потребностях. В результате семантико-синтаксического анализа в документах пользователя выделяются классы онтологии, экземпляры классов и отношения между классами и экземплярами. Для этого необходимо установить соответствие между фрагментами ЕЯ-текста (словами и словосочетаниями) и элементами онтологии (классами и отношениями).

Чтобы формализовать эту связь, создается и пополняется специальная лексическая онтология. В лексической онтологии хранятся словоформы для классов, экземпляров классов и отношений онтологии ПрО. Например, отношению «состоит из» соответствуют такие фрагменты ЕЯ-текста, как «сделан из», «изготовлен из», «содержит в себе».

Если в одном предложении встретились два фрагмента, которые связаны с экземплярами классов лексической онтологии, но в онтологии ПрО не зафиксированы отношения между этими классами, то необходимо спросить пользователя о необходимости пополнения онтологии ПрО новым отношением. Онтология пополняется новым классом, если в одном абзаце текста есть фрагменты, связанные с уже существующими классами, а также фрагменты, связанные с отношением. Тогда в предложении выделяется словоформа для нового класса онтологии.

Стандарты Semantic Web обеспечивают интероперабельное представление знаний в Web: RDF позволяет создавать метаописания ИР, в которых явным образом описывается их семантика; OWL позволяет представлять знания о ПрО в виде онтологий, которые можно использовать и обрабатывать в различных приложениях; URI позволяет однозначно идентифицировать различные ресурсы (причем не только ресурсы Web, но и абстрактные понятия и объекты реального мира) для их автоматизированной обработки; язык запросов SPARQL позволяет извлекать из метаописаний RDF и онтологий OWL необходимые пользователю сведения.

Для представления и обработки OWL

существует теоретический базис в виде семейства логик DL, обеспечивающий доказательность логического вывода на онтологиях, а различные ризонеры позволяют осуществлять на структурированных данных (OWL и RDF) логический вывод.

Уже существует достаточно большое количество структурированных метаданных, описывающие различные типы объектов, которые базируются как на стандартах Semantic Web, так и на социальных сетях (FOAF и т.д.), а также создано достаточное количество онтологий разного уровня и объема, формализующие знания самых разных ПрО.

Разработан ряд методов и инструментов для автоматизированного построения онтологий и тезаурусов по полнотекстовым ИР, существуют средства сопоставления запросов и ресурсов, ориентированные на семантический поиск Web-сервисов, которые могут использоваться и для поиска других типов ресурсов, а также есть методы и средства сопоставления онтологий (например, онтологии запроса пользователя и онтологии ИР).

Существуют ИПС, ориентированные на поиск среди структурированных данных (в частности, представленных в форматах OWL и RDF).

Но есть и ряд проблем: основная часть ИР, представленных сегодня в Web, не сопровождаются метаданными RDF (а если и сопровождаются, то доверие к этим метаданным остается открытым вопросом); построение онтологий ИР может быть автоматизировано только частично и в любом случае требует участия человека на ряде этапов, оставаясь при этом достаточно длительным и трудоемким процессом; процесс сопоставления двух независимых онтологий является сложной и трудоемкой процедурой.

Таким образом, самое простое и очевидное решение проблемы семантического поиска – построить репозиторий семантических метаописаний всех доступных в Web ИР, а потом сопоставлять их с запросом пользователя, также представленным в виде онтологии, – на сегодня не может быть реализовано. Тем не менее можно предложить следующее альтернативное решение:

Этап 1. Строится формальная модель информационной потребности пользователя (на основе онтологии интересующей его ПрО, онтологии самого пользователя и т.д.);

Этап 2. Эта модель сопоставляется с доступными структурированными данными (например, онтологиями и метаописаниями ИР), причем подобное сопоставление включает: 1) поиск соответствующих онтологий – с помощью специализированных ИПС или в собственном репозитории онтологий; 2) пополнение онтологии запроса; 3) элементы классификации запроса;

Этап 3. Постановка задачи трансформируется (в сторону конкретизации, расширения, связи с

конкретными URI и т.д.) на основе анализа этих структурированных данных, и трансформированный запрос передается с помощью МАИПС к внешним ИПС;

Этап 4. Анализируются метаданные и естественно-языковой контент найденных ИР, строятся (или пополняются) онтологии этих ИР, которые затем сопоставляются с онтологией запроса;

Этап 5. Полученные результаты выполнения запроса упорядочиваются с учетом формальной модели (онтологии) запроса и знаний, полученных на предыдущем этапе.

Преимуществами предложенного двухэтапного семантического поиска является:

- используются уже существующие онтологические базы знаний, связанные с предметной областью поиска;
- для поиска соответствующих знаний используются уже существующие поисковые механизмы;
- поиск осуществляется не только среди ИР, сопровождаемых метаданными, а среди всего контента Web;
- построение онтологий ИР и их сопоставление с онтологией запроса надо осуществлять только для относительно небольшого подмножества ИР;
- при построении онтологии запроса можно использовать знания, накопленные при выполнении предыдущих запросов.

Система семантического поиска МАИПС

Мультиагентная информационно-поисковая система МАИПС с развитыми средствами интеллектуализации ее поведения, которая детально описана в [Рогушина, 2010], функционально направлена на выполнение сложных многообразных запросов в довольно узких областях, связанных с профессиональными или научными интересами пользователей, и предоставляет пользователю высоко релевантные результаты поиска. Такие результаты достигается благодаря ориентации системы на пользователей, имеющих в сети постоянные информационные интересы и требующих постоянного поступления соответствующей информации – для этого МАИПС позволяет сохранять и повторно выполнять запросы, отслеживать появление аналогичных запросов у других пользователей, сохранять формальное описание области интересов пользователя в виде онтологии и т.д.

Особенностью МАИПС является интегрированное использование ряда семантических технологий. Основой МАИПС являются технологии Semantic Web, в частности, язык представления онтологий OWL и средства его обработки. Для представления знаний об

интересующей пользователя ПрО используются онтологии и тезаурусы ПрО, а теоретико-множественные операции над тезаурусом позволяют более точно описывать нужную ПрО. Реализована генерация тезаурусов по естественноречевым текстам, которые описывают семантику этих ИР, а объединение таких тезаурусов позволяет формировать тезаурус ПрО.

В МАИПС применяются технологии Web 2.0: облака тэгов используются для визуализации поисковых тезаурусов, а социальные сервисы позволяют осуществлять взаимодействие между пользователями системы со схожими интересами.

Для упорядочения информационных ресурсов, найденных системой, разработаны оригинальные алгоритмы, работающие с учетом веса (значимости) онтологических терминов для конкретного запроса. Критерии оценки уровня читабельности текста применяются для поиска информации, которая соответствует персональным потребностям пользователя [Рогущина, 2007]. Методы индуктивного вывода позволяют обобщить опыт работы МАИПС.

Для формализованного описания поведения системы используется мультиагентный подход к созданию модели интеллектуальной информационно-поисковой системы и представление компонентов системы как интеллектуальных BDI-агентов, а парадигма интеллектуальных Web-сервисов – для описания функций агентов системы для их интероперабельного использования.

Пользователь может обращаться к онтологиям, созданным другими пользователями – пересматривать их, задавать за ними контекст поиска, копировать из них нужные фрагменты, но не имеет права изменять их. ИПС может обеспечить поиск онтологий, которые содержат введенные пользователем термины, а также поиск онтологий, похожих на выбранную пользователем онтологию. Это позволяет создавать группы пользователей с общими информационными интересами и предотвратить дублирование в выполнении одинаковых многообразных запросов разных пользователей.

ЗАКЛЮЧЕНИЕ

В работе представлен подход к осуществлению семантического поиска среди ресурсов Web, который, с одной стороны, направлен на максимальное использование имеющихся средств представления и обработки распределенных знаний, а с другой – учитывает недостаточную наполненность существующего контента Web соответствующими семантическими метаописаниями и потому представляет собственные средства и методы для создания и обработки знаний о ресурсах и пользователях. В

качестве основного средства представления знаний в работе используются онтологии и тезаурусы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Cimiano, 2006] Cimiano P. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. – Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006. – 347 p.
- [Euzenat, 2007] Euzenat J., Shvaiko P. *Ontology matching*. – Springer-Verlag Berlin Heidelberg, 2007. – 332 p.
- [Gladun, 2009] Gladun A., Rogushina J. Use of Semantic Web technologies in design of informational retrieval systems // in Book “Building and Environment”, 2009 Nova Scientific Publishing, New-York, USA. – P.89-103.
- [Гладун, 2009] Гладун А.Я., Рогущина Ю.В. Использование технологии Semantic Web для интеллектуального управления в динамических распределенных системах // *International Book Series “Information Sciences and Computing”*, 2009, Varna, Bulgarien. – P.143-153.
- [Рогущина, 2007] Рогущина Ю.В. Показатели индивидуальной легкости чтения текста как критерий поиска информационных ресурсов в сети Интернет // *УСиМ*, № 3, 2007. – С.76-84.
- [Рогущина, 2010] Рогущина Ю.В., Гришанова И.Ю. Літературний твір наукового характеру “Модель мультиагентної інформаційно-пошукової системи “МАИПС”(“Модель МАИПС”). – Свідчення про реєстрацію авторського права на твір №32068, 2010.
- [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.

SEMANTIC SEARCH AS A COMPONENT OF KNOWLEDGE MANAGEMENT IN SEMANTIC WEB

Rogushina J.V.

Institute of software systems of National Academy of Sciences of Ukraine, Kiev, Ukraine

ladamandraka2010@gmail.com

The problems of the ontological knowledge management in the Web are analysed, in particular relating to the integration of knowledge from various sources, acquisition of new knowledge from the available information and the retrieval of knowledge that the user needs for some purpose. Methods of automatized creation of meta-description of information resources and of search personalization based on thesauri and ontologies, describing the subject area of interest to the user, are proposed. The proposed methods are implemented in the development of an IRS MAIPS where the search is personified on base of agent approach and the ontological analysis.

Keywords: ontology, Semantic Web.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.9:510

ПОДХОД К СЕМАНТИЧЕСКОМУ ПОИСКУ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ В НАУЧНЫХ ТЕКСТАХ

Биряльцев Е.В.^{*}, Галимов М.Р.^{*}, Жильцов Н.Г.^{*}, Невзорова О.А.^{**}

^{} Казанский (Приволжский) федеральный университет,
г. Казань, Россия*

IgenBir@yandex.ru

glmvmrt@gmail.com

nikita.zhiltsov@gmail.com

*^{**} Научно-исследовательский институт «Прикладная семиотика» АН Республики Татарстан,
г. Казань, Россия*

onevzoro@gmail.com

В работе рассматривается подход к семантическому поиску математических выражений, позволяющий выполнять запросы на поиск математических формул по текстовым наименованиям переменных, входящих в формулы. Обсуждается прототип системы поиска и методы решения основных проблем предложенного подхода.

Ключевые слова: семантический поиск, математический поиск, разметка формул.

ВВЕДЕНИЕ

Поиск по математическим документам [1] – актуальная и быстроразвивающаяся область исследований. Современные математические поисковые системы условно можно разделить на три группы. К первой относятся системы поиска научных публикаций и поисковые интерфейсы крупнейших научных коллекций, которые предлагают сервис полнотекстового поиска по ключевым словам с учетом метаданных публикации (автор, название, журнал, краткое описание). Эти системы индексируют значительные объемы актуальных научных статей в области математики в формате PDF или LaTeX. К числу таких систем относятся хорошо известные системы, такие как Google Scholar [Google], Microsoft Academic Search [Microsoft], CiteseerX и Scirus. Полнотекстовый поиск по ключевым словам достаточно удобен для конечного пользователя.

Отличительная особенность систем второй группы состоит в том, что они используют семантику математической нотации и реализуют поиск по формулам и выражениям. Данные системы работают со специальным семантическим представлением математических формул, выраженным на языках Content/Presentation MathML и OpenMath. В качестве результатов поиска

возвращаются ссылки на документы, содержащие релевантные формулы. Основными трудностями при разрешении запросов данного типа являются: зависимость форм нотации от контекста; неоднозначность трактовки одних и тех же символов; идентичность формул с точностью до обозначений. Специализированные системы поиска по математическим формулам предлагают средства для формулирования запроса в синтаксисе языка разметки LaTeX (например, Springer LaTeXSearch [LatexSearch] индексирует базу статей издательства Springer; (uni)quation [Uniquation] — научные тематические сайты, форумы и Wikipedia) или MathML, используя соответствующие графические интерфейсы [MathWebSearch].

Третья группа – приложения], в основном, представляющие собой расширения систем автоматических доказательств, использующие строго формализованное представление математических документов на таких языках, как Mizar и Coq. Они позволяют выполнять сложные семантические запросы, например, искать теоремы для применения в данном доказательстве.

Общее направление развития современных поисковых технологий ориентировано на широкое привлечение семантики, учитывающей особенности предметной области, для повышения качественных характеристик поиска.

Подход, представленный в настоящей статье, направлен на интеграцию функциональных возможностей полнотекстового поиска и поиска по математическим формулам, при котором конечному пользователю предлагается формулировать поисковый запрос на поиск математической формулы в форме ключевых слов. Наиболее близким подходом является подход математической поисковой системы EgoMath (доступна по адресу <http://egomath.projekty.ms.mff.cuni.cz>), которая в данный момент предоставляет возможности традиционного формульного поиска в синтаксисе LaTeX и механизм переформулирования запроса (от ключевых слов к символьным обозначениям), однако алгоритм этого связывания не раскрыт в оригинальной статье авторов EgoMath [Misutka et al., 2008]. Новизна предлагаемого подхода состоит в том, что основной метод связывания рассматривает локальный контекст формулы, включающий текстовое расширение и тип структурного элемента (например, теорема, доказательство или раздел), содержащего математическое выражение. Это позволяет, с одной стороны, более точно определять семантику переменных в формуле и, с другой стороны, представлять пользователю релевантный фрагмент документа в выдаваемых результатах. Как правило, возможность выдачи более точно локализуемых положений формулы также не предоставляется поисковыми системами, рассмотренными выше.

Кратко рассмотрим основные подходы к поиску математических выражений.

Известна постановка задачи поиска математических формул по ее фрагментам, например, найти все формулы, в которые входит конструкция " $X^n + 1$ ". Данный подход в настоящее время доведен до уровня экспериментальных систем [wolframalpha.com], [<http://uniquation.ru/ru/>]. Несмотря на внешнюю простоту данной задачи, ее решение в классических полнотекстовых системах затруднено. Если мы хотим указать достаточно сложную математическую конструкцию, то нам необходимо иметь возможность использовать в запросах некоторый язык разметки математических формул. Таким образом, задачу поиска математических формул необходимо решать в рамках контекста некоторой группы языков разметки, причем как на уровне интерфейса пользователя, так и внутренних механизмов индексирования и выполнения запросов.

Для наиболее популярных языков математической разметки TeX и MathML известны решения по конвертации TeX/LaTeX в MathML и частичных обратных преобразований [http://www.w3.org/Math/Software/mathml_software_cat_converters.html]. В рамках настоящей работы мы будем рассматривать только запросы к текстам, содержащим математические формулы в символьном виде с использованием языков математической разметки. При использовании языков разметки презентационного уровня, таких

как TeX или MatML презентационного уровня, возникают некоторые проблемы, связанные с семантикой математических формул, в частности, не определяются свойства операций (коммутативность, ассоциативность, транзитивность), что влияет на выполнение эквивалентных формульных преобразований. Для приведенного выше примера семантика операции "+" может допускать или не допускать коммутативность, поэтому выражения " $X^n + 1$ " и " $1 + X^n$ " могут быть семантически эквивалентными или неэквивалентными. Семантический уровень математической разметки, такой как MathML семантического уровня или OpenMath [<http://www.openmath.org/>], частично снимает эту проблему. Фиксированный (MathML) или расширяемый (OpenMath) набор математических операций позволяет однозначно описать и разобрать описание сходных (по написанию) математических конструкций. Можно встроить в поисковые механизмы алгоритмы, учитывающие свойства операций, что повышает релевантность ответов на запрос, т.е. в ответах выдаются ссылки на документы, содержащие формулы вида " $X^n + T + 1$ ", " $1 + X^n$ ", но не " $N^x + 1$ " (коммутативность и ассоциативность операции "+" и некоммутативность операции возведения в степень).

Дальнейшее рассмотрение темы полноты и релевантности поиска математических формул по их фрагментам приводит в необходимости решения вопроса семантической эквивалентности выражений " $X^n + 1$ " и, например, " $Y^n + 1$ ", т.е. вопроса об эквивалентности математических выражений в различных системах обозначений переменных. Здесь необходимо различать, что собственно ищет пользователь - абстрактную математическую конструкцию или в переменные вкладывается некоторый нематематический смысл. Очевидно, что если речь идет о поиске собственно математических конструкций с некоторой структурой заданной последовательностью математических операций, то выражения " $X^n + 1$ " и " $Y^n + 1$ " эквивалентны. В противном случае необходимо рассматривать нематематическую семантику переменных, участвующих в поисковых запросах.

Данную интерпретацию можно рассматривать как новый тип запросов к коллекциям математических документов. Действительно, в отличие от рассмотренной задачи поиска математической формулы по ее фрагменту в формулировке запроса должны использоваться не математические конструкции, а словесные наименования переменных, входящих в искомую формулу. Результатом поискового запроса также должен быть тематический реферат найденного текста, содержащий искомую формулу и определения обозначений использованных в ней имен переменных. В отличие от полнотекстовых запросов результатом поиска является не просто фрагменты документа из коллекции, содержащие слова строки запроса в определенном количестве и

находящиеся в достаточной близости, а таковые фрагменты, содержащие нетекстовый объект (формулу) и фрагменты документа, связывающие переменные формулы и их текстовые определения вне зависимости от их местонахождения в тексте. Для выполнения поискового запроса нам необходимо выделить в анализируемом тексте обозначение переменных по их текстовому описанию и формулы, в которых данные переменные представлены наиболее полно. Таким образом, данный тип запроса не сводится ни к поиску формул по их фрагментам, ни к полнотекстовому поиску, и требует рассмотрения возможности его реализации, особенно в части методов разметки и индексирования коллекций.

В качестве иллюстрации рассмотрим математическую конструкцию вида $a = bc^2$. Как минимум, две популярные формулы имеют такую структуру: формула вычисления площади круга $S = \pi R^2$, и формула, связывающая массу и полную энергию, $E = mc^2$. Поиск только по математическому контексту осложняется тем, что в случае с естественнонаучными формулами мы не можем просто указать, что требуется найти формулу, содержащую переменные S и R, или E и m, так как многие символы перегружены и имеют в различных областях науки различный смысл. Так S - это площадь, пройденный путь, энтропия, а также символ химического элемента серы; E - символ энергии, напряженности электромагнитного поля, прописная e - основание натуральных логарифмов и т.п. Кроме того, в новых и специальных областях науки и техники устоявшиеся обозначения могут отсутствовать, различные научные школы могут придерживаться различных обозначений. Словесная терминология, как правило, более устойчива, в научных текстах "энергия" и "площадь" однозначно обозначают соответствующие физические величины. Таким образом, для поиска конкретной формулы необходимо задать запрос, в котором явно указывается, что требуется найти формулу, связывающую "радиус круга" и "площадь", или формулу с параметрами "энергия", "масса" и "скорость света".

1. Подход к семантическому поиску математических выражений

В рассматриваемой постановке задачи в естественнонаучных текстах выделяются следующие виды сущностей: естественнонаучные термины, символьные условные обозначения терминов, математические фрагменты (формулы) и структурные элементы документа (теоремы, доказательства и пр.), содержащие формулы. Все перечисленные сущности составляют расширенный контекст формулы.

Таким образом, для решения поставленной задачи необходимо вычислять в тексте расширенный контекст формулы, который включает

выделение отношений: «термины - условные обозначения» и «условные обозначения - формулы». Первое отношение есть текстовое определение значения символа в некотором контексте с помощью терминов, второе отношение указывает на вхождение символа в формулу. Пример расширенного формульного контекста приведен на рис. 1, который содержит фрагмент статьи Wikipedia о площади треугольника. Из рисунка видно, что определения переменных a, b, c как длин сторон треугольника и α, β, γ как углов треугольника даны в структурном элементе «примечание», а формула, которая связывает данные переменные, находится в структурном элементе «теорема синусов». Таким образом, локальный контекст формулы включает структурный элемент типа «теорема», а также структурный элемент типа «примечание», в котором содержится спецификация семантики формульных переменных. Эта информация отображается в поисковых результатах, что упрощает задачу пользователя по визуальной оценке релевантности найденной формулы.

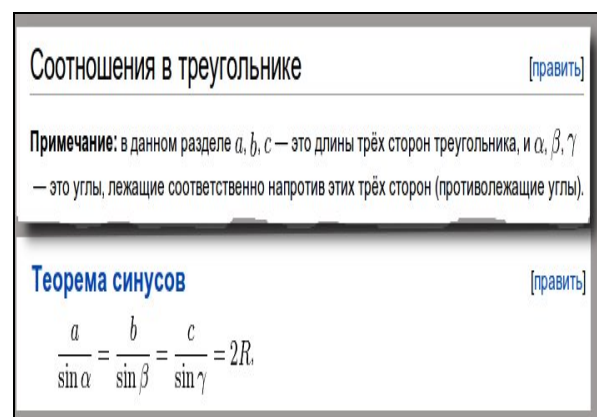


Рисунок 1 – Пример расширенного формульного контекста (фрагмент статьи Wikipedia)

Фиксация выделенных сущностей и зависимостей между ними может представляться как в виде разметки (внесения дополнительных символьных конструкций непосредственно в анализируемый текст), так и построением соответствующих индексов (внешних файлов по отношению к анализируемому тексту). В статье детально рассматривается метод аннотирования, так как построение индексов является более техническим аспектом, связанным с быстродействием, объемами данных и др. аспектами реализации, кроме того индекс всегда может быть построен по разметке. Затем рассматриваются концептуальные аспекты подхода к поиску. При этом технические решения реализации поиска могут существенно различаться в зависимости от объемов коллекции, структуры запросов к ней и других факторов. В частности, на реализацию могут существенно повлиять возможности полнотекстовой поисковой машины, используемой совместно с рассматриваемыми специализированными поисковыми алгоритмами, требования к полноте и релевантности результата,

наличие других разметок и индексов, например разметка логической структуры естественнонаучного текста.

1.1. Разметка и индексирование

Математические выражения в естественнонаучных текстах в настоящее время задаются с использованием специализированных языков разметки (MathML презентационного и семантического уровня; Tex, LaTeX, MathType и их конверторы в MathML; OpenMath и др.). Рассмотрим цель дополнительной разметки в рассматриваемой задаче. Как отмечалось выше, связующим звеном между элементами поискового запроса (текстовыми определениями некоторых переменных и результатом поиска в виде математической формулы) является набор условных обозначений данных переменных. Эти условные обозначения представляют собой фрагменты файла с MathML-разметкой. При анализе разметки необходимо выяснить какие из фрагментов являются (или могут являться) математическими формулами, какие - символьными обозначениями переменных, а какие - не относящимися к рассматриваемому вопросу конструкциями, например, таблицами или диаграммами.

Разметка MathML презентационного уровня включает теги, позволяющие представить в математических выражениях все необходимые символы и расположить их в соответствии с правилами записи соответствующей математической нотации. Элементарными конструкциями в MathML презентационного уровня являются токены (символьные последовательности, ограниченные тегами). Например, `<mi>` - идентификатор; `<nm>` - число; `<ms>` - литерал; `<mo>` - оператор, ограничитель, разделитель; `<mtext>` - текст (комментарий). Символы внутри тегов могут представляться обычными ASCII-символами, либо шестнадцатеричными кодами Unicode, что позволяет использовать все необходимые специальные символы. Остальные теги позволяют управлять размещением токенов относительно друг друга, в том числе формировать дроби, индексы, надстрочные и подстрочные символы (например, пределы интегрирования), матрицы, и управлять отступами между токенами.

Тег `<mi>` ограничивает многосимвольные описания идентификаторов, при этом описание сложной переменной (с индексами, подстрочными и надстрочными символами) никак не ограничивается. Таким образом, имена используемых в формулах переменных необходимо вычислять путем сложного контекстного анализа (не всегда однозначного). В MathML презентационного уровня также отсутствует явное указание на то, является ли некоторый фрагмент описанием уравнения или описанием переменной. Можно использовать атрибут `class` тега `<mo>`, который может принимать значения *open*, *close*, *op*,

rel для обозначения открывающих и закрывающих скобок различного типа, операций (в том числе имеющих буквенное написание) и отношений. Наличие тега `<mo>` с атрибутом `class` позволяет сделать вывод, что фрагмент описания не является определением переменной, а представляет более сложную математическую конструкцию.

Если трактовать запросы рассматриваемого типа как поиск уравнений, а не просто математических выражений, то можно еще более сузить класс релевантных запросу объектов, рассматривая только те математические конструкции, в которых присутствует некоторое отношение (как правило, знак равенства). Этот признак также может быть выделен из разметки путем анализа атрибута `class` тега `<mo>`.

Таким образом, на первом проходе при анализе разметки можно выявить фрагменты размеченного текста, предположительно содержащие имена переменных в окружении сплошного текста и фрагменты, предположительно содержащие некоторые уравнения или отношения.

На втором проходе делается попытка установить связь между "отдельно стоящими" именами переменных и фрагментами - кандидатами в формулы. Данный этап позволяет отсеять отдельные фрагменты, не связанные с формулами, и определить принадлежность переменных формулам. Установление принадлежности можно проводить прямым поиском вхождения выделенных фрагментов (кандидатов в имена переменных) во фрагменты (кандидатов в уравнения). Результатом анализа является взаимная разметка фрагментов установленными связями.

Разметка MathML семантического уровня включает значительно большее количество тегов. Для рассматриваемой задачи важно, что на данном уровне присутствуют теги, позволяющие явно описать сколь угодно сложную переменную (тег `<ci>`), и тег `<apply>`, явно указывающий на то, что его содержимое является математическим выражением. В последнем случае, если требуется выделить только выражения, содержащие отношения, в частности, отношение равенства, необходимо анализировать наличие тега `<eq>` (для выделения уравнений) и тегов других отношений (в более общем случае).

Таким образом, выделение фрагментов, содержащих описания переменных и описания математических выражений (в частном случае, уравнений), при наличии разметки семантического уровня упрощается и становится более определенным. Вместе с тем, это не отменяет выявления связей между переменными и формулами, аналогично соответствующему этапу анализа разметки презентационного уровня. Установление связей между фрагментами (определениями переменных) и фрагментами (формулами) выполняется поиском соответствующих вхождений. Результатом анализа

является взаимная разметка установленными связями.

Более богатый синтаксис MathML семантического уровня ставит некоторые вопросы, отсутствующие на презентационном уровне. Так, наличие тегов `<declare>` и `<lambda>`, позволяющих вводить пользовательские типы переменных и функций, требует дополнительного анализа. В простейшем случае можно игнорировать все описания пользовательских типов переменных и функций, так как они описывают математическую семантику вводимых пользователем обозначений. Но, вполне вероятно, что из семантики пользовательских определений можно извлечь дополнительную информацию для решения рассматриваемых задач.

Формализация результатов взаимной разметки может быть произведена в рамках синтаксиса семантического MathML. Для этого можно использовать тег `<semantics>`, который включает первый атрибут как произвольное высказывание семантического MathML, и произвольное количество последующих атрибутов как произвольные текстовые или XML-последовательности, при этом семантика данных атрибутов не регламентируется MathML, а интерпретация осуществляется обрабатывающими программами. Различать соответствующие семантики позволяет атрибут *encoding* элемента тега `<annotation>`, в котором указывается имя обрабатывающей программы или иной ключ, позволяющий определять, как интерпретировать содержащуюся в данном атрибуте информацию. Таким образом, в качестве первого атрибута указывается найденный фрагмент текста MathML (формула), а качестве дополнительных атрибутов фиксируются ссылки на соответствующие формулы, задающие описания переменных. Без ограничения общности, на данном этапе рассматривается двусторонняя разметка (от формул к переменным и наоборот), также пока является открытым вопрос о формате записи взаимных ссылок (через смещения относительно друг друга, указания смещения от начала файла или каким другим способом).

Таким образом, построенная аннотация на основе математической разметки позволяет эксплицитно неявную информацию о связях между фрагментами (описаниями переменных) и фрагментами (формулами) и зафиксировать ее в виде взаимной системы ссылок. Процедура разметки включает операцию анализа взаимного вхождения фрагментов с вычислительной сложностью порядка n^2 , где n - количество выделенных фрагментов, поэтому выполнение запроса «на лету» (без предварительного аннотирования текста) является весьма трудоемким. Другой важной задачей является задача установления связей между синтаксически тождественными определениями переменных. Действительно, тождественные фрагменты MathML-текста, идентифицированные как описания

переменных, могут встречаться в тексте неоднократно. На каждое из таких вхождений, согласно рассматриваемой концепции, требуется зафиксировать ссылку в соответствующей формуле, что многократно увеличивает объем ссылок. В этой ситуации желательно провести некоторую "нормализацию" полученных отношений, составив словарь определений переменных с последующим его использованием для формирования ссылок в формуле. Для решения этой задачи требуется получить семантическую интерпретацию фрагмента-определения переменной в тексте. Определение включает словесное описание некоторой символической конструкции, которая в дальнейшем может многократно использоваться в формулах. В то же время, появление той же символической конструкции вне формулы может означать переопределение ее смысла, а также обсуждение некоторых свойств данной величины или математической абстракции. Априори практически невозможно разделить эти случаи, поэтому создание единого словаря, в котором, возможно, смешиваются случаи вхождения величин, имеющих разный смысл, но одинаковое обозначение, представляется на данном этапе анализа преждевременным.

Рассмотрение вышеуказанных проблем ставит задачу о возможности и целесообразности предварительной семантической разметки. Семантическая разметка предполагает маркирование текстовых фрагментов элементами фиксированного (или расширяемого) словаря (тезауруса, онтологии) и важным моментом является выбор необходимого семантического уровня представления. Так, использование тематических классификаций достаточно затруднительно в связи с ориентацией на широкий класс предметных областей, а применение высокоуровневых онтологий имеет слишком высокий уровень абстракций. Возможным решением является использование онтологии структурных элементов [Solovyev et al., 2010], разработанной авторами статьи в стиле подхода OMDoc [<https://trac.omdoc.org/OMDoc>].

Важнейшей разработкой последних лет является основанный на XML формат математических документов OMDoc, предназначенный для детализированного представления содержимого математических текстов и впоследствии расширенный на другие разделы науки. Формат OMDoc выделяет три уровня структурной семантической разметки. Первый уровень – уровень объектов. Для этого OMDoc содержит выразительные средства для разметки формул в форматах OpenMath и MathML. Второй уровень – уровень утверждений – для выделения теорем, лемм, определений, ссылок. Наконец, третий уровень – уровень теорий, который предоставляет средства для выражения связей между прикладными математическими теориями. Связанная с форматом OMDoc онтология, реализованная на языке OWL-DL, концептуально выражает терминологию

данного формата и содержит описание структурных элементов математического документа как классов и отношений между ними.

Группой исследователей из Бременского университета разработаны методы для полуавтоматического извлечения структурных элементов из математических текстов. Основная технология выглядит следующим образом. Исходный текст математического документа аннотируется с помощью команд макропакета sTeX [Kohlhase M., 2005], который представляет собой расширение популярного в математическом сообществе пакета AMS-LaTeX. С помощью программной утилиты LaTeXML [Miller B., 2007], которая использует мэппинг между командами sTeX и тэгами OMDoc, исходный текст транслируется в формат OMDoc. Далее из математического документа в формате OMDoc автоматически извлекаются RDF триплеты – утверждения вида субъект-предикат-объект в терминах онтологии OMDoc [Lange C., 2009]. Таким образом, в полуавтоматическом режиме происходит обогащение семантической информации о структуре исходных текстов. Другая технология представления математического документа, развиваемая на базе OMDoc, предполагает представление исходных математических текстов в формате XHTML страниц с семантическими вставками в виде RDFa аннотаций [RDFa, 2008]. Это позволит будущим реализациям семантических поисковиков извлекать дополнительные метаданные непосредственно из страниц, опубликованных в вебе.

Подход OMDoc, предусматривающий разметку высокоуровневых структурных элементов математического текста, таких как теорема, доказательство, определение, пример и др., и связей между ними, позволяет наметить решение указанной выше проблемы разрешения смысла многократных вхождений переменной в сплошной текст. Математическая культура диктует употребление переменных в рамках рассмотрения одного вопроса (теоремы, примера и т.п.), и только в одном смысле. Таким образом, в границах выделенного структурного элемента можно уверенно предполагать, что формульный контекст является постоянным, и символическая конструкция, обозначающая переменную, в рамках одного структурного элемента имеет один смысл.

1.2. Поиск

В алгоритме поиска используются два типа сущностей: набор словосочетаний, определенных пользователем в поисковом запросе как названия некоторых величин, взаимосвязь между которыми в виде математической формулы требуется найти, и математические конструкции в размеченном документе. Для работы с математическими конструкциями используются элементы математической разметки. Для работы со словосочетаниями, составляющими поисковый

запрос, используются методы поиска в сплошных текстах.

Реализация поиска может быть выполнена двумя способами. В первом варианте поиск начинается с выполнения полнотекстового запроса, затем следует фильтрация результатов и использование разметки математического текста. Во втором варианте на основе математической разметки проводится отбор документов, которые могут содержать искомые формулы, и далее выполняется фильтрация документов по полнотекстовым запросам. Выбор того или иного варианта зависит от типа коллекции, по которой производится поиск. Если поиск производится по разнородной коллекции, например, Интернету в целом, то второй вариант может оказаться предпочтительнее, так как количество документов, содержащих математические формулы, значительно меньше общего числа документов коллекции. Если поиск производится по специализированным математическим коллекциям, то более целесообразно начать с выполнения полнотекстового поиска.

Полнотекстовый поиск словосочетаний поискового запроса обладает некоторыми особенностями. Словосочетания запроса, именуемые естественнонаучные величины, являются именными группами с достаточно сложной внутренней структурой (например, многословный термин «допустимый прогиб квадратной деревянной балки при сосредоточенной нагрузке»). Обнаружив в тексте словосочетание из запроса и математическую конструкцию в непосредственной близости, можно предположить, что локализован фрагмент документа с текстовым определением символьной конструкции.

С другой стороны, культура написания технических текстов допускает, определив предмет обсуждения, именовать относящиеся к нему величины без упоминания самого объекта рассмотрения. Так, если статья называется «О квадратных деревянных балках под сосредоточенной нагрузкой» определение интересующей нас переменной может быть дано кратко, например как «допустимый прогиб» или «допустимый прогиб балки». Таким образом, полнотекстовый поиск должен отмечать как релевантные все вхождения не только исходного определения, но и всех возможных усечений именной группы, вплоть до единственного главного слова именной группы. С другой стороны, вхождение большого количества и, даже всех, слов исходного словосочетания в текст, но не в соответствующей одному из усечений именной группы форме, не должен рассматриваться как релевантный.

Определив позиции релевантных вхождений словосочетаний запроса в анализируемый текст, далее исследуется гипотеза о наличии определения символьного обозначения некоторой величины. Для этого необходимо проанализировать, с каким из вхождений фрагментов математической разметки,

соответствующих отдельной переменной, соотносится данный текст. Это довольно сложный вопрос, требующий отдельного рассмотрения. Стилистика естественнонаучных текстов допускает достаточно произвольные формы определения символьных обозначений, наряду с общепотребительными конструкциями типа "... , где x - допустимый прогиб...", или "... пусть x - допустимый прогиб...", определения могут иметь более сложную структуру, например, иметь табличную структуру, в которой кроме обозначений содержится дополнительная информация. Несколько подряд идущих определений ставят вопрос об ассоциативности: с какой ближайшей символьной конструкцией должно быть ассоциировано вхождение словосочетания запроса. В настоящее время выявлено несколько эвристик, однако этот вопрос нуждается в дальнейших исследованиях.

После установления связи между словосочетаниями запроса и символьными конструкциями производится проверка, входят ли данные символьные конструкции в одну математическую формулу. Эта задача решается на основе построенной разметки формулы, в которой с каждым вхождением переменной в тексте ассоциируются все ее вхождения в математические формулы, имеющиеся в тексте. Если в анализируемом тексте имеются одна или несколько математических формул, в которой использованы все символьные определения, соответствующие словосочетаниям запроса, то найденный текст полностью релевантен запросу.

Случаи частичной релевантности требуют дополнительного рассмотрения. Возможна частичная релевантность нескольких типов: установлено наличие формул, связывающих не

полные словосочетания запроса, а некоторые их усечения; присутствует формула, связывающая часть словосочетаний запроса. Каждый из этих случаев требует своего метода сортировки по релевантности и метода их взаимного ранжирования. Решение этих вопросов требует проведения вычислительных экспериментов на реальных коллекциях.

2. Прототип системы семантического поиска математических выражений

Для проверки базовых концепций поиска математических формул была реализован прототип системы семантического поиска математических выражений. разметку, весьма близкую к TeX - разметке. параметров, а также ссылка на страницу Википедии в Интернете. Прототип системы специализирован для поиска математических формул в статьях русской Википедии. Выбор Википедии для экспериментов связан, с одной стороны, с тем, что Википедия является одной из крупнейших коллекций научных текстов из различных областей знаний, с другой стороны, математические выражения в Википедии имеют унифицированную форму представления. В текущей реализации запрос поддерживает ввод до пяти наименований текстовых параметров формулы (рис. 2). Результатом поиска является список страниц Википедии, на которых отображается формула, фрагменты содержимого страницы с определением заданных параметров, а также ссылка на страницу Википедии в Интернете.

2.1. Архитектура системы

На рис. 3 представлена архитектура разработанного прототипа системы семантического поиска математических выражений.

Поиск формул в Википедии

Лаборатория технологий баз данных, НИИММ, КФУ, Казань, 2010.

Введите наименования параметров:

Описание системы

Система ищет математические формулы в [русской Википедии](#) по ключевым словам, с использованием в качестве таковых общепринятых наименований параметров, зависимость между которыми мы хотим найти.

Например, мы хотим найти формулу, связывающую два параметра: средняя скорость и пройденный путь. Для этого мы вводим в поля поиска термины «средняя скорость» и «путь».

Примеры использования. (для заполнения полей просто нажмите на соответствующую ссылку)

- [средняя скорость, путь](#)
- [емкость, заряд](#)
- [сила тока, сопротивление](#)

Рисунок 2 – Web-страница ввода параметров семантического поиска формул

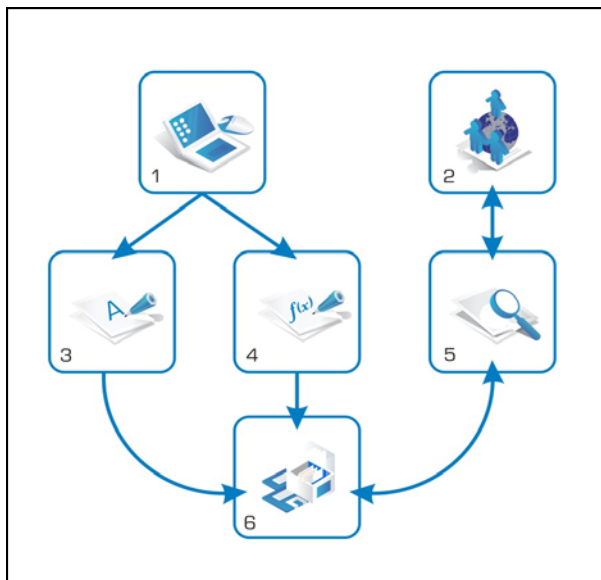


Рисунок 3 – Архитектура прототипа системы семантического поиска математических выражений

Прототип включает следующие взаимосвязанные подсистемы (рис. 3):

- подсистема загрузки и анализа данных Википедии;
- подсистема взаимодействия с пользователем;
- подсистема полнотекстового индексирования;
- подсистема индексирования математических формул и переменных;
- подсистема поиска и ранжирования;
- подсистема хранения данных.

2.1.1. Подсистема загрузки и анализа данных

Технически возможно несколько вариантов загрузки исходной информации (постраничное копирование статей Википедии в реальном масштабе времени или формирование архива статей Википедии в xml –формате). В настоящий момент реализован второй вариант как наиболее предпочтительный на этапе разработки, т.е. первоначально производится импорт данных русской Википедии из предварительно полученного архива. Во время импорта осуществляется анализ и обработка загружаемых данных. Единицей анализируемой и загружаемой информации является html-страница. Страница разбирается на структурные части, выделяется заголовок и контент. Процедура разбора проводится с помощью стандартного Java SAX-парсера. Затем контент страницы анализируется на наличие математических формул. Анализ проводится стандартными средствами Java и будет описан далее. Страницы, содержащие формулы, сохраняются в базе данных для дальнейшего использования, остальные отбрасываются. Завершается подготовка системы к работе индексированием сохраненной информации для обеспечения возможности поиска.

2.1.2. Подсистемы индексирования

Для обеспечения возможности высокоскоростного поиска ключевых фраз и математических формул необходимо осуществлять их индексирование. В системе производится индексирование входных документов как текстовых данных, а также дополнительное индексирование математических формул.

Для решения задачи поиска релевантных документов по набору ключевых слов (полнотекстового поиска) используется библиотека Apache Lucene [Lucene]. Подсистема хранения Lucene, как и в большинстве современных поисковых систем, организована в виде так называемого обратного или инвертированного индекса. Обратный индекс представляет собой список слов, документированных в алфавитном порядке с указанием позиции и других параметров вхождения слова, т.е. организован аналогично предметному указателю в конце книги – каждому слову соответствует список документов, в которых он встречается. Такая структура позволяет практически за константное время извлекать список документов, в которых встречается определенное слово. Кроме собственно номеров документов для каждого слова сохраняется ряд атрибутов, таких как все позиции, в которых встречается данное слово в данном документе, сдвиги относительно предыдущего слова, ряд других встроенных атрибутов, а также дополнительные атрибуты. Индекс Lucene фактически представляет собой документно-ориентированную базу данных с объектами и полями этих объектов. В отличие от реляционных баз объекты в ней не обязаны подчиняться какой-либо схеме. Каждый документ может иметь произвольное число произвольных полей, независимо от того, какие поля имеют другие документы.

Для индексирования математических фрагментов выполняются следующие операции: обнаружение в тексте; классификация (формула, переменная, другие); построение позиционных индексов формул и переменных; построение индексов соответствия переменных формулам.

Для записи математической формулы в Википедии используется формат записи формул LaTeX. В настоящее время для обработки математических текстов доступны ряд открытых библиотек (Jeuclid [<http://jeuclid.sourceforge.net>] - поддерживает формат MathML; TeXlipse [<http://texlipse.sourceforge.net>] - формат LaTeX; SnuggleTeX [<http://www2.ph.ed.ac.uk/snuggletex>] поддерживает преобразование LaTeX в MathML). Перечисленные инструменты по ряду причин не используются в настоящей реализации (требуется модификация исходного кода инструментов, существует зависимость результатов разбора формул от целевой функции и др.), однако в дальнейшем планируется их использование для более полного разбора формул. Прототип системы поиска использует оригинальный алгоритм

обнаружения и классификации математических фрагментов. Фрагментом формулы считается любой текст между специализированными тегами разметки $\langle \text{math} \rangle \langle / \text{math} \rangle$ (рис.4).

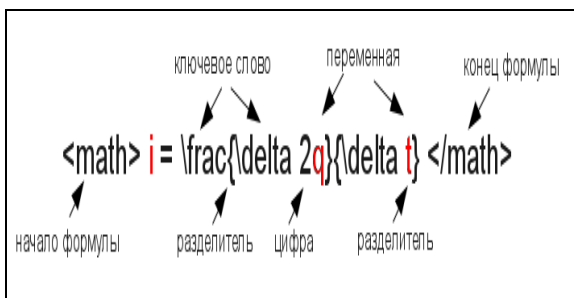


Рисунок 4 – Структура математической формулы

Полученный фрагмент очищается от служебных символов языка разметки Википедии, лишних пробельных символов. Далее фрагмент проверяется на соответствие ряду критериев (длина, количество переменных, наличие операторов отношений и операций). Если фрагмент удовлетворяет основным критериям, то он считается формулой (переменной или другим, например, таблицей). Позиции формул и переменных в тексте запоминаются в соответствующих индексах.

При построении индексов соответствия формул и переменных важным является наличие уникальных переменных в формуле, поэтому анализ формулы значительно упрощается (в отличие от полного грамматического разбора). В качестве инструмента анализа используется язык регулярных выражений. Сначала формула разбивается на фрагменты, разделителями считаются различные символы скобок, символы арифметических и

логических операций, знаки пунктуации, пробельные символы и т. п. Полученные фрагменты анализируются на принадлежность к специальным группам (ключевые слова (начинаются с символа «\»), нижние индексы (начинаются с символа «_»), числа и т. п.). Если фрагмент на этом этапе не классифицирован, то с большей долей вероятности его можно считать переменной. Выявленные ранее переменные в тексте и выявленные переменные в формулах проверяются на соответствие, затем строится индекс вхождения переменных в формулы.

2.1.3. Подсистема поиска и ранжирования

Процесс решения задачи семантического поиска математических выражений выполняется в несколько этапов. На первом этапе производится полнотекстовый поиск всех вхождений ключевых словосочетаний в тексты. Для каждого вхождения определяется, есть ли в некоторой окрестности ключевой фразы (не более 50 символов) переменная. По найденным переменным определяется соответствующая формула. Для каждой формулы строится группа текстовых фрагментов, включающих ключевые словосочетания и переменные.

На втором этапе производится поиск наилучшей группы текстовых фрагментов и соответствующих им наборов переменных для всей совокупности введенных ключевых фраз. Для этого составляются все возможные сочетания полученных текстовых фрагментов ключевых фраз в документе и проверяются по критерию близости. В качестве критерия близости использован минимум среднеквадратичного отклонения найденных

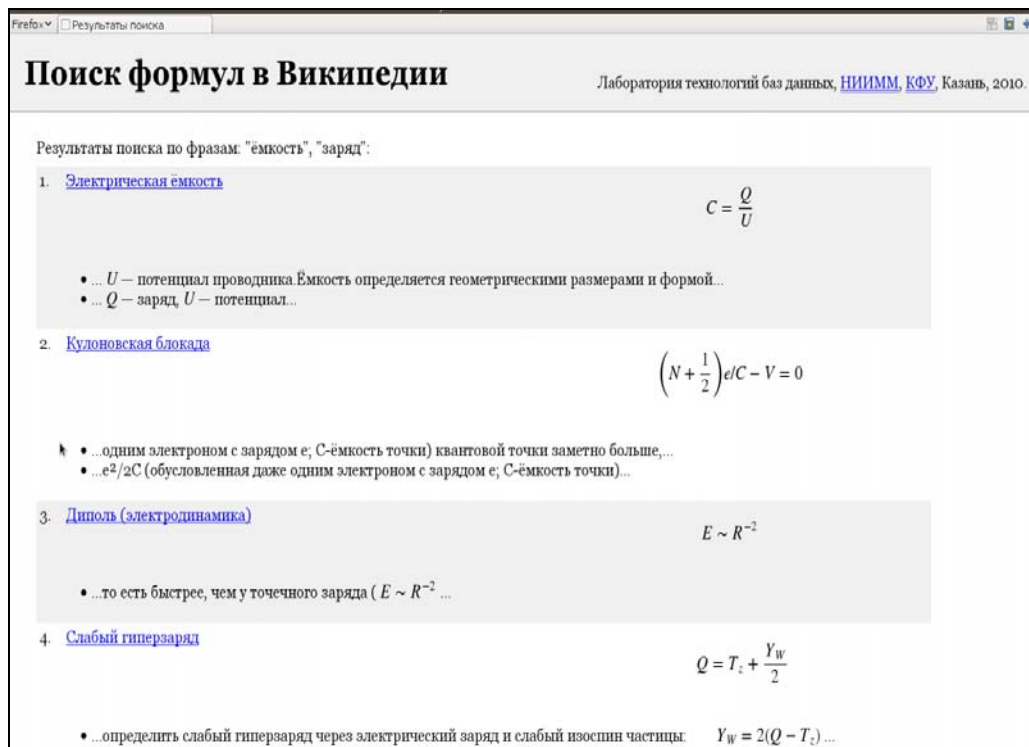


Рисунок 5 – Результаты математического поиска по параметрам “емкость”, “заряд”

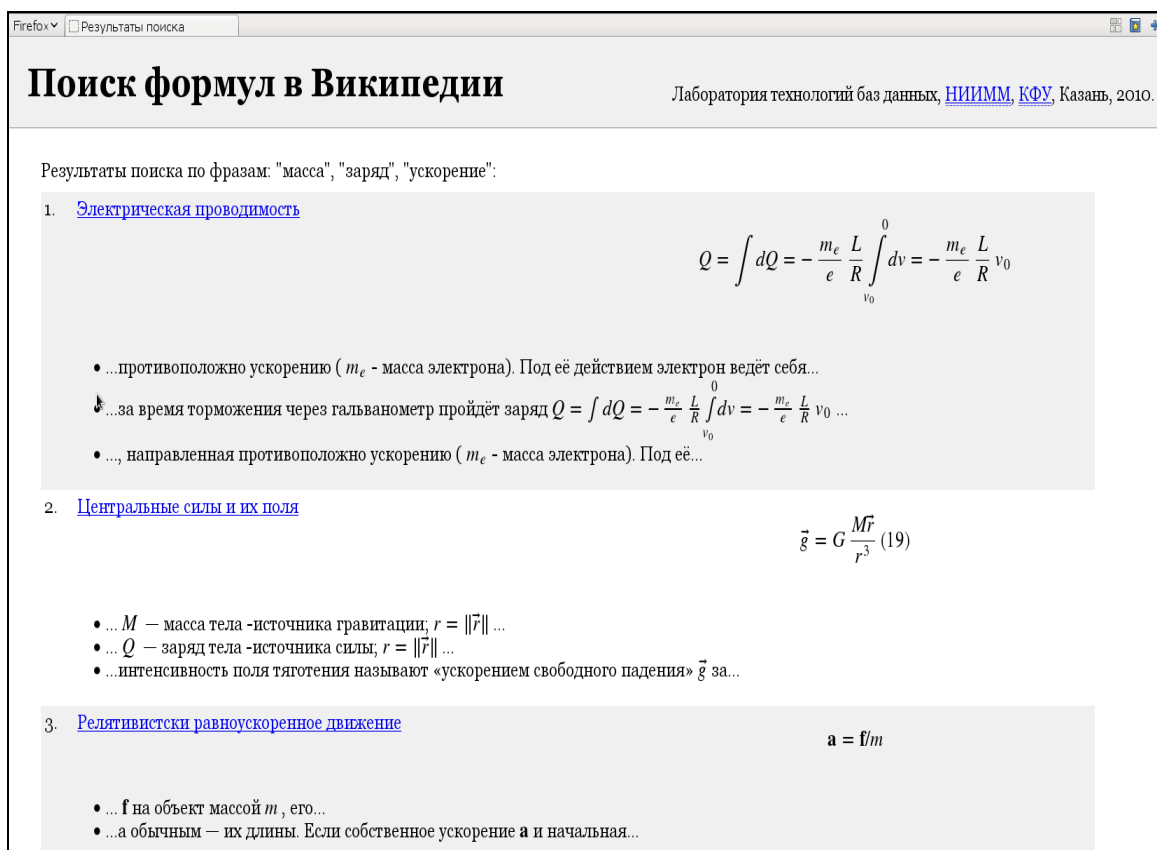


Рисунок 6 – Результаты математического поиска по параметрам “масса”, “заряд”, “ускорение”

фрагментов с определениями переменных от позиции соответствующей формулы. Предполагается, таким образом, что формула и определения, входящих в них переменных, должны быть достаточно близки друг к другу. В результате для каждого документа получаем оптимальную группу текстовых фрагментов (потенциальных определений) и относящуюся к ним формулу. Результаты для всех документов сортируются по критерию близости и дополнительным критериям релевантности.

3. Программная реализация

Пользовательский интерфейс представляет собой Web-приложение, доступ к которому осуществляется через любой современный браузер с поддержкой JavaScript. Интерфейс состоит из двух Web-страниц: страница ввода запросов пользователей и страницы представления результатов поиска.

На странице ввода запроса (рис. 2) пользователь может указать в текстовых полях до пяти названий параметров, которые должны присутствовать в искомой формуле, и после этого инициировать поиск. Дополнительно на странице можно выбрать predefined запросы («Примеры использования») и информацию программе («Подробнее о программе»).

Результаты поиска отображаются на странице результатов (рис. 5, 6), где выводится запрос и ранжированный по релевантности список ссылок на страницы Википедии.

Для каждой ссылки приводятся наиболее вероятная формула и фрагменты, содержащие параметры запроса. Так, например, на рис. 6 приведены результаты поиска для параметров «масса», «заряд», «ускорение». Результатом поиска являются формулы, связывающие физические величины запроса.

В качестве основной программной платформы реализации выбрана кроссплатформенная технология Java 1.6. Все используемые в системе сторонние продукты и библиотеки являются свободно распространяемыми открытыми решениями. Сервер приложений организован на основе встроенного Apache Tomcat 7.0 [<http://tomcat.apache.org>], а в качестве хранилища данных выбрана СУБД PostgreSQL 8.4 [<http://www.postgresql.org>], но также возможно использование других СУБД при использовании соответствующих драйверов. Объектная модель системы связывается с таблицами базы данных посредством ORM-технологии Apache OpenJPA [<http://openjpa.apache.org>].

Выполнение основных стандартных операций полнотекстового поиска по заданным ключевым

фразам производится с помощью библиотеки Apache Lucene [<http://lucene.apache.org>].

Для отображения найденных в документах формул возможно использование как библиотеки LaTeXMath [<http://forge.scilab.org/index.php/p/jlatexmath>] преобразования текстовой записи формулы в изображение png-формата, так и библиотеки MathJax [www.mathjax.org] динамического отображения формул в Web-контенте.

ЗАКЛЮЧЕНИЕ

В статье рассмотрен подход к семантическому поиску математических выражений, позволяющий выполнять запросы на поиск математических формул по текстовым наименованиям переменных, входящих в формулы.

В заключение обсудим ряд важных проблем будущих исследований. Одной из сложных задач является разработка методов разрешения многозначности, связанной с множественными вхождениями в текст определений переменных. На первый взгляд представляется, что эта проблема может существенно снизить релевантность поиска, однако можно привести некоторые возражения. Аналогичная проблема изучалась при рассмотрении задачи организации поиска в базах данных на основе запросов на естественном языке [Биряльцев и др., 2006], [Биряльцев и др., 2007]. В этой задаче рассматривалась многозначность при ассоциации логических (пользовательских) наименований полей БД с физическими полями в конкретном экземпляре БД. Однако, было показано, что логика самой БД в виде группировки полей в таблицы и миграции ключей между таблицами отсеивает нерелевантные сочетания [Биряльцев и др., 2007а]. Можно предположить, что при наличии множества определений одной и той же символической конструкции в тексте, логика предметной области устранил большинство нерелевантных сочетаний этих неоднозначно трактуемых символических обозначений. Достаточно маловероятно, что в одном и том же тексте автор так переопределил используемые им символические обозначения, что они одновременно появятся в формулах, имеющих различный смысл. Таким образом, проблему множественности определений символических обозначений, весьма сложную и требующую привлечения лингвистического и семантического анализа, возможно, не придется решать вовсе, а разрешение неоднозначности будет, в подавляющем числе случаев, выполняться автоматически на этапе поиска. Окончательная проверка данного предположения может быть проведена на основе численного эксперимента с коллекциями математических текстов.

Рассмотренный в статье прототип системы, несмотря на некоторые функциональные

упрощения, показал принципиальную работоспособность предложенного подхода. Реализованный в системе алгоритм поиска математических формул по введенным ключевым фразам (названиям параметров формулы), в процессе эксплуатации показал достаточную релевантность в сочетании с высокой скоростью поиска. Анализ результатов поиска показал, что выдаваемые результаты практически всегда имеют непосредственное отношение к задаваемому запросу, и на первой странице поиска находится формула, отвечающая запросам пользователя.

Однако можно указать и некоторые наиболее существенные проблемы реализации, которые будут исследованы в дальнейшем. В первую очередь необходимо дополнить методы ранжирования фрагментов, содержащих переменную и поисковый запрос, частичным синтаксическим анализом. Это позволит отфильтровывать конструкции, не имеющие синтаксических признаков определения, а также производить более адекватное аннотирование найденных определений.

В ближайшей перспективе планируется провести экспериментальное оценивание разработанного подхода, в том числе, оценку эффективности алгоритмов поиска и ранжирования в терминах стандартных метрик точности, полноты и nDCG [Jarvelin et al., 2000].

Также ведется работа по интеграции предложенной модели поиска по формулам с поиском по структурным элементам и терминологии [Биряльцев и др., 2010] для более точного задания и разрешения поисковых запросов математической предметной области.

БЛАГОДАРНОСТИ

Работа выполнена при финансовой поддержке РФФИ, грант № 11-07-00507а.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Биряльцев и др., 2006] Биряльцев Е. В., Гусенков А. М., Галимов М. Р. Особенности лексико-семантической структуры наименований артефактов реляционных баз данных // Тр. Казан. школы по компьютерной и когнитивной лингвистике TEL'2005. – Казань: Изд-во Казан. ун-та, 2006. – Вып. 9. – С. 4-12.
- [Биряльцев и др., 2007] Биряльцев Е.В., Гусенков А.М., Елизаров А.М. О доступе к электронным коллекциям в виде реляционных баз данных на основе онтологий // Труды 9-й Всерос. науч. конф. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – RCDL'2007, Переславль-Залесский, Россия, 15–18 октября 2007 г. – Переславль-Залесский : Изд-во «Университет города Переславля», 2007. – С. 211–216.
- [Биряльцев и др., 2007а] Биряльцев Е.В., Гусенков А.М. Интеграция реляционных баз данных на основе онтологий. // Ученые записки Казанского государственного университета. Серия Физико-математические науки. – 2007. – Том 149. – Книга 2. – Казань: Казан. ун-т, 2007. – С. 13–25.
- [Биряльцев и др., 2010] Биряльцев Е.В., Елизаров А.М., Жильцов Н.Г., Иванов В.В., Невзорова О.А., Соловьев В.Д. Модель семантического поиска в коллекциях математических документов на основе онтологий // Труды XII Всероссийской научной конференции RCDL'2010. – Казань: Казан. ун-т. – 2010. – С. 296–300.

[Google] Google Scholar. URL: <http://scholar.google.com>
[Jarvelin et al., 2000] Jarvelin, K., Kekalainen, J. IR evaluation methods for retrieving highly relevant documents // Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. – ACM. – 2000. – P. 41–48.

[Kohlhase M., 2005] Kohlhase M. sTeX: Semantic Markup in TeX/LaTeX. – 2005. – Режим

доступа: <https://svn.kwarc.info/repos/stex/trunk/sty/stex.pdf>

[Lange C., 2009] Lange C. Krestor – An Extensible XML -> RDF Extraction Framework // CEUR Workshop Proceedings. – 2009 – V. 449.

[LatexSearch] The Springer LaTeX Search. URL: <http://latexsearch.com>

[Lucene] Apache Lucene. URL: <http://lucene.apache.org>

[MathWebSearch] MathWebSearch.

URL: <http://search.mathweb.org/index.xhtml>

[Microsoft] Microsoft Academic Search.

URL: <http://academic.research.microsoft.com>

[Miller B., 2007] Miller B. LaTeXML: A LaTeX to XML converter. – 2007. – Режим доступа: <http://dlmf.nist.gov/LaTeXML>

[Misutka et al., 2008] Misutka, J., Galambos, L. Extending Full Text Search Engine for Mathematical Content // Proceedings of DML. – 2008. – P. 55–67.

[RDFa, 2008] RDFa in XHTML: Syntax and processing. Recommendation W3C. – 2008.

[Solovyev et al., 2010] Solovyev V., Zhiltsov N. Logical structure analysis of scientific publications in mathematics // Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS). – ACM. – 2011. - P. 21:1-21:9

[UniQuation] (uni)quation. URL: <http://uniquation.ru>

A NOVEL APPROACH TO SEMANTIC SEARCH OF MATHEMATICAL EXPRESSIONS IN THE SCIENTIFIC DOCUMENTS

Birialtsev E.V. *, Galimov M.R. *,
Zhiltsov N.G. *, Nevzorova O.A. **

* *Kazan (Volga Region) Federal University,
Kazan, Russia*

IgenBir@yandex.ru
glmvmt@gmail.com
nikita.zhiltsov@gmail.com

** *Research Institute of Applied Semiotics of the
Academy of Sciences of Tatarstan Republic,
Kazan, Russia*

onevzoro@gmail.com

We propose a novel approach to semantic search of mathematical expressions. It enables users to query scientific documents (e.g. scholarly papers or Wikipedia web pages) using textual representation of variables involved in the relevant formulas. We also present a prototype implementing semantic search of mathematical expressions. As a conclusion, we discuss a few current issues of the present implementation and future work.

INTRODUCTION

Currently, there are a lot of approaches to search of mathematical formulas that slightly differ from each other. Most of them require either proficiency in formalizing a search query in terms of the specific representation languages, such as LaTeX, or dealing

with cumbersome UI query editors. However, none of them attempt to elicit semantic relationships between mathematical symbols and their corresponding textual definitions. Thus, our approach aims to bring the gap between conventional keyword search and mathematical formula search by leveraging the semantics related to mathematical variables involved in a given mathematical expression.

The main task while searching mathematical formulas is parsing initial expressions in the representation formats and converting them into semantically enriched models. As a matter of fact, MathML and OpenMath are two standard languages that provide capabilities to capture the underlying semantics of mathematical expressions including the formula structure, involved variables and properties of operations between them. In practice, particular tools are used to convert from LaTeX based representation into MathML.

Given a full text index and MathML represented formulas, our first specific task is to associate a term or a group of terms from the index with an expression variable symbol. Finally, we should provide possibility for a user to query using indexed terms and retrieve relevant formulas that contain associated variables.

MAIN PART

We use Content and Presentation layers of MathML as internal representation formats for parsed formulas. Besides, our full text indexing machinery is based on the Apache Lucene library.

A set of our core algorithms includes indexing the document text contents relying on the document logical structure, parsing LaTeX expressions, detecting symbol definitions and connecting them with stored formulas. At the moment, detecting local contexts of variables performs runtime while querying. During the query resolution process, our method seeks over all the text definitions as a fixed-sized token window around occurrences of a given variable.

Our search prototype provides opportunities to query the search index built upon a Wikipedia page dump using a concise user interface and view search snippets enriched with the local context information related to a given relevant formula.

CONCLUSION

Among main issues of our approach, there are low disambiguation of particular text definition parts and insufficient robustness of ranking mechanisms to marginally useful keywords.

We are going to study efficiency of our approach on the real-world use cases and conduct experiments on evaluating its performance using several standard search quality metrics.

We also have some work in progress on incorporating the document structural analysis mechanisms into our formula search prototype.



УДК 681.3: 519.68

**ФОРМИРОВАНИЕ МНОГОАГЕНТНЫХ СИСТЕМ С ПОМОЩЬЮ
РЕСУРСНО-ЦЕЛЕВЫХ СЕТЕЙ**

Дюндюков. В.С., Тарасов В.Б.

*Московский государственный технический университет им. Н.Э. Баумана,
г. Москва, Россия*

Vsd89@yandex.ru

Tarasov@rk9.bmstu.ru

В работе вводится понятие ресурсно-целевых сетей, которое позволяет учитывать типологию агентов и их целевые установки при обмене ресурсами. Предварительно изложены основные причины и характеристики взаимодействия агентов и условия формирования многоагентных систем. Проанализированы основные виды взаимодействий между агентами, представленных с помощью двусторонних знаковых графов. Даны ключевые определения, варианты классификации и свойства ресурсов. На основе выделенных по двум критериям классов агентов введено определение ресурсно-целевых сетей, рассмотрены ситуации обмена ресурсами для агентов разных типов. Сформулированы и доказаны две теоремы о предельных состояниях ресурсно-целевой сети. Построены соответствующие иллюстративные примеры. Описана программная реализация ресурсно-целевых сетей на языке Python.

Ключевые слова: интеллект искусственный, агент, система многоагентная, взаимодействие агентов, ресурс, обмен ресурсами, влияние агента, граф двусторонний взвешенный, сеть ресурсно-целевая.

ВВЕДЕНИЕ

Настоящая работа посвящена проблемам построения ресурсно-целевых сетей и их использования при формировании многоагентных систем. В теории графов сетью называется взвешенный граф (мультиграф, гиперграф)

$$G = \langle X, C, W \rangle, \quad (1)$$

где X – множество вершин, C – множество дуг (ребер), а $W: C \rightarrow \mathbf{R}^+, \mathbf{R}^+$ – множество неотрицательных действительных чисел. Например, *транспортная сеть* [Берж, 1962; Харари, 1973] – это конечный граф без петель, каждой дуге c которого поставлено в соответствие неотрицательное целое число w , называемое пропускной способностью дуги.

Пусть C_x^- – множество дуг сети G , заходящих в x , а C_x^+ – множество дуг сети G , выходящих из x . Функция f , определенная на C и принимающая целочисленные значения $f: C \rightarrow \mathbf{N}^+$, называется потоком по транспортной сети, если:

$$1) f(c) \geq 0; 2) \sum_{C_x^-} f(c) - \sum_{C_x^+} f(c) \quad (2)$$

Классическим алгоритмом для решения задачи нахождения максимального потока в сети является алгоритм Форда – Фалкерсона [Форд и др., 1973].

В своей работе [Кузнецов, 2009] О.П.Кузнецов предложил формальный аппарат *ресурсных сетей*. Ресурсной сетью называется двусторонний

ориентированный граф, вершинам x_i которого поставлены в соответствие неотрицательные числа $q_i(t)$, изменяющиеся в дискретном времени t и называемые ресурсами, а дугам – неотрицательные числа w_{ij} , именуемые пропускными способностями (проводимостями). В отличие от транспортных сетей и классической потоковой модели Форда-Фалкерсона, в которых ресурс течет от источников к стокам и расположен в дугах, в ресурсной сети ресурсы находятся в вершинах, а обмен ресурсами зависит от проводимостей дуг [Кузнецов, 2009; Кузнецов и др., 2010].

Создание многоагентной системы (МАС) предполагает организацию взаимодействий между агентами. Под агентом понимается открытый, активный объект, обладающий целенаправленным поведением [Тарасов, 2002]. Ядро любого агента составляет четверка «цель-ресурс-восприятие-действие». Процесс взаимодействия агентов при создании МАС зависит от следующих основных параметров: 1) тип агента; 2) совместимость целей или намерений агентов; 3) отношение агентов к ресурсам и величины имеющихся у них ресурсов; 3) опыт агентов, связанный с некоторой проблемной областью; 4) обязательства агентов друг перед другом.

Таким образом, разработка МАС требует согласованного восприятия внешней среды и

формирования общих целей агентов, обмена ресурсами между агентами, реализации коллективных действий. Введение *ресурсно-целевых сетей* [Дюндюков и др., 2011] с разными типами вершин и двумя типами дуг, обозначаемыми сплошными и пунктирными линиями, позволит учитывать целевые установки агентов при их обмене ресурсами, а также отобразить асимметричность такого обмена

1. Агенты, взаимодействия и ресурсы

Главными характеристиками взаимодействия агентов являются: *направленность, взаимность, знак, сила, динамичность, адаптивность*. Будем вначале представлять область возможных взаимодействий между агентами в виде $L_3 = \{-, 0, +\}$. Тогда все варианты анализа взаимодействий между двумя агентами сведутся к следующим восьми случаям, изображенным на рисунке 1 в виде двухсторонних знаковых графов (тривиальный 9-й случай отсутствия взаимодействия здесь опущен). Выделяются три класса отношений [Тарасов, 2010]: а) взаимные (симметричные); б) слабоконтрастные (антисимметричные); в) контрастные (кососимметричные)

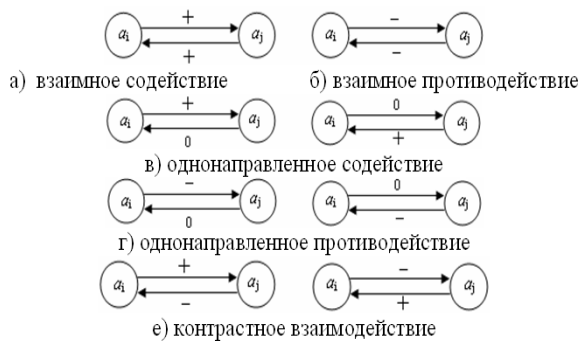


Рисунок 1 – Виды взаимодействий между агентами

В общем случае, следует говорить о степени содействия или противодействия агента другому агенту, откуда видна целесообразность применения при анализе взаимоотношений нечетких знаковых графов; примеры таких графов приведены на рисунке 2 ($w_1 > w_2$).

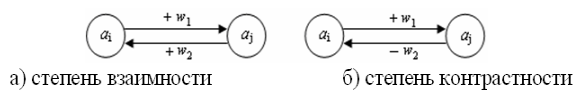


Рисунок 2 – Примеры представления силы взаимных и контрастных отношений

Любая МАС является полиструктурной и представляет собой единство экстенсивных структур, в основном развертывающихся в пространстве и интенсивных структур, развивающихся во времени [Волков, 1986; Тарасов, 2002]. При построении структур с помощью ориентированных графов развертыванию экстенсивных структур соответствует добавление новых вершин в исходный граф, а развитию интенсивных структур – добавление новых дуг.

В данной работе главное внимание уделяется исследованию и моделированию взаимосвязей между типами агентов, характером формируемых или принимаемых ими целей и ситуациями обмена (совместного использования) ресурсами.

Под ресурсами здесь понимаются любые средства, полезные для достижения цели агента или МАС. Термин «ресурс» дословно означает «восполнение или восстановление источника». Величина имеющегося у агента ресурса тесно связана с такими характеристиками МАС как роль агента в МАС и взаимосвязи между ролями. Создание и функционирование МАС предполагает построение семейства процедур распределения, перераспределения и коллективного использования ресурсов отдельных агентов.

В ситуации обмена ресурсами следует выделить два базовых типа агентов – носители ресурсов и искатели ресурсов. Носителем ресурса в момент времени t будет являться агент, который имеет определенный ресурс и может в данный момент времени удовлетворить потребности партнера в обмене, а искателем – агент, испытывающий потребности в ресурсе.

Выделяются две основные характеристики ресурса [Волков, 1986]: а) «объем ресурса» – его мера в пространстве (например, объем перерабатываемой информации, объем памяти компьютера); б) «действие ресурса» – его мера во времени. Очевидно, что один и тот же объем ресурса может оказывать различное влияние на реализацию процессов при различном его действии.

Ресурсы подразделяются на: материальные и информационные, ограниченные и неограниченные. В отличие от ограниченных материальных ресурсов, для которых в МАС действует закон сохранения суммарного ресурса, информационные ресурсы по своей природе являются бесконечными. При этом для информационных ресурсов справедливо свойство супераддитивности: ресурсы агентов в процессе обмена только увеличиваются. Поэтому суммарный ресурс МАС будет больше суммы ресурсов отдельных агентов.

Таким образом, возникают задачи расширения методов динамического структурного анализа на основе теории сетей с целью явного описания процессов обмена ресурсами в условиях приоритета индивидуальных или коллективных целей.

2. Ресурсно-целевые сети и типология агентов.

Следуя О.П.Кузнецову, для моделирования обмена ресурсами в МАС будем опираться на ресурсные сети.

Определение 1 [Кузнецов, 2009]. Ресурсной сетью для многоагентной системы называется взвешенный двусторонний ориентированный граф

$$G_{RES} = \langle A, C, RES, W \rangle, \quad (3)$$

где множество вершин A есть множество агентов, образующих МАС, множество дуг C есть множество

связей между агентами, RES – множество ресурсов МАС, причем каждый агент $a_i \in A$ имеет определенный ресурс $res(a_i) \in RES$, W – множество проводимостей w_{ij} дуг $c_{ij} \in C$ в МАС. Каждой дуге (связи между агентами) $c_{ij} \in C$ приписывается неотрицательное число $w_{ij} \in W$, называемое проводимостью от агента a_i к агенту a_j .

По сути, проводимость в МАС задает предельный объем ресурса, который один агент может передавать другому в определенный промежуток времени.

Ресурсная сеть называется однородной, если все проводимости в ней равны. В общем случае, когда МАС состоит из агентов разных типов, ресурсные сети являются неоднородными, поскольку в них как объем и действие ресурса, так и проводимости зависят от типа агента. Более того, типология агентов непосредственно определяет саму возможность формирования МАС.

Рассмотрим подробнее классификацию агентов, предложенную в [Тарасов, 2002]. Она построена по двум критериям: отношение агента к себе и отношение агента к другим агентам, в простейшем случае оценки по этим критериям сводятся к двум полярным значениям + или -. Здесь критерии классификации слегка видоизменены: отношение агента к себе понимается как его способность (+) или неспособность (-) сформировать собственную цель, а отношение к другим – как желание обмениваться ресурсами с пользой (+) или вредом (-) для себя и других агентов. Основные типы агентов показаны в таблице 1.

Таблица 1 – Основные типы агентов

Тип агента	Отношение к себе	Отношение к другим
Благонамеренный	+	+
Эгоистичный	+	-
Альтруистичный	-	+
Злонамеренный (Камикадзе)	-1	-1

Отсюда видно, что тип агента можно представить парой знаков:

- 1) Благонамеренный – (+, +);
- 2) Эгоистичный – (+, -);
- 3) Альтруистичный – (-, +);
- 4) Камикадзе – (-, -).

Ниже ограничимся анализом благонамеренных, альтруистичных и эгоистичных агентов и дадим их краткое описание с позиций ресурсно-целевого подхода [Тарасов, 2002].

Благонамеренный агент a_b – это агент, имеющий свои цели (интересы) и способный формировать или принимать коллективные цели. Он участвует в обмене ресурсами, если такая операция выгодна ему и другим агентам и не содержит злого умысла.

Эгоистичный агент a_e стремится к достижению исключительно своих целей, игнорирует цели

других агентов и неспособен к формированию общих (коллективных) целей. Он участвует в обмене ресурсами тогда и только тогда, когда этот обмен ему необходим и выгоден; при этом наблюдается скорее не обмен, а перетекание ресурсов к эгоистичному агенту.

Альтруистичный агент a_a априори считается неспособным к построению собственных целей и принимает чужую цель как общую. Такой агент всегда участвует в обмене ресурсами, даже если обмен будет неравнозначным и он от него проиграет.

Ниже вводится вариант модификации формализма ресурсных сетей и определяются ресурсно-целевые сети, где вершины задаются двумя параметрами (типом агента и объемом ресурса), а дуги – двумя видами проводимости (по целям и по ресурсам). Тип агента полностью характеризует его цели, относящиеся к ресурсам, а входные проводимости могут отличаться от выходных.

Определение 2 [Дюндюков, 2010, Дюндюков и др., 2011]. Ресурсно-целевой сетью называется взвешенный ориентированный мультиграф (гиперграф)

$$G_{RO} = \langle A, C, K, RES, W, t \rangle, \quad (4)$$

где множество вершин есть множество агентов A , а множество дуг C разбито на два непересекающихся подмножества: множество целевых связей C_G и множество ресурсных связей C_{RES} : $C = C_G \cup C_{RES}$, $C_G \cap C_{RES} = \emptyset$, t – множество дискретных моментов времени, $t = 0, 1, 2, \dots, n$. Будем обозначать ресурсные связи сплошными стрелками, а целевые связи – пунктирными.

Каждая вершина ресурсно-целевой сети (РЦС) $a_i \in A$ определяется следующими параметрами – тип агента $k_i \in K$ и объем ресурса $res(a_i) \in RES$, а каждая дуга $c_{ij} \in C$ взвешивается с помощью значения проводимости или пропускной способности $w_{ij} \in W$. У любых двух агентов a_i, a_j выделяются проводимости по целям $w_O(a_i, a_j)$ и проводимости по ресурсам $w_{RES}(a_i, a_j)$. В общем случае обмену целями и ресурсами соответствуют направленные навстречу друг другу стрелки орграфа, причем весовые коэффициенты дуг орграфа (значения входных и выходных проводимостей) могут быть различными.

В каждый дискретный момент времени t состояние сети будет определяться вектором состояния $RES(t) = (res(a_1), \dots, res(a_m))$, где m – число агентов в РЦС. Если вектор состояния ресурсно-целевой сети не изменяется со временем, то такое состояние называется устойчивым. Состояние РЦС является асимптотически достижимым, если для заранее заданного малого ζ , $\zeta > 0$, $\forall i = 1, \dots, m$, выполняется неравенство:

$$|res(a_i(t+1)) - res(a_i(t))| < \zeta. \quad (5)$$

При обмене ограниченными материальными ресурсами РЦС полагается замкнутой системой, для которой справедливо равенство:

$$\sum_{i=1}^m res(a_i) = RES_{SUM}, \quad (6)$$

$$RES_{SUM} = const.$$

В общем случае тип агента может описываться тройкой параметров, характеризующих степени благонамеренности, эгоизма и альтруизма, т.е. в ресурсно-целевом графе $K = \{a_b, a_e, a_a\}$. Общая схема возможных взаимодействий агентов дана на рисунке 3.

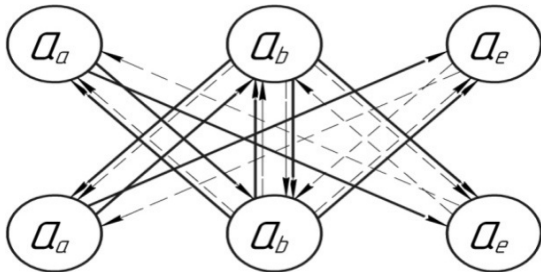


Рисунок 3 – Общая схема взаимодействий между агентами

Отсюда видно, что запрещенными являются взаимодействия между двумя эгоистичными и двумя альтруистичными агентами, поскольку при таких взаимодействиях не может быть образована многоагентная система. Поскольку эгоистичные агенты неспособны к образованию коллективных целей и равноправному обмену ресурсами, они не смогут эффективно взаимодействовать. Создание МАС только из альтруистичных агентов также невозможно из-за того, что они не имеют собственных целей и, как следствие, не смогут построить коллективную цель.

При взаимодействии агентов в МАС возможны четыре варианта взаимодействия, которые приведены на рисунке 4. Для двух агентов различных типов справедливо выражение: $w_{RES}(a_{ej}, a_{ai}) = w_{RES}(a_{ej}, a_{bi}) = \emptyset$, $w_O(a_{aj}, a_{bi}) = w_O(a_{ai}, a_{ej}) = \emptyset$.

Кратко опишем теперь возможные варианты взаимодействия агентов. В случае взаимодействия двух благонамеренных агентов происходит равноправный обмен информацией целевого характера, в результате которого формируется общая цель, а также осуществляется обмен ресурсами. Многоагентная система, состоящая из n подобных агентов, представляется наиболее эффективной для реализации стратегий децентрализованного искусственного интеллекта, когда формируется структура типа «полный граф».

При взаимодействии эгоистичного и альтруистичного агентов первый навязывает второму свою цель и использует для ее достижения чужие ресурсы. Фактически происходит перекачка ресурсов от a_a к a_e , которая может завершиться гибелью a_a , если объем ресурса $res(a_a(t)) < res_{min}$.

В случае взаимодействия благонамеренного и эгоистичного агентов возникает иллюзия обмена ресурсами между агентами. Эгоистичному агенту нужны ресурсы, но в ответ он старается ничего не отдавать. Благонамеренный агент будет избегать такого взаимодействия и участвовать в нем только в

критических для себя случаях. При этом взаимодействие прекращается, если $res(a_b(t))$ близок к res_{min} (наличие «инстинкта самосохранения» у a_b).

При взаимодействии благонамеренного и альтруистичного агентов происходит эффективный обмен ресурсами, причем a_a разделяет цели a_b . В силу своей благонамеренности a_b не допускает ситуации истощения ресурсов у a_a .

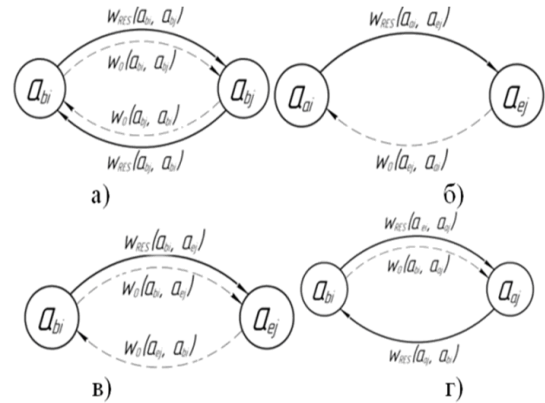


Рисунок 4 – Примеры ресурсно-целевых сетей

- а) обмен между благонамеренными агентами;
- б) обмен между эгоистичным и альтруистичным агентом;
- в) обмен между эгоистичным и благонамеренным агентами;
- г) обмен между благонамеренным и альтруистичным агентами.

Здесь сплошными линиями изображены ресурсные связи, а пунктирными – целевые связи между агентами

Как известно, степенью вершины неориентированного графа называется число ребер, инцидентных данной вершине. Для ресурсно-целевого орграфа следует учитывать как число выходящих из вершины дуг, так и их суммарную пропускную способность. Таким образом, влияние агента a в МАС определяется собственным ресурсом, числом связей с другими агентами (т.е. числом выходящих из вершины a дуг), а также суммарными значениями выходных проводимостей по целям w_O и ресурсам w_{RES} . Формально влияние агента a в момент времени t определяется имеющимся у него объемом ресурса $res(a(t))$, мощностью множества выходных целевых связей $|C_{Out}(t)|$ и/или выходных ресурсных связей $|C_{RESout}(t)|$, а также соответствующими величинами суммарной проводимости $W_{Out}(t)$ и $W_{RESout}(t)$.

Определяющим критерием влияния агента в МАС является его отношение к ресурсу. Чем большим относительным ресурсом обладает агент, тем большее влияние он оказывает на других агентов. Влияние агента в МАС заключается в способности предоставлять необходимые ресурсы, а также находить ресурсы, требуемые для достижения общих целей.

В процессе обмена объем ресурса агента изменяется. Так при уменьшении ресурса агента степень его влияния на МАС уменьшается.

В предельном случае агент может поменять свой тип, например, превратиться из благонамеренного в альтруистичного. Напротив, при значительном

увеличении своего влияния благонамеренный агент может стать эгоистичным.

Можно выделить необходимые условия существования агентов в МАС: 1) $res(a_i(t)) \geq res_{min}$ – условие индивидуального выживания агента; 2) $res(a_i(t)) \leq res_{max}$ – условие социального поведения агента.

Если объем ресурса агента $res(a_i(t)) \geq res_{max}$, то возникает ситуация бесконтрольного возрастания его влияния, что может привести к асоциальному поведению в МАС. Здесь предельным состоянием МАС будет являться ее распад, так как эгоистичный агент не сможет участвовать в обмене ресурсами из-за отсутствия участников обмена.

При обмене ресурсами благонамеренный агент отдает для обмена весь ресурс, который превышает прожиточный минимум, $res(a_i(t)) - res_{min}$. Вершины ресурсно-целевой сети, которым соответствуют благонамеренные агенты, могут рассматриваться как вершины с петлями. Для упрощения дальнейшего рассмотрения взаимодействия агентов примем:

$$res^*(a_i(t)) = res(a_i(t)) - res_{min}, RES_{SUM}^* = RES_{SUM} - \sum_{i=1}^m RES_{min i} \quad (7)$$

Теорема 1. Если ресурсно-целевая сеть состоит из благонамеренных агентов и для каждой из дуг w_0 и w_{RES} равны между собой и $RES_{SUM} \leq W_{RES}$, где W_{RES} – максимально отдаваемый агентами сети ресурс за одну-единственную итерацию, $W_{RES} = \sum_{i=1}^m \sum_{j=1}^m W_{RES}(a_i, a_j) = W_{RES} m(m-1)$, то с течением времени ресурс системы распределится между агентами равномерно и $RES(t) = (\frac{RES_{SUM}}{m}, \dots, \frac{RES_{SUM}}{m})$.

В противном случае, если хотя бы у двух агентов ресурсы не будут равны в начальный момент времени, то выравнивание ресурса в сети не произойдет.

Доказательство. 1) Если $RES_{SUM} > W_{RES} m(m-1)$, то выравнивания не произойдет, так как агенты будут получать и передавать одинаковое количество ресурса.

2) Упорядочим агентов сети таким образом, чтобы $res(a_k(0)) > W_{RES}(m-1)$ и $res(a_{k+1}(0)) \leq W_{RES}(m-1)$, тогда в результате получим:

$$\begin{aligned} res(a_1(0)) &\geq \dots \geq res(a_k(0)) \geq \\ &\geq res(a_{k+1}(0)) \geq \dots \geq res(a_m(0)), \\ res(a_1(0)) &= W_{RES}(m-1) + car(a_1(0)), \dots \\ \dots, res(a_k(0)) &= W_{RES}(m-1) + car(a_k(0)), \quad (8) \\ res(a_{k+1}(0)) &= W_{RES}(m-1) - sel(a_{k+1}(0)), \dots \\ \dots, res(a_m(0)) &= W_{RES}(m-1) - sel(a_m(0)), \end{aligned}$$

где $car(a_1(0)), \dots, car(a_k(0)) > 0$ – излишки ресурса, которые остаются у i -го агента после обмена ресурсами, а $sel(a_{k+1}(0)), \dots, sel(a_m(0)) > 0$ – ресурс, которого не хватает i -му агенту для удовлетворения потребности в обмене в полной мере.

Назовем множество агентов, которые имеют определенный ресурс и могут удовлетворить

потребность в обмене, множеством носителей CAR . Данное множество определяется набором первых k агентов упорядоченного множества, для которых $res(a_i(0)) > W_{RES}(m-1)$. Тогда множество искателей SEL есть множество таких агентов, испытывающих потребность в ресурсе, которые в данный момент времени не могут удовлетворить потребности других агентов в обмене ресурсами, $res(a_i(0)) \leq W_{RES}(m-1)$.

Если сложить суммарный излишек у агентов из множества носителей, то получим суммарный профицит носителей $CAR(t)$.

$$CAR(t) = \sum_{i=1}^k car(a_i(t)).$$

Величину общего ресурса агентов сети в момент времени t , которого не хватает для удовлетворения потребности в обмене, назовем дефицитом искателей $SEL(t)$.

Пример 1. $m = 5, W_{RES} = 2, W_{RES}(m-1) = 0, W_{RES} m(m-1) = 40, RES(0) = (17, 14, 7, 5, 2), RES_{SUM} = 45$. Так как $RES_{SUM} > W_{RES} m(m-1)$, то суммарный ресурс сети не распределится равномерно среди агентов. $k=2, car(a_1(0)) = 9, car(a_2(0)) = 6, sel(a_3(0)) = 1, sel(a_4(0)) = 3,$

$$sel(a_5(0)) = 6.$$

Просуммировав выражения ресурсов агентов, определенные через величины излишка и необходимого ресурса, получим:

$$\begin{aligned} \sum_{i=1}^k res(a_i(t)) &= kW_{RES}(m-1) + \sum_{i=1}^k car(a_i(t)) = \\ &= kW_{RES}(m-1) + CAR(t) \\ \sum_{i=k+1}^m res(a_i(t)) &= (m-k)W_{RES}(m-1) - \\ &- \sum_{i=k+1}^m sel(a_i(t)) = (m-k)W_{RES}(m-1) - SEL(t) \end{aligned} \quad (9)$$

Таблица 2 – Изменение ресурсов у агентов

Номер итерации	Номер агента				
	1	2	3	4	5
1	17	14	7	5	2
2	14,5	11,5	5,75	6,25	7
3	13,25	10,25	7,31	7,19	7
4	12,63	9,62	7,55	7,58	7,62
5	12,3	9,3	7,8	7,8	7,8
6	12,16	9,16	7,89	7,89	7,9
7	12,08	9,07	7,95	7,95	7,95
8	12,04	9,05	7,97	7,97	7,97
9	12,015	9,015	7,99	7,99	7,99
10	12	9	8	8	8

Просуммировав эти выражения, получим:

$$\sum_{i=1}^k res(a_i(t)) + \sum_{i=k+1}^m res(a_i(t)) = \sum_{i=1}^m res(a_i(t)) = RES_{SUM} = mW_{RES}(m-1) + CAR(t) - SEL(t) \quad (10)$$

Если k не изменяется с течением времени t , то

$SEL(t) - CAR(t) = const = \Delta$. Причем $\Delta = 0$, если $W_{RES}m(m-1) = RES_{SUM}$.

Рассмотрим обмен ресурсами между множеством носителей и множеством искателей. В момент времени $t=1$ CAR отдает SEL величину ресурса $\frac{W_{RES}k(m-k)}{W_{RES}m(m-1) - CAR(0)}$, а получает от SEL $\frac{SEL(0)}{W_{RES}m(m-1) - CAR(0)}$. Найдем зависимости профицита носителей и дефицита искателей в произвольный момент времени t от их начального значения.

$$CAR(1) = CAR(0) - \frac{W_{RES}k(m-k)}{W_{RES}m(m-1) - CAR(0)} + \frac{SEL(0)}{W_{RES}m(m-1) - CAR(0)} = CAR(0) - k \frac{SEL(0)}{W_{RES}m(m-1) - CAR(0)}$$

$$CAR(t) = CAR(0) \left(\frac{W_{RES}m(m-1) - CAR(0)}{W_{RES}m(m-1) - CAR(0) - k SEL(0)} \right)^t$$

$$SEL(1) = SEL(0) - k \frac{SEL(0)}{W_{RES}m(m-1) - CAR(0)} = SEL(0) \frac{W_{RES}m(m-1) - CAR(0) - k SEL(0)}{W_{RES}m(m-1) - CAR(0)}$$

$$SEL(t) = SEL(0) \left(\frac{W_{RES}m(m-1) - CAR(0) - k SEL(0)}{W_{RES}m(m-1) - CAR(0)} \right)^t$$

$$SEL(t) = SEL(0) \left(\frac{W_{RES}m(m-1) - CAR(0) - k SEL(0)}{W_{RES}m(m-1) - CAR(0)} \right)^t$$

Рассмотрим 3 возможных случая: $k=0$, $k=1$, $k>1$.

Случай 1. Пусть $k=0$. В данном случае множество носителей пусто и $\forall i = 1, \dots, m$, $res(a_i(0)) \leq W_{RES}(m-1)$. Все агенты сети отдадут свой ресурс для обмена. Поэтому в каждый последующий момент времени ресурс агента будет состоять из того, что он получит от других агентов:

$$res(a_1(1)) = \frac{RES_{SUM} - res(a_1(0))}{m-1}, \dots, res(a_m(1)) = \frac{RES_{SUM} - res(a_m(0))}{m-1}$$

Так как $res(a_1(0)) \geq \dots \geq res(a_m(0))$, то $res(a_m(1)) \geq \dots \geq res(a_1(1))$. Максимальная разность между ресурсами РЦС в момент времени $t=1$ будет равна:

$$res(a_m(1)) - res(a_1(1)) = \frac{res(a_m(0)) - res(a_1(0))}{m-1}$$

$$\lim_{t \rightarrow \infty} res(a_m(t)) - res(a_1(t)) = \lim_{t \rightarrow \infty} \frac{res(a_m(0)) - res(a_1(0))}{m-1}$$

Пример 2. $m = 5$, $W_{RES} = 2$, $W_{RES}(m-1) = 8$, $W_{RES}m(m-1) = 40$, $RES(0) = (7, 5, 4, 3, 2)$,

$RES_{SUM} = 21$.

Таблица 3 – Изменение ресурсов у агентов

Номер	Номер агента
-------	--------------

итерации	1	2	3	4	5
1	7	5	4	3	2
2	3,5	4	4,25	4,5	4,75
3	4,38	4,25	4,19	4,12	4,06
4	4,16	4,19	4,2	4,22	4,23
5	4,21	4,2	4,2	4,195	4,195
6	4,2	4,2	4,2	4,2	4,2

Случай 2. $k=1$. В этом случае $res(a_1(0)) = W_{RES}(m-1) + car(a_1(0))$, а для всех остальных агентов сети справедливо условие $res(a_i(0)) \leq W_{RES}(m-1)$. Причем $car(a_1(t)) = CAR(t)$.

В данном случае необходимо рассмотреть два возможных варианта: $RES_{SUM} = W_{RES}m(m-1)$ и $RES_{SUM} > W_{RES}m(m-1)$.

Случай 2.1. $RES_{SUM} = W_{RES}m(m-1)$, $\Delta = 0$.

Профицит носителей и дефицит искателей равен:

$$CAR(t) = CAR(0) \left(\frac{W_{RES}m(m-1) - CAR(0)}{W_{RES}m(m-1) - CAR(0) - SEL(0)} \right)^t$$

$$SEL(t) = SEL(0) \left(\frac{W_{RES}m(m-1) - CAR(0) - SEL(0)}{W_{RES}m(m-1) - CAR(0)} \right)^t$$

$$res(a_1(0)) = W_{RES}(m-1) + car(a_1(0))$$

$$res(a_i(0)) = W_{RES}(m-1) - sel(a_i(0)), \quad \forall i=2, \dots, m$$

$\lim_{t \rightarrow \infty} CAR(t) = \lim_{t \rightarrow \infty} SEL(t) = 0$,

$\lim_{t \rightarrow \infty} RES(t) = (W_{RES}(m-1), \dots, W_{RES}(m-1))$.

Причем $res(a_1(0))$ стремится к 0 сверху, а $res(a_i(0))$ – снизу.

Пример 3. $m = 5$, $W_{RES} = 2$, $W_{RES}(m-1) = 8$, $W_{RES}m(m-1) = 40$, $RES(0) = (15, 8, 7, 6, 5)$,

$RES_{SUM} = 41$.

Таблица 4 – Изменение ресурсов у агентов

Номер итерации	Номер агента				
	1	2	3	4	5
1	15	8	7	9	5
2	13,5	6,5	6,75	7	7,25
3	12,38	7,25	7,19	7,12	7,06
4	11,53	7,34	7,36	7,38	7,39
5	10,9	7,53	7,53	7,52	7,52
6	10,42	7,64	7,64	7,65	7,65
...
22	9,012	7,997	7,997	7,997	7,997
23	9	8	8	8	8

Случай 2.2. $RES_{SUM} > W_{RES}m(m-1)$. Число носителей k не меняется, если $CAR(t) = SEL(t) - \Delta$. В выражении $CAR(t) = CAR(0) \left(\frac{W_{RES}m(m-1) - CAR(0)}{W_{RES}m(m-1) - CAR(0) - SEL(0)} \right)^t - \Delta$ первый член с течением времени стремится к 0, а второй возрастает. Наступит такой момент времени t , для которого $CAR(t) \leq 0$ и число агентов в множестве носителей будет равно 0. Следовательно, будет выполнено условие случая 1.

Пример 4. $m = 5$, $W_{RES} = 2$,

$$w_{RES}(m-1) = 8, w_{RES}m(m-1) = 40, RES(0) = (12, 5, 4, 3, 2).$$

$$RES_{SUM} = 26.$$

Таблица 5 – Изменение ресурсов у агентов

Номер итерации	Номер агента				
	1	2	3	4	5
1	12	5	4	3	2
2	7,5	4,25	4,5	4,75	5
3	4,62	5,38	5,38	5,31	5,31
4	5,34	5,14	5,16	5,17	5,19
5	5,16	5,21	5,21	5,21	5,21
6	5,208	5,198	5,198	5,198	5,198
7	5,2	5,2	5,2	5,2	5,2

Случай 3. $k > 1$. Возможны два варианта:
 $car(a_1(0)) = car(a_2(0)) = \dots = car(a_k(0))$ и
 $car(a_1(0)) \geq car(a_2(0)) \geq \dots \geq car(a_k(0))$.

Случай 3.1. $k > 1$.

$$car(a_1(0)) = car(a_2(0)) = \dots = car(a_k(0)).$$

Обмен между агентами из множества носителей будет равновесным, агенты отдают друг другу и получают ресурс $w_{RES}(m-1)$. Обмен между множествами искателей и носителей описывается формулами:

$$CAR(t) = CAR(0) \left(\frac{w_{RES}(m-1)}{w_{RES}} \right)^t - \frac{w_{RES}}{\sum_{i=1}^k (w_{RES} - w_{RES}^i)}, \quad (23)$$

$$SEL(t) = SEL(0) \left(\frac{w_{RES}(m-1)}{w_{RES}} \right)^t. \quad (24)$$

Если $RES_{SUM} = w_{RES}m(m-1)$, $\Delta = 0$. Тогда:

$$CAR(t) = CAR(0) \left(\frac{w_{RES}(m-1)}{w_{RES}} \right)^t. \quad (25)$$

$$SEL(t) = SEL(0) \left(\frac{w_{RES}(m-1)}{w_{RES}} \right)^t. \quad (26)$$

$$\lim CAR(t) = \lim SEL(t) = 0. \quad (27)$$

$$\lim RES(t) = (w_{RES}(m-1), \dots, w_{RES}(m-1)). \quad (28)$$

Если $RES_{SUM} > w_{RES}m(m-1)$, то в выражении $CAR(t) = CAR(0) \left(\frac{w_{RES}(m-1)}{w_{RES}} \right)^t - \frac{w_{RES}}{\sum_{i=1}^k (w_{RES} - w_{RES}^i)}$ наступит такой момент времени t , для которого $CAR(t) \leq 0$. Число агентов в множестве носителей будет равно 0. Следовательно, будет выполнено условие случая 1.

Случай 3.2. $k > 1$, $car(a_1(0)) \geq car(a_2(0)) \geq \dots \geq car(a_k(0))$. Так как в процессе обмена ресурсами между агентами разность $car(a_1(0)) - car(a_k(0))$ не изменяется, а $\lim CAR(t) = 0$, то это приведет к тому, что число носителей k уменьшится и случай 3.2 сведется к случаю 3.1.

Теорема 2. Если $RES_{SUM} > w_{RES}m(m-1)$, то предельному состоянию ресурсно-целевой сети будет соответствовать вектор состояния

$$RES(t) = (p_1 \cdot car(a_1(0)) - p_1, \dots, p_k \cdot car(a_k(0)) - p_k, \dots, w_{RES}(1), \dots, w_{RES}(m-1)) \quad (29)$$

где $p_i = \frac{w_{RES} \cdot car(a_i(0))}{w_{RES} \cdot car(a_i(0)) + w_{RES} - w_{RES}^i}$, $i = k$, если $car(a_k(0)) \geq \frac{w_{RES} - w_{RES}^k}{w_{RES}}$, иначе $p_i = \frac{w_{RES} - w_{RES}^i}{w_{RES} \cdot car(a_i(0)) + w_{RES} - w_{RES}^i}$, $i \leq k$ и $car(a_i(0)) \geq \frac{w_{RES} - w_{RES}^i}{w_{RES}}$.

Доказательство. Множество носителей CAR не может увеличиваться со временем, но может

уменьшаться, если часть агентов перейдет из множества CAR в SEL . В каждый момент времени агенты из множества носителей передают агентам множества искателей одинаковый ресурс $w_{RES}(m-1)$ и получают одинаковый ресурс. Поэтому из ресурсов агентов множества носителей в каждый момент времени вычитается одинаковый ресурс. Для доказательства теоремы необходимо доказать, что предельному состоянию РЦС будет соответствовать вектор состояния $RES(t)$, вид которого определен выше. Рассмотрим 2 случая.

Случай 1. Число агентов в множестве носителей не меняется со временем, $l=k$. Рассмотрим выражение для дефицита искателей, выраженного для произвольного момента времени t . $SEL(t) = SEL(0) \left(\frac{w_{RES}(m-1)}{w_{RES}} \right)^t$. Так как k не изменяется, то $\lim_{t \rightarrow \infty} SEL(t) = 0$.

Пример

5.

$m=5$,

$$w_{RES} = 2,$$

$$w_{RES}(m-1) = 8, w_{RES}m(m-1) = 40, RES(0) = (20, 15, 7, 5, 4).$$

$$RES_{SUM} = 51, SEL(0) = 8.$$

Множество искателей стремится к выравниванию ресурса и предельное состояние существует. В предельном состоянии дефицит искателей увеличится на величину $SEL(0)$, а профицит носителей уменьшится на величину $SEL(0)$. Поскольку носители в процессе обмена отдают одинаковый ресурс, то из ресурса каждого агента будет вычтена величина $p = \frac{SEL(0)}{w_{RES}}$. Первая часть теоремы доказана.

Таблица 6 – Изменение ресурсов у агентов

Номер итерации	Номер агента				
	1	2	3	4	5
1	20	15	7	5	4
2	18	13	6,25	6,75	7
3	17	12	7,44	7,31	7,25
4	16,5	11,5	7,64	7,67	7,69
5	16,25	11,25	7,84	7,83	7,83
6	16,13	11,13	7,91	7,91	7,92
...
22	16	11	8	8	8

Случай 2. Если $car(a_k(0)) < \frac{w_{RES} - w_{RES}^k}{w_{RES}}$, то в некоторый момент времени агент a_k перейдет из множества носителей в множество искателей. Множество носителей уменьшится на одного агента, данный процесс будет продолжаться до тех пор, пока число агентов из множества носителей не станет равным 1. Новое состояние РЦС можно рассматривать как сеть с множеством искателей, состоящим из 1 агента. Для данной сети будут выполнены условия случая 1.

Пример

6.

$m =$

5,

$$w_{RES} = 2,$$

$$w_{RES}(m-1) = 8, w_{RES}m(m-1) = 40, RES(0) = (20, 10, 7, 5, 4).$$

$$RES_{SUM} = 46, SEL(t) = 8.$$

Таблица 7 – Изменение ресурсов у агентов

Номер итерации	Номер агента				
	1	2	3	4	5
1	20	10	7	5	4
2	18	8	6,25	6,75	7
3	17	7	7,44	7,31	7,25
4	16,25	7,5	7,39	7,42	7,44
5	15,69	7,56	7,59	7,58	7,58
6	15,27	7,69	7,68	7,68	7,68
...
22	14,012	7,997	7,997	7,997	7,997
23	14	8	8	8	8

3. Программная реализация.

Программа поддержки построения ресурсно-целевых сетей была реализована на языке Python. Ее интерфейс представлен на рисунке 5. В качестве вводимых параметров берутся количество агентов, величины минимального и максимального ресурса у агентов, что соответствует условиям индивидуального выживания и асоциального поведения. Также пользователю предлагается ввести типы агентов, начальное распределение ресурсов у них и величины проводимостей по ресурсам и целям.

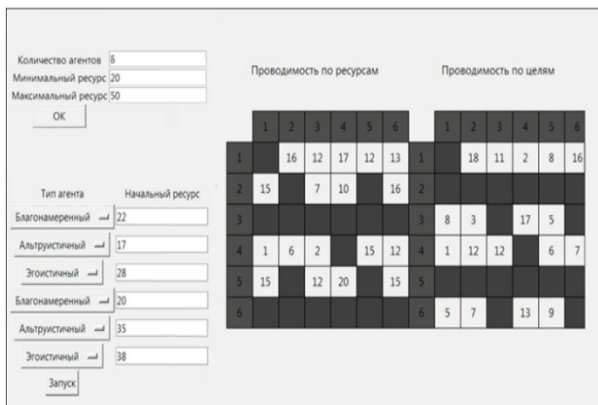


Рисунок 5 – Интерфейс программы поддержки построения ресурсно-целевых сетей на языке Python

На рисунке 6 приведен алгоритм реализованной программы. Процесс обмена ресурсами в данной программе зависит от типа агента, имеющегося у него ресурса и значений ресурсной и целевой проводимостей. Благонамеренный агент отдает для обмена ресурс $Res(a_i(t)) - Res_{min}$. Альтруистичный агент для обмена жертвует весь имеющийся ресурс, даже если на следующей итерации его ресурс окажется меньше прожиточного минимума, что приведет к его гибели. Эгоистичный агент не отдает ресурс для обмена. Величина отдаваемого агентом a_i агенту a_j ресурса $Res(a_i, a_j)$ распределяется пропорционально проводимостям по целям:

$$Res(a_i, a_j) = \frac{w_o(a_i, a_j)}{\sum_{j=1}^n (w_o(a_i, a_j))} Res(a_i(t))$$

В случае равенства проводимостей по целям между агентами, ресурс распределяется пропорционально проводимостям по ресурсам.

Данная программа позволяет рассматривать различные типовые структуры ресурсно-целевых графов (рисунок 7), такие как «кольцо», «центр», «колесо», «полный граф», а также произвольные структуры. При структуре типа «кольцо» обмен ресурсами между агентами происходит по цепочке. Каждый агент имеет возможность обмениваться только с двумя соединенными с ним агентами. Перераспределение ресурсов при данной организационной структуре замедляется. Система распадается на отдельные составляющие, если в ней появляются эгоистичные агенты.

При организационной структуре типа «центр» головным агентом для благоприятного развития сети должен стать благонамеренный агент, рационально перераспределяющий ресурсы.

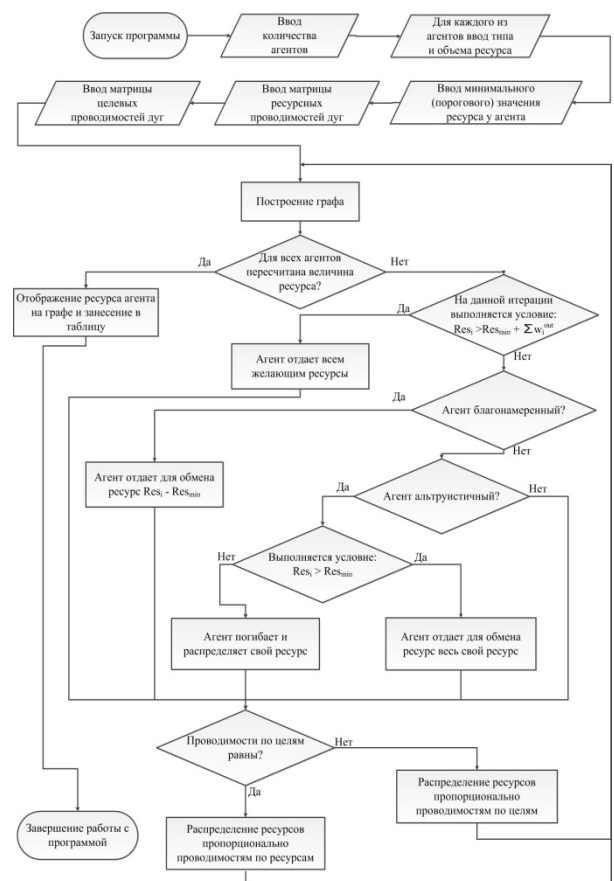


Рисунок 6 – Алгоритм программы поддержки построения ресурсно-целевых сетей

В случае организационной структуры типа «колесо» выбор в качестве центра эгоистичного агента приведет к ее быстрому превращению в структуру «звезда» и захвату ресурсов всех узлов центральным узлом.

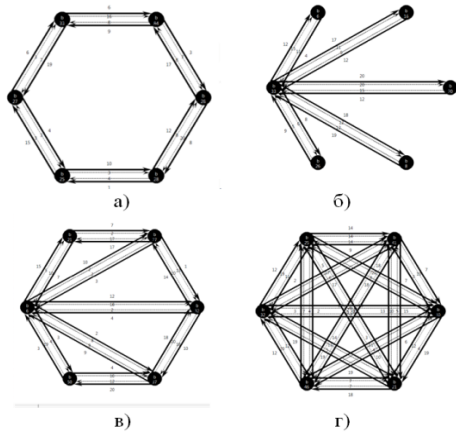


Рисунок 7 – Примеры структур ресурсно-целевых сетей: а) «кольцо»; б) «центр»; в) «колесо»; г) «полная сеть»

Наиболее благоприятными организационными структурами являются структура «полная сеть» и близкие к ней структуры, когда агент, стремясь максимизировать свои связи, имеет право выбора того, с кем обмениваться ресурсами в зависимости от своих целей. Данная структура «горизонтальной организации» лучше всего работает при комбинации благонамеренных и альтруистичных агентов.

ЗАКЛЮЧЕНИЕ

В статье предложены варианты наглядного представления процессов взаимодействия агентов, формирования и развития многоагентных систем с помощью аппарата ресурсно-целевых сетей. Показано, что этот аппарат может стать удобным средством семантико-прагматического анализа и онтологического моделирования различных систем группового (коллективного, децентрализованного, централизованного) искусственного интеллекта.

Работа выполнена при финансовой поддержке РФФИ, проекты №10-01-00844 и №11-07-13165-офи-м-2011-РЖД

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Берж, 1962] Берж, К. Теория графов и ее применения: Перевод с франц. – М.: Изд-во иностранной литературы, 1962.
- [Волков, 1986] Волков, А.М. Основы структурно-функционального анализа операторской деятельности. – М.: Изд-во МАИ, 1986.
- [Дюндюков, 2010] Дюндюков, В.С. Моделирование взаимодействия интеллектуальных агентов: применение ресурсных графов/ В.С.Дюндюков// Труды международного конгресса по интеллектуальным системам и информационным технологиям (AIS-IT'2010, Дивноморское, 2-9 сентября 2010 г.). – М.: Физматлит, 2010. – Т.1. – С.204-210.
- [Дюндюков и др., 2011] Дюндюков, В.С. Ресурсно-целевые сети: использование в многоагентных системах/ В.С.Дюндюков, В.Б. Тарасов// Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов VI-й Международной научно-практической конференции (Коломна, 16-19 мая 2011 г.). – М.: Физматлит, 2011. – Т.1. – С.483-495.
- [Кузнецов, 2009] Кузнецов, О.П. Однородные ресурсные сети. I. Полные графы/ О.П.Кузнецов// Автоматика и телемеханика. – 2009. – № 11. – С.136-147.
- [Кузнецов и др., 2010] Кузнецов, О.П. Двусторонние ресурсные сети – новая потоковая модель/ О.П.Кузнецов,

Л.Ю.Жилиякова// Доклады Академии наук. – 2010. – Т.433, №5. – С.609-612.

[Тарасов, 2002] Тарасов, В.Б. От многоагентных систем к интеллектуальным организациям. – М.: Эдиториал УРСС, 2002.

[Тарасов В.Б., 2010] Тарасов, В.Б. Методы и модели алгебраической логики в анализе взаимоотношений и переговоров агентов в многоагентных системах/ В.Б.Тарасов// Сборник трудов X-й международной научной конференции им. Т.А.Таран «Интеллектуальный анализ информации» (ИАИ-2010, Киев, 18-21 мая 2010 г.). – Киев: Просвіта, 2010. – С.304-316.

[Форд и др., 1973] Форд, Л.Р., Фалкерсон, Д.Р. Потоки в сетях: Пер. с англ. – М.: Мир, 1973.

[Харари Ф., 1973] Харари, Ф. Теория графов: Пер. с англ. – М.: Мир, 1973.

GENERATION OF MULTI-AGENT SYSTEMS BY USING RESOURCE-GOAL NETWORKS

Dyundyukov V.S., Tarassov V.B.

Moscow Bauman State Technical University

Vsd89@yandex.ru

Tarasov@rk9.bmstu.ru

The concept of resource-goal network is introduced for taking into account various agents types and their goals while exchanging resources. Primarily main reasons and characteristics of agents interaction together with some conditions for generating multi-agent systems are stated. Basic cases of interactions between agents are analyzed by using bilateral signed graphs. Some resource definitions, classifications and properties are discussed. The definition of resource-goal network (RGN) is introduced on the basis of earlier specified agent classes; some key situations of exchanging resource by agents of various classes are considered. Two theorems concerning RGN boundary states are formulated and proved. Some appropriate illustrative examples are constructed. Finally the software tools to support the development of RGN with using Python language are presented.

Keywords: Artificial Intelligence, Agent, Multi-Agent System, Agents Interaction, Resource, Resource Exchange, Agent Influence, Bilateral Weighted Graph, Resource-Goal Network

INTRODUCTION

This paper faces the problems of building resource-goal networks and their using in Artificial Intelligence to construct multi-agent systems. Usually in graph theory network is defined as a weighted graph (multigraph, hypergraph) $G = \langle X, C, W \rangle$, where X is a vertex set, C is an arc set, and $W: C \rightarrow \mathbf{R}^+$, \mathbf{R}^+ is the set of all non-negative real numbers. For instance, transportation set is given as a finite graph without loops, where each arc is equipped with a non-negative integer w , called arc conductivity. A classical algorithm to solve the problems of calculating flows in the networks is Ford-Fulkerson's algorithm.

Recently O.P.Kuznetsov has suggested the formalism of resource networks defined as bilateral oriented graphs, in which vertices are equipped with non-negative numbers $q_i(t)$ called resources (t is discrete time) and arcs are characterized by non-

negative numbers seen as conductivities. To differ from transportation networks and classical Ford-Fulkerson's flow model, where the flows are located in the arcs, the resources are placed in the vertices, and the resource exchange depends on arc conductivity.

The development of multi-agent systems (MAS) supposes organizing the interactions between agents. Here agent is viewed as an open active goal-directed object possessing its own behavior. The kernel of any agent is given by a quadruple «goal-resource-perception-action». The interaction between agents to create multi-agent system depends on the following basic parameters: 1) agent type; 2) compatibility of agent goals or intentions; 3) statute of agents and amount of available resources; 4) agents experience related to some problem area; 5) agents commitments.

Thus, the development of MAS requires distributed perception of external environment, joint goals generation, resource exchange and concerted actions. So the introduction of resource-goal networks with various types of vertices and two types of arcs denoted by continuous lines and dotted lines respectively allows us to take into account agents' goals while exchanging resources and reflect asymmetries in these exchanges.

MAIN PART

The paper considers interactions between agents from the viewpoint of purposeful resource sharing and exchanging. First of all agents interaction area is represented by a simple bi-polar scale $L_3 = \{-, 0, +\}$. Generally, three classes of relations between agents are revealed: a) mutual (symmetrical) relations; b) weakly contrasting (anti-symmetrical) relations; c) contrasting (oblique symmetrical) relations.

A crucial idea of RGN is based on the agents classification by two criteria: agent relation to himself and agent relation to other agents. Here these criteria are re-interpreted: first one is seen as agent's possibility (+) or impossibility (-) of generating his individual goal and second one as agent's will to exchanging resources or agent's possibility (+) – impossibility (-) to form collective goal. So four types of agents appear: 1) benevolent agent given by (+,+); 2) self-interested (egoist) agent denoted by (+, -); 3) altruistic agent (-, +); 4) destroyer (kamikaze) agent (-, -). In other words, benevolent agent has his own goal and is capable to participate in forming collective goal and mutual resource exchanges necessary for multi-agent system effective work. Self-interested agent is focused in his proper goal and neglects goals and interests of other agents; hence, in interaction processes he tends to impose his goal and appropriate all resources. It is obvious that multi-agent system cannot exist in case of many self-interested agents. Altruistic agent is ready to give his resources even in dangerous situations; on the contrary, he is not able to generate his own goal and easily accepts the goal of other agent. Like the selfish team including only self-interested agents, the society of altruistic agents cannot develop any collective goal. The properties of kamikaze agent are omitted in this paper.

This classification of agents underlies the definition of RGN below. By resource-goal network we mean a weighted directed multigraph (hypergraph)

$$G_{RO} = \langle A, C, K, RES, W, t \rangle,$$

where the set of vertices A is interpreted as the set of agents, the set of arcs C consists of goal connections C_0 and resource connections C_{RES} , i.e. it is a partition $C = C_0 \cup C_{RES}$, $C_0 \cap C_{RES} = \emptyset$, t is the discrete time, $t = 0, 1, 2, \dots, n$. Each agent $a_i \in A$ being RGN-vertex is given by the following parameters – agent type $k_i \in K$ and the amount of available resources $res(a_i) \in RES$, whereas each arc $c_{ij} \in C$ is weighted by the value of conductivity $w_{ij} \in W$. Here two types of conductivity are specified: conductivity by goal $w_0(a_i, a_j)$ and conductivity by resources $w_{RES}(a_i, a_j)$. The resource exchange and goal exchange are modeled by directed graph arrows, where arc weights (input or output conductivity values) may be different.

The state of the resource-goal network for each discrete time t is given by a state vector $RES(t) = (res(a_1), \dots, res(a_m))$, where m is the number of agents in RGN. If this state vector does not change, then the network state is called stable.

In course of resource exchange process RGN is supposed to be a closed system, where the conservation law for material resources holds. Contrarily, in case of information resources synergetic (superadditivity) phenomena arise.

Generally agent type may be viewed as a triple $K = \{a_b, a_e, a_a\}$, where a_b, a_e, a_a stand for degrees of benevolence, self-interest and altruism respectively.

Let us present basic cases of interactions between agents. The interaction between benevolent agents supposes individual goal information exchange to generate collective goal. Here mutually profitable resource exchange is performed. Multi-agent system constructed on the basis of such interacting agents seems to be very effective from the viewpoint of decentralized artificial intelligence strategies; the corresponding structure is a complete network.

In course of interaction between self-interested and altruistic agents the first one imposes his goal and tends to capture all resources. In fact, we observe resource transfer from a_a to a_e ; it can bring about the death of a_a , if $res(a_a(t)) < res_{min}$.

The interaction scenario for benevolent and self-interested agents is similar. Self-interested agent tends to appropriate all resources, but in response he mainly disseminates his beliefs and goal. So benevolent agent is not interested in such interactions and accepts them only in critical situation. In any case, he is provided with the mechanism of self-preservation: the interaction is interrupted in case of $res(a_b(t)) \rightarrow res_{min}$.

Finally, the interaction between benevolent and altruistic agents, when a_a shares the goal of a_b , supposes an efficient resource exchange. Here a_b does not accept the resource extinction for a_a due to his benevolence.

Another important agent characteristic in RGN is his influence degree. Here we need taking into account both vertex degree and total conductivity. Hence, agent influence in MAS is given by agent's own resource, the number of connections with other agents and sum

values of resource conductivities and goal conductivities.

As a result, two theorems about resource distribution in MAS have been proved.

The software program to support the construction of resource-goal network has been developed with using Python language.

CONCLUSION

Some ways of visual representation to model interaction processes between agents, as well as multi-agent systems generation and development have been proposed in the paper with using resource-goal networks. It has been shown that this formalism may become a quite suitable tool for semantic-pragmatic analysis and ontological modeling while considering various systems of collective (decentralized or centralized) artificial intelligence.



УДК 681.3: 519.68

**ФОРМИРОВАНИЕ МНОГОАГЕНТНЫХ СИСТЕМ С ПОМОЩЬЮ
РЕСУРЧНО-ЦЕЛЕВЫХ СЕТЕЙ**

Дюндюков. В.С., Тарасов В.Б.

*Московский государственный технический университет им. Н.Э. Баумана,
г. Москва, Россия*

Vsd89@yandex.ru

Tarasov@rk9.bmstu.ru

В работе вводится понятие ресурсно-целевых сетей, которое позволяет учитывать типологию агентов и их целевые установки при обмене ресурсами. Предварительно изложены основные причины и характеристики взаимодействия агентов и условия формирования многоагентных систем. Проанализированы основные виды взаимодействий между агентами, представленных с помощью двусторонних знаковых графов. Даны ключевые определения, варианты классификации и свойства ресурсов. На основе выделенных по двум критериям классов агентов введено определение ресурсно-целевых сетей, рассмотрены ситуации обмена ресурсами для агентов разных типов. Сформулированы и доказаны две теоремы о предельных состояниях ресурсно-целевой сети. Построены соответствующие иллюстративные примеры. Описана программная реализация ресурсно-целевых сетей на языке Python.

Ключевые слова: интеллект искусственный, агент, система многоагентная, взаимодействие агентов, ресурс, обмен ресурсами, влияние агента, граф двусторонний взвешенный, сеть ресурсно-целевая.

ВВЕДЕНИЕ

Настоящая работа посвящена проблемам построения ресурсно-целевых сетей и их использования при формировании многоагентных систем. В теории графов сетью называется взвешенный граф (мультиграф, гиперграф)

$$G = \langle X, C, W \rangle, \quad (1)$$

где X – множество вершин, C – множество дуг (ребер), а $W: C \rightarrow \mathbf{R}^+, \mathbf{R}^+$ – множество неотрицательных действительных чисел. Например, *транспортная сеть* [Берж, 1962; Харари, 1973] – это конечный граф без петель, каждой дуге c которого поставлено в соответствие неотрицательное целое число w , называемое пропускной способностью дуги.

Пусть C_x^- – множество дуг сети G , заходящих в x , а C_x^+ – множество дуг сети G , выходящих из x . Функция f , определенная на C и принимающая целочисленные значения $f: C \rightarrow \mathbf{N}^+$, называется потоком по транспортной сети, если:

$$1) f(c) \geq 0; 2) \sum_{C_x^-} f(c) - \sum_{C_x^+} f(c) \quad (2)$$

Классическим алгоритмом для решения задачи нахождения максимального потока в сети является алгоритм Форда – Фалкерсона [Форд и др., 1973].

В своей работе [Кузнецов, 2009] О.П.Кузнецов предложил формальный аппарат *ресурсных сетей*. Ресурсной сетью называется двусторонний

ориентированный граф, вершинам x_i которого поставлены в соответствие неотрицательные числа $q_i(t)$, изменяющиеся в дискретном времени t и называемые ресурсами, а дугам – неотрицательные числа w_{ij} , именуемые пропускными способностями (проводимостями). В отличие от транспортных сетей и классической потоковой модели Форда-Фалкерсона, в которых ресурс течет от источников к стокам и расположен в дугах, в ресурсной сети ресурсы находятся в вершинах, а обмен ресурсами зависит от проводимостей дуг [Кузнецов, 2009; Кузнецов и др., 2010].

Создание многоагентной системы (МАС) предполагает организацию взаимодействий между агентами. Под агентом понимается открытый, активный объект, обладающий целенаправленным поведением [Тарасов, 2002]. Ядро любого агента составляет четверка «цель-ресурс-восприятие-действие». Процесс взаимодействия агентов при создании МАС зависит от следующих основных параметров: 1) тип агента; 2) совместимость целей или намерений агентов; 3) отношение агентов к ресурсам и величины имеющихся у них ресурсов; 3) опыт агентов, связанный с некоторой проблемной областью; 4) обязательства агентов друг перед другом.

Таким образом, разработка МАС требует согласованного восприятия внешней среды и

формирования общих целей агентов, обмена ресурсами между агентами, реализации коллективных действий. Введение *ресурсно-целевых сетей* [Дюндюков и др., 2011] с разными типами вершин и двумя типами дуг, обозначаемыми сплошными и пунктирными линиями, позволит учитывать целевые установки агентов при их обмене ресурсами, а также отобразить асимметричность такого обмена

1. Агенты, взаимодействия и ресурсы

Главными характеристиками взаимодействия агентов являются: *направленность, взаимность, знак, сила, динамичность, адаптивность*. Будем вначале представлять область возможных взаимодействий между агентами в виде $L_3 = \{-, 0, +\}$. Тогда все варианты анализа взаимодействий между двумя агентами сведутся к следующим восьми случаям, изображенным на рисунке 1 в виде двухсторонних знаковых графов (тривиальный 9-й случай отсутствия взаимодействия здесь опущен). Выделяются три класса отношений [Тарасов, 2010]:

- взаимные (симметричные);
- слабоконтрастные (антисимметричные);
- контрастные (кососимметричные)

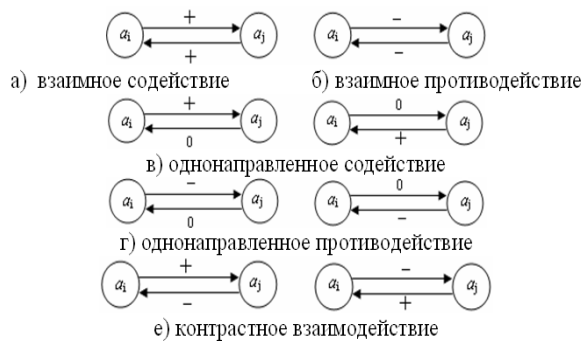


Рисунок 1 – Виды взаимодействий между агентами

В общем случае, следует говорить о степени содействия или противодействия агента другому агенту, откуда видна целесообразность применения при анализе взаимоотношений нечетких знаковых графов; примеры таких графов приведены на рисунке 2 ($w_1 > w_2$).

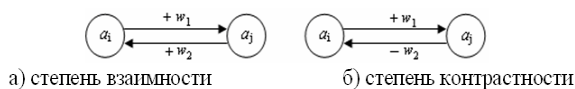


Рисунок 2 – Примеры представления силы взаимных и контрастных отношений

Любая МАС является полиструктурной и представляет собой единство экстенсивных структур, в основном развертывающихся в пространстве и интенсивных структур, развивающихся во времени [Волков, 1986; Тарасов, 2002]. При построении структур с помощью ориентированных графов развертыванию экстенсивных структур соответствует добавление новых вершин в исходный граф, а развитию интенсивных структур – добавление новых дуг.

В данной работе главное внимание уделяется исследованию и моделированию взаимосвязей между типами агентов, характером формируемых или принимаемых ими целей и ситуациями обмена (совместного использования) ресурсами.

Под ресурсами здесь понимаются любые средства, полезные для достижения цели агента или МАС. Термин «ресурс» дословно означает «восполнение или восстановление источника». Величина имеющегося у агента ресурса тесно связана с такими характеристиками МАС как роль агента в МАС и взаимосвязи между ролями. Создание и функционирование МАС предполагает построение семейства процедур распределения, перераспределения и коллективного использования ресурсов отдельных агентов.

В ситуации обмена ресурсами следует выделить два базовых типа агентов – носители ресурсов и искатели ресурсов. Носителем ресурса в момент времени t будет являться агент, который имеет определенный ресурс и может в данный момент времени удовлетворить потребности партнера в обмене, а искателем – агент, испытывающий потребности в ресурсе.

Выделяются две основные характеристики ресурса [Волков, 1986]: а) «объем ресурса» – его мера в пространстве (например, объем перерабатываемой информации, объем памяти компьютера); б) «действие ресурса» – его мера во времени. Очевидно, что один и тот же объем ресурса может оказывать различное влияние на реализацию процессов при различном его действии.

Ресурсы подразделяются на: материальные и информационные, ограниченные и неограниченные. В отличие от ограниченных материальных ресурсов, для которых в МАС действует закон сохранения суммарного ресурса, информационные ресурсы по своей природе являются бесконечными. При этом для информационных ресурсов справедливо свойство супераддитивности: ресурсы агентов в процессе обмена только увеличиваются. Поэтому суммарный ресурс МАС будет больше суммы ресурсов отдельных агентов.

Таким образом, возникают задачи расширения методов динамического структурного анализа на основе теории сетей с целью явного описания процессов обмена ресурсами в условиях приоритета индивидуальных или коллективных целей.

2. Ресурсно-целевые сети и типология агентов.

Следуя О.П.Кузнецову, для моделирования обмена ресурсами в МАС будем опираться на ресурсные сети.

Определение 1 [Кузнецов, 2009]. Ресурсной сетью для многоагентной системы называется взвешенный двусторонний ориентированный граф

$$G_{RES} = \langle A, C, RES, W \rangle, \quad (3)$$

где множество вершин A есть множество агентов, образующих МАС, множество дуг C есть множество

связей между агентами, RES – множество ресурсов МАС, причем каждый агент $a_i \in A$ имеет определенный ресурс $res(a_i) \in RES$, W – множество проводимостей w_{ij} дуг $c_{ij} \in C$ в МАС. Каждой дуге (связи между агентами) $c_{ij} \in C$ приписывается неотрицательное число $w_{ij} \in W$, называемое проводимостью от агента a_i к агенту a_j .

По сути, проводимость в МАС задает предельный объем ресурса, который один агент может передавать другому в определенный промежуток времени.

Ресурсная сеть называется однородной, если все проводимости в ней равны. В общем случае, когда МАС состоит из агентов разных типов, ресурсные сети являются неоднородными, поскольку в них как объем и действие ресурса, так и проводимости зависят от типа агента. Более того, типология агентов непосредственно определяет саму возможность формирования МАС.

Рассмотрим подробнее классификацию агентов, предложенную в [Тарасов, 2002]. Она построена по двум критериям: отношение агента к себе и отношение агента к другим агентам, в простейшем случае оценки по этим критериям сводятся к двум полярным значениям $+$ или $-$. Здесь критерии классификации слегка видоизменены: отношение агента к себе понимается как его способность ($+$) или неспособность ($-$) сформировать собственную цель, а отношение к другим – как желание обмениваться ресурсами с пользой ($+$) или вредом ($-$) для себя и других агентов. Основные типы агентов показаны в таблице 1.

Таблица 1 – Основные типы агентов

Тип агента	Отношение к себе	Отношение к другим
Благонамеренный	$+$	$+$
Эгоистичный	$+$	$-$
Альтруистичный	$-$	$+$
Злонамеренный (Камикадзе)	-1	-1

Отсюда видно, что тип агента можно представить парой знаков:

- 1) Благонамеренный – $(+, +)$;
- 2) Эгоистичный – $(+, -)$;
- 3) Альтруистичный – $(-, +)$;
- 4) Камикадзе – $(-, -)$.

Ниже ограничимся анализом благонамеренных, альтруистичных и эгоистичных агентов и дадим их краткое описание с позиций ресурсно-целевого подхода [Тарасов, 2002].

Благонамеренный агент a_b – это агент, имеющий свои цели (интересы) и способный формировать или принимать коллективные цели. Он участвует в обмене ресурсами, если такая операция выгодна ему и другим агентам и не содержит злого умысла.

Эгоистичный агент a_e стремится к достижению исключительно своих целей, игнорирует цели

других агентов и неспособен к формированию общих (коллективных) целей. Он участвует в обмене ресурсами тогда и только тогда, когда этот обмен ему необходим и выгоден; при этом наблюдается скорее не обмен, а перетекание ресурсов к эгоистичному агенту.

Альтруистичный агент a_a априори считается неспособным к построению собственных целей и принимает чужую цель как общую. Такой агент всегда участвует в обмене ресурсами, даже если обмен будет неравнозначным и он от него проиграет.

Ниже вводится вариант модификации формализма ресурсных сетей и определяются ресурсно-целевые сети, где вершины задаются двумя параметрами (типом агента и объемом ресурса), а дуги – двумя видами проводимости (по целям и по ресурсам). Тип агента полностью характеризует его цели, относящиеся к ресурсам, а входные проводимости могут отличаться от выходных.

Определение 2 [Дюндюков, 2010, Дюндюков и др., 2011]. Ресурсно-целевой сетью называется взвешенный ориентированный мультиграф (гиперграф)

$$G_{RO} = \langle A, C, K, RES, W, t \rangle, \quad (4)$$

где множество вершин есть множество агентов A , а множество дуг C разбито на два непересекающихся подмножества: множество целевых связей C_G и множество ресурсных связей C_{RES} : $C = C_G \cup C_{RES}$, $C_G \cap C_{RES} = \emptyset$, t – множество дискретных моментов времени, $t = 0, 1, 2, \dots, n$. Будем обозначать ресурсные связи сплошными стрелками, а целевые связи – пунктирными.

Каждая вершина ресурсно-целевой сети (РЦС) $a_i \in A$ определяется следующими параметрами – тип агента $k_i \in K$ и объем ресурса $res(a_i) \in RES$, а каждая дуга $c_{ij} \in C$ взвешивается с помощью значения проводимости или пропускной способности $w_{ij} \in W$. У любых двух агентов a_i, a_j выделяются проводимости по целям $w_O(a_i, a_j)$ и проводимости по ресурсам $w_{RES}(a_i, a_j)$. В общем случае обмену целями и ресурсами соответствуют направленные навстречу друг другу стрелки орграфа, причем весовые коэффициенты дуг орграфа (значения входных и выходных проводимостей) могут быть различными.

В каждый дискретный момент времени t состояние сети будет определяться вектором состояния $RES(t) = (res(a_1), \dots, res(a_m))$, где m – число агентов в РЦС. Если вектор состояния ресурсно-целевой сети не изменяется со временем, то такое состояние называется устойчивым. Состояние РЦС является асимптотически достижимым, если для заранее заданного малого ζ , $\zeta > 0$, $\forall i = 1, \dots, m$, выполняется неравенство:

$$|res(a_i(t+1)) - res(a_i(t))| < \zeta. \quad (5)$$

При обмене ограниченными материальными ресурсами РЦС полагается замкнутой системой, для которой справедливо равенство:

$$\sum_{i=1}^m res(a_i) = RES_{SUM}, \quad (6)$$

$$RES_{SUM} = const.$$

В общем случае тип агента может описываться тройкой параметров, характеризующих степени благонамеренности, эгоизма и альтруизма, т.е. в ресурсно-целевом графе $K=\{a_b, a_e, a_a\}$. Общая схема возможных взаимодействий агентов дана на рисунке 3.

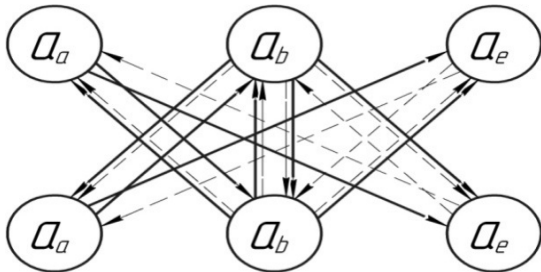


Рисунок 3 – Общая схема взаимодействий между агентами

Отсюда видно, что запрещенными являются взаимодействия между двумя эгоистичными и двумя альтруистичными агентами, поскольку при таких взаимодействиях не может быть образована многоагентная система. Поскольку эгоистичные агенты неспособны к образованию коллективных целей и равноправному обмену ресурсами, они не смогут эффективно взаимодействовать. Создание МАС только из альтруистичных агентов также невозможно из-за того, что они не имеют собственных целей и, как следствие, не смогут построить коллективную цель.

При взаимодействии агентов в МАС возможны четыре варианта взаимодействия, которые приведены на рисунке 4. Для двух агентов различных типов справедливо выражение: $w_{RES}(a_{ej}, a_{ai}) = w_{RES}(a_{ej}, a_{bi}) = \emptyset$, $w_0(a_{aj}, a_{bi}) = w_0(a_{ai}, a_{ej}) = \emptyset$.

Кратко опишем теперь возможные варианты взаимодействия агентов. В случае взаимодействия двух благонамеренных агентов происходит равноправный обмен информацией целевого характера, в результате которого формируется общая цель, а также осуществляется обмен ресурсами. Многоагентная система, состоящая из n подобных агентов, представляется наиболее эффективной для реализации стратегий децентрализованного искусственного интеллекта, когда формируется структура типа «полный граф».

При взаимодействии эгоистичного и альтруистичного агентов первый навязывает второму свою цель и использует для ее достижения чужие ресурсы. Фактически происходит перекачка ресурсов от a_a к a_e , которая может завершиться гибелью a_a , если объем ресурса $res(a_a(t)) < res_{min}$.

В случае взаимодействия благонамеренного и эгоистичного агентов возникает иллюзия обмена ресурсами между агентами. Эгоистичному агенту нужны ресурсы, но в ответ он старается ничего не отдавать. Благонамеренный агент будет избегать такого взаимодействия и участвовать в нем только в

критических для себя случаях. При этом взаимодействие прекращается, если $res(a_b(t))$ близок к res_{min} (наличие «инстинкта самосохранения» у a_b).

При взаимодействии благонамеренного и альтруистичного агентов происходит эффективный обмен ресурсами, причем a_a разделяет цели a_b . В силу своей благонамеренности a_b не допускает ситуации истощения ресурсов у a_a .

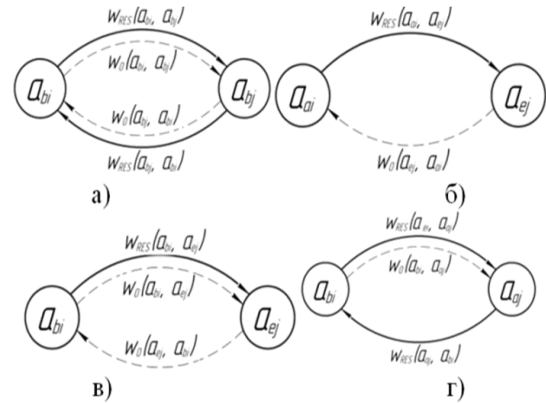


Рисунок 4 – Примеры ресурсно-целевых сетей

- а) обмен между благонамеренными агентами;
- б) обмен между эгоистичным и альтруистичным агентом;
- в) обмен между эгоистичным и благонамеренным агентами;
- г) обмен между благонамеренным и альтруистичным агентами.

Здесь сплошными линиями изображены ресурсные связи, а пунктирными – целевые связи между агентами

Как известно, степенью вершины неориентированного графа называется число ребер, инцидентных данной вершине. Для ресурсно-целевого орграфа следует учитывать как число выходящих из вершины дуг, так и их суммарную пропускную способность. Таким образом, влияние агента a в МАС определяется собственным ресурсом, числом связей с другими агентами (т.е. числом выходящих из вершины a дуг), а также суммарными значениями выходных проводимостей по целям w_0 и ресурсам w_{RES} . Формально влияние агента a в момент времени t определяется имеющимся у него объемом ресурса $res(a(t))$, мощностью множества выходных целевых связей $|C_{Out}(t)|$ и/или выходных ресурсных связей $|C_{RESOut}(t)|$, а также соответствующими величинами суммарной проводимости $W_{Out}(t)$ и $W_{RESOut}(t)$.

Определяющим критерием влияния агента в МАС является его отношение к ресурсу. Чем большим относительным ресурсом обладает агент, тем большее влияние он оказывает на других агентов. Влияние агента в МАС заключается в способности предоставлять необходимые ресурсы, а также находить ресурсы, требуемые для достижения общих целей.

В процессе обмена объем ресурса агента изменяется. Так при уменьшении ресурса агента степень его влияния на МАС уменьшается.

В предельном случае агент может поменять свой тип, например, превратиться из благонамеренного в альтруистичного. Напротив, при значительном

увеличении своего влияния благонамеренный агент может стать эгоистичным.

Можно выделить необходимые условия существования агентов в МАС: 1) $res(a_i(t)) \geq res_{min}$ – условие индивидуального выживания агента; 2) $res(a_i(t)) \leq res_{max}$ – условие социального поведения агента.

Если объем ресурса агента $res(a_i(t)) \geq res_{max}$, то возникает ситуация бесконтрольного возрастания его влияния, что может привести к асоциальному поведению в МАС. Здесь предельным состоянием МАС будет являться ее распад, так как эгоистичный агент не сможет участвовать в обмене ресурсами из-за отсутствия участников обмена.

При обмене ресурсами благонамеренный агент отдает для обмена весь ресурс, который превышает прожиточный минимум, $res(a_i(t)) - res_{min}$. Вершины ресурсно-целевой сети, которым соответствуют благонамеренные агенты, могут рассматриваться как вершины с петлями. Для упрощения дальнейшего рассмотрения взаимодействия агентов примем:

$$res^*(a_i(t)) = res(a_i(t)) - res_{min}, RES_{SUM}^* = RES_{SUM} - \sum_{i=1}^m RES_{min i} \quad (7)$$

Теорема 1. Если ресурсно-целевая сеть состоит из благонамеренных агентов и для каждой из дуг w_0 и w_{RES} равны между собой и $RES_{SUM} \leq W_{RES}$, где W_{RES} – максимально отдаваемый агентами сети ресурс за одну-единственную итерацию, $W_{RES} = \sum_{i=1}^m \sum_{j=1}^m w_{RES}(a_i, a_j) = w_{RES}m(m-1)$, то с течением времени ресурс системы распределится между агентами равномерно и $RES(t) = (\frac{RES_{SUM}}{m}, \dots, \frac{RES_{SUM}}{m})$.

В противном случае, если хотя бы у двух агентов ресурсы не будут равны в начальный момент времени, то выравнивание ресурса в сети не произойдет.

Доказательство. 1) Если $RES_{SUM} > w_{RES}m(m-1)$, то выравнивания не произойдет, так как агенты будут получать и передавать одинаковое количество ресурса.

2) Упорядочим агентов сети таким образом, чтобы $res(a_k(0)) > w_{RES}(m-1)$ и $res(a_{k+1}(0)) \leq w_{RES}(m-1)$, тогда в результате получим:

$$\begin{aligned} res(a_1(0)) &\geq \dots \geq res(a_k(0)) \geq \\ &\geq res(a_{k+1}(0)) \geq \dots \geq res(a_m(0)), \\ res(a_1(0)) &= w_{RES}(m-1) + car(a_1(0)), \dots \\ \dots, res(a_k(0)) &= w_{RES}(m-1) + car(a_k(0)), \quad (8) \\ res(a_{k+1}(0)) &= w_{RES}(m-1) - sel(a_{k+1}(0)), \dots \\ \dots, res(a_m(0)) &= w_{RES}(m-1) - sel(a_m(0)), \end{aligned}$$

где $car(a_1(0)), \dots, car(a_k(0)) > 0$ – излишки ресурса, которые остаются у i -го агента после обмена ресурсами, а $sel(a_{k+1}(0)), \dots, sel(a_m(0)) > 0$ – ресурс, которого не хватает i -му агенту для удовлетворения потребности в обмене в полной мере.

Назовем множество агентов, которые имеют

определенный ресурс и могут удовлетворить потребность в обмене, множеством носителей CAR . Данное множество определяется набором первых k агентов упорядоченного множества, для которых $res(a_i(0)) > w_{RES}(m-1)$. Тогда множество искателей SEL есть множество таких агентов, испытывающих потребность в ресурсе, которые в данный момент времени не могут удовлетворить потребности других агентов в обмене ресурсами, $res(a_i(0)) \leq w_{RES}(m-1)$.

Если сложить суммарный излишек у агентов из множества носителей, то получим суммарный профицит носителей $CAR(t)$.

$$CAR(t) = \sum_{i=1}^k car(a_i(t)).$$

Величину общего ресурса агентов сети в момент времени t , которого не хватает для удовлетворения потребности в обмене, назовем дефицитом искателей $SEL(t)$.

Пример 1. $m = 5$, $w_{RES} = 2$, $w_{RES}(m-1) = 8$, $w_{RES}m(m-1) = 40$, $RES(0) = (17, 14, 7, 5, 2)$, $RES_{SUM} = 45$. Так как $RES_{SUM} > w_{RES}m(m-1)$, то суммарный ресурс сети не распределится равномерно среди агентов. $k=2$, $car(a_1(0)) = 9$, $car(a_2(0)) = 6$, $sel(a_3(0)) = 1$, $sel(a_4(0)) = 3$, $sel(a_5(0)) = 6$.

Просуммировав выражения ресурсов агентов, определенные через величины излишка и необходимого ресурса, получим:

$$\begin{aligned} \sum_{i=1}^k res(a_i(0)) &= k w_{RES}(m-1) + \sum_{i=1}^k car(a_i(t)) = \\ &= k w_{RES}(m-1) + CAR(0), \\ \sum_{i=k+1}^m res(a_i(0)) &= (m-k) w_{RES}(m-1) - \\ &- \sum_{i=k+1}^k sel(a_i(t)) = (m-k) w_{RES}(m-1) \\ &- SEL(0) \end{aligned} \quad (9)$$

Таблица 2 – Изменение ресурсов у агентов

Номер итерации	Номер агента				
	1	2	3	4	5
1	17	14	7	5	2
2	14,5	11,5	5,75	6,25	7
3	13,25	10,25	7,31	7,19	7
4	12,63	9,62	7,55	7,58	7,62
5	12,3	9,3	7,8	7,8	7,8
6	12,16	9,16	7,89	7,89	7,9
7	12,08	9,07	7,95	7,95	7,95
8	12,04	9,05	7,97	7,97	7,97
9	12,015	9,015	7,99	7,99	7,99
10	12	9	8	8	8

Просуммировав эти выражения, получим:

$$\begin{aligned} \sum_{i=1}^k res(a_i(0)) + \sum_{i=k+1}^m res(a_i(0)) &= \sum_{i=1}^m res(a_i(0)) = \\ &= RES_{SUM} = m w_{RES}(m-1) + CAR(0) - SEL(0). \end{aligned} \quad (10)$$

Если k не изменяется с течением времени t , то $SEL(t) - CAR(t) = const = \Delta$. Причем $\Delta = 0$, если $w_{RES}m(m-1) = RES_{SUM}$.

Рассмотрим обмен ресурсами между множеством носителей и множеством искателей. В момент времени $t=1$ CAR отдает SEL величину ресурса $w_{RES}k(m-k)$, а получает от SEL $\frac{w_{RES}k(m-k)(m-1) - kSEL(0)}{(m-1)} = w_{RES}k(m-k) - k\frac{SEL(0)}{(m-1)}$.

Найдем зависимости профицита носителей и дефицита искателей в произвольный момент времени t от их начального значения.

$$CAR(1) = CAR(0) - w_{RES}k(m-k) + w_{RES}k(m-k) - k\frac{SEL(0)}{(m-1)} = CAR(0) - k\frac{SEL(0)}{(m-1)} = CAR(0) - k\frac{\Delta - CAR(0)}{(m-1)} \quad (11)$$

$$= CAR(0)\frac{m-k-1}{m-1} - \frac{k\Delta}{m-1}$$

$$CAR(t) = CAR(0)\left(\frac{m-k-1}{m-1}\right)^t - \frac{k\Delta}{m-1}\sum_{i=1}^{t-1}\left(\frac{m-k-1}{m-1}\right)^i, \quad (12)$$

$$SEL(1) = SEL(0) - k\frac{SEL(0)}{(m-1)} = SEL(0)\frac{m-k-1}{(m-1)}, \quad (13)$$

$$SEL(2) = SEL(1)\frac{m-k-1}{(m-1)} = SEL(0)\left(\frac{m-k-1}{(m-1)}\right)^2, \quad (14)$$

$$SEL(t) = SEL(0)\left(\frac{m-k-1}{(m-1)}\right)^t. \quad (15)$$

Рассмотрим 3 возможных случая: $k=0, k=1, k>1$.

Случай 1. Пусть $k=0$. В данном случае множество носителей пусто и $\forall i = 1, \dots, m, res(a_i(0)) \leq w_{RES}(m-1)$. Все агенты сети отдают свой ресурс для обмена. Поэтому в каждый последующий момент времени ресурс агента будет состоять из того, что он получил от других агентов:

$$res(a_1(1)) = \frac{RES_{SUM} - res(a_1(0))}{m-1}, \dots, res(a_m(1)) = \frac{RES_{SUM} - res(a_m(0))}{m-1}. \quad (16)$$

Так как $res(a_1(0)) \geq \dots \geq res(a_m(0))$, то $res(a_m(1)) \geq \dots \geq res(a_1(1))$. Максимальная разность между ресурсами РЦС в момент времени $t=1$ будет равна:

$$res(a_m(1)) - res(a_1(1)) = \frac{res(a_1(0)) - res(a_m(0))}{m-1}. \quad (17)$$

$$\lim_{t \rightarrow \infty} res(a_m(1)) - res(a_1(1)) = \lim_{t \rightarrow \infty} \frac{res(a_1(0)) - res(a_m(0))}{m-1} = 0. \quad (18)$$

Пример 2. $m=5, w_{RES}=2, w_{RES}(m-1)=8, w_{RES}m(m-1)=40, RES(0)=(7,5,4,3,2), RES_{SUM}=21$.

Таблица 3 – Изменение ресурсов у агентов

Номер итерации	Номер агента				
	1	2	3	4	5
1	7	5	4	3	2
2	3,5	4	4,25	4,5	4,75
3	4,38	4,25	4,19	4,12	4,06
4	4,16	4,19	4,2	4,22	4,23
5	4,21	4,2	4,2	4,195	4,195
6	4,2	4,2	4,2	4,2	4,2

Случай 2. $k=1$. В этом случае $res(a_1(0)) = w_{RES}(m-1) + car(a_1(0))$, а для всех остальных агентов сети справедливо условие $res(a_i(0)) \leq$

$w_{RES}(m-1)$. Причем $car(a_1(t)) = CAR(t)$.

В данном случае необходимо рассмотреть два возможных варианта: $RES_{SUM} = w_{RES}m(m-1)$ и $RES_{SUM} > w_{RES}m(m-1)$.

Случай 2.1. $RES_{SUM} = w_{RES}m(m-1), \Delta=0$. Профицит носителей и дефицит искателей равен:

$$CAR(t) = CAR(0)\left(\frac{m-k-1}{m-1}\right)^t, \quad (19)$$

$$SEL(t) = SEL(0)\left(\frac{m-k-1}{(m-1)}\right)^t. \quad (20)$$

$$res(a_1(0)) = w_{RES}(m-1) + car(a_1(0)) \quad (21)$$

$$res(a_i(0)) = w_{RES}(m-1) - sel(a_i(0)), \quad (22)$$

$\forall i=2, \dots, m, \lim_{t \rightarrow \infty} CAR(t) = \lim_{t \rightarrow \infty} SEL(t) = 0$,

$\lim_{t \rightarrow \infty} RES(t) = (w_{RES}(m-1), \dots, w_{RES}(m-1))$.

Причем $res(a_1(0))$ стремится к 0 сверху, а $res(a_i(0))$ – снизу.

Пример 3. $m=5, w_{RES}=2, w_{RES}(m-1)=8, w_{RES}m(m-1)=40, RES(0)=(15,8,7,6,5), RES_{SUM}=41$.

Таблица 4 – Изменение ресурсов у агентов

Номер итерации	Номер агента				
	1	2	3	4	5
1	15	8	7	9	5
2	13,5	6,5	6,75	7	7,25
3	12,38	7,25	7,19	7,12	7,06
4	11,53	7,34	7,36	7,38	7,39
5	10,9	7,53	7,53	7,52	7,52
6	10,42	7,64	7,64	7,65	7,65
...
22	9,012	7,997	7,997	7,997	7,997
23	9	8	8	8	8

Случай 2.2. $RES_{SUM} > w_{RES}m(m-1)$. Число носителей k не меняется, если $CAR(t) = SEL(t) - \Delta$. В выражении $CAR(t) = CAR(0)\left(\frac{m-k-1}{m-1}\right)^t - \frac{k\Delta}{m-1}\sum_{i=1}^{t-1}\left(\frac{m-k-1}{m-1}\right)^i$ первый член с течением времени стремится к 0, а второй возрастает. Наступит такой момент времени t , для которого $CAR(t) \leq 0$ и число агентов в множестве носителей будет равно 0. Следовательно, будет выполнено условие случая 1.

Пример 4. $m=5, w_{RES}=2, w_{RES}(m-1)=8, w_{RES}m(m-1)=40, RES(0)=(12,5,4,3,2), RES_{SUM}=26$.

Таблица 5 – Изменение ресурсов у агентов

Номер итерации	Номер агента				
	1	2	3	4	5
1	12	5	4	3	2
2	7,5	4,25	4,5	4,75	5
3	4,62	5,38	5,38	5,31	5,31
4	5,34	5,14	5,16	5,17	5,19
5	5,16	5,21	5,21	5,21	5,21
6	5,208	5,198	5,198	5,198	5,198
7	5,2	5,2	5,2	5,2	5,2

Случай 3. $k > 1$. Возможны два варианта:
 $car(a_1(0)) = car(a_2(0)) = \dots = car(a_k(0))$ и
 $car(a_1(0)) \geq car(a_2(0)) \geq \dots \geq car(a_k(0))$.

Случай 3.1. $k > 1$.

$$car(a_1(0)) = car(a_2(0)) = \dots = car(a_k(0)).$$

Обмен между агентами из множества носителей будет равновесным, агенты отдают друг другу и получают ресурс $w_{RES}(m-1)$. Обмен между множествами искателей и носителей описывается формулами:

$$CAR(t) = CAR(0) \left(\frac{m-k-1}{m-1}\right)^t - \frac{k\Delta}{m-1} \sum_{i=1}^{t-1} \left(\frac{m-k-1}{m-1}\right)^i, \quad (23)$$

$$SEL(t) = SEL(0) \left(\frac{m-k-1}{m-1}\right)^t. \quad (24)$$

Если $RES_{SUM} = w_{RES}m(m-1)$, $\Delta = 0$. Тогда:

$$CAR(t) = CAR(0) \left(\frac{m-k-1}{m-1}\right)^t, \quad (25)$$

$$SEL(t) = SEL(0) \left(\frac{m-k-1}{m-1}\right)^t. \quad (26)$$

$$\lim_{t \rightarrow \infty} CAR(t) = \lim_{t \rightarrow \infty} SEL(t) = 0, \quad (27)$$

$$\lim_{t \rightarrow \infty} RES(t) = (w_{RES}(m-1), \dots, w_{RES}(m-1)). \quad (28)$$

Если $RES_{SUM} > w_{RES}m(m-1)$, то в выражении $CAR(t) = CAR(0) \left(\frac{m-k-1}{m-1}\right)^t - \frac{k\Delta}{m-1} \sum_{i=1}^{t-1} \left(\frac{m-k-1}{m-1}\right)^i$ наступит такой момент времени t , для которого $CAR(t) \leq 0$. Число агентов в множестве носителей будет равно 0. Следовательно, будет выполнено условие случая 1.

Случай 3.2. $k > 1$, $car(a_1(0)) \geq car(a_2(0)) \geq \dots \geq car(a_k(0))$. Так как в процессе обмена ресурсами между агентами разность $car(a_j(0)) - car(a_i(0))$ не изменяется, а $\lim_{t \rightarrow \infty} CAR(t) = 0$, то это приведет к тому, что число носителей k уменьшится и случай 3.2 сведется к случаю 3.1.

Теорема 2. Если $RES_{SUM} > w_{RES}m(m-1)$, то предельному состоянию ресурсно-целевой сети будет соответствовать вектор состояния

$$RES(t) = (res(a_1(0)) - p, \dots, res(a_l(0)) - p, \dots, w_{RES}m-1, \dots, w_{RES}m-1), \quad (29)$$

где $p = \frac{SEL(t)}{k}$, $l = k$, если $car(a_k(0)) \geq \frac{SEL(t)}{k}$, иначе $p = \frac{\sum_{i=1}^k car(a_i(0)) + SEL(t)}{1}$, $l \leq k$ и $car(a_l(0)) \geq \frac{SEL(t)}{k}$.

Доказательство. Множество носителей CAR не может увеличиваться со временем, но может уменьшаться, если часть агентов перейдет из множества CAR в SEL . В каждый момент времени агенты из множества носителей передают агентам множества искателей одинаковый ресурс $w_{RES}(m-1)$ и получают одинаковый ресурс. Поэтому из ресурсов агентов множества носителей в каждый момент времени вычитается одинаковый ресурс. Для доказательства теоремы необходимо доказать, что предельному состоянию РЦС будет соответствовать вектор состояния $RES(t)$, вид которого определен выше. Рассмотрим 2 случая.

Случай 1. Число агентов в множестве носителей не меняется со временем, $l=k$. Рассмотрим выражение для дефицита искателей, выраженного

для произвольного момента времени t . $SEL(t) = SEL(0) \left(\frac{m-k-1}{m-1}\right)^t$. Так как k не изменяется, то $\lim_{t \rightarrow \infty} SEL(t) = 0$.

Пример 5. $m=5$, $w_{RES} = 2$, $w_{RES}(m-1) = 8$, $w_{RES}m(m-1) = 40$, $RES(0) = (20, 15, 7, 5, 4)$, $RES_{SUM} = 51$, $SEL(0) = 8$.

Множество искателей стремится к выравнованию ресурса и предельное состояние существует. В предельном состоянии дефицит искателей увеличится на величину $SEL(0)$, а профицит носителей уменьшится на величину $SEL(0)$. Поскольку носители в процессе обмена отдают одинаковый ресурс, то из ресурса каждого агента будет вычтена величина $p = \frac{SEL(t)}{k}$. Первая часть теоремы доказана.

Таблица 6 – Изменение ресурсов у агентов

Номер итерации	Номер агента				
	1	2	3	4	5
1	20	15	7	5	4
2	18	13	6,25	6,75	7
3	17	12	7,44	7,31	7,25
4	16,5	11,5	7,64	7,67	7,69
5	16,25	11,25	7,84	7,83	7,83
6	16,13	11,13	7,91	7,91	7,92
...
22	16	11	8	8	8

Случай 2. Если $car(a_k(0)) < \frac{SEL(t)}{k}$, то в некоторый момент времени агент a_k перейдет из множества носителей в множество искателей. Множество носителей уменьшится на одного агента, данный процесс будет продолжаться до тех пор, пока число агентов из множества носителей не станет равным 1. Новое состояние РЦС можно рассматривать как сеть с множеством искателей, состоящим из 1 агента. Для данной сети будут выполнены условия случая 1.

Пример 6. $m = 5$, $w_{RES} = 2$, $w_{RES}(m-1) = 8$, $w_{RES}m(m-1) = 40$, $RES(0) = (20, 10, 7, 5, 4)$, $RES_{SUM} = 46$, $SEL(0) = 8$.

Таблица 7 – Изменение ресурсов у агентов

Номер итерации	Номер агента				
	1	2	3	4	5
1	20	10	7	5	4
2	18	8	6,25	6,75	7
3	17	7	7,44	7,31	7,25
4	16,25	7,5	7,39	7,42	7,44
5	15,69	7,56	7,59	7,58	7,58
6	15,27	7,69	7,68	7,68	7,68
...
22	14,012	7,997	7,997	7,997	7,997
23	14	8	8	8	8

3. Программная реализация.

Программа поддержки построения ресурсно-целевых сетей была реализована на языке Python. Ее интерфейс представлен на рисунке 5. В качестве вводимых параметров берутся количество агентов, участвующих в процессе обмена, величины минимального и максимального ресурса у агентов, что соответствует условиям индивидуального выживания и асоциального поведения. Также пользователю предлагается ввести типы агентов, начальное распределение ресурсов у них и величины проводимостей по ресурсам и целям.

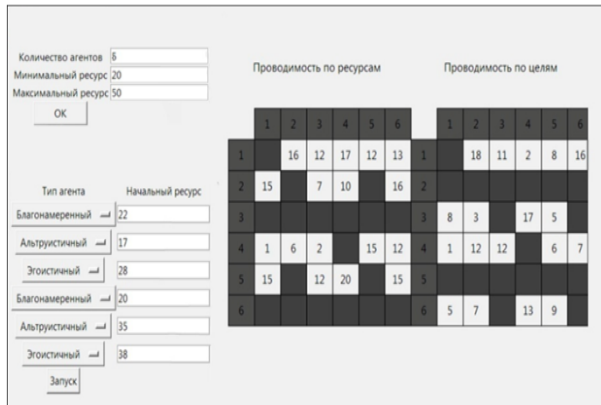


Рисунок 5 – Интерфейс программы поддержки построения ресурсно-целевых сетей на языке Python

На рисунке 6 приведен алгоритм реализованной программы. Процесс обмена ресурсами в данной программе зависит от типа агента, имеющегося у него ресурса и значений ресурсной и целевой проводимостей. Благонамеренный агент отдает для обмена ресурс $Res(a_i(t)) - Res_{min}$. Альтруистичный агент для обмена жертвует весь имеющийся ресурс, даже если на следующей итерации его ресурс окажется меньше прожиточного минимума, что приведет к его гибели. Эгоистичный агент не отдает ресурс для обмена. Величина отдаваемого агентом a_i агенту a_j ресурса $Res(a_i, a_j)$ распределяется пропорционально проводимостям по целям:

$$Res(a_i, a_j) = \frac{w_o(a_i, a_j)}{\sum_{j=1}^n (w_o(a_i, a_j))} Res(a_i(t))$$

В случае равенства проводимостей по целям между агентами, ресурс распределяется пропорционально проводимости по ресурсам.

Данная программа позволяет рассматривать различные типовые структуры ресурсно-целевых графов (рисунок 7), такие как «кольцо», «центр», «колесо», «полный граф», а также произвольные структуры. При структуре типа «кольцо» обмен ресурсами между агентами происходит по цепочке. Каждый агент имеет возможность обмениваться только с двумя соединенными с ним агентами. Перераспределение ресурсов при данной организационной структуре замедляется. Система распадается на отдельные составляющие, если в ней появляются эгоистичные агенты.

При организационной структуре типа «центр» головным агентом для благоприятного развития

сети должен стать благонамеренный агент, рационально перераспределяющий ресурсы.

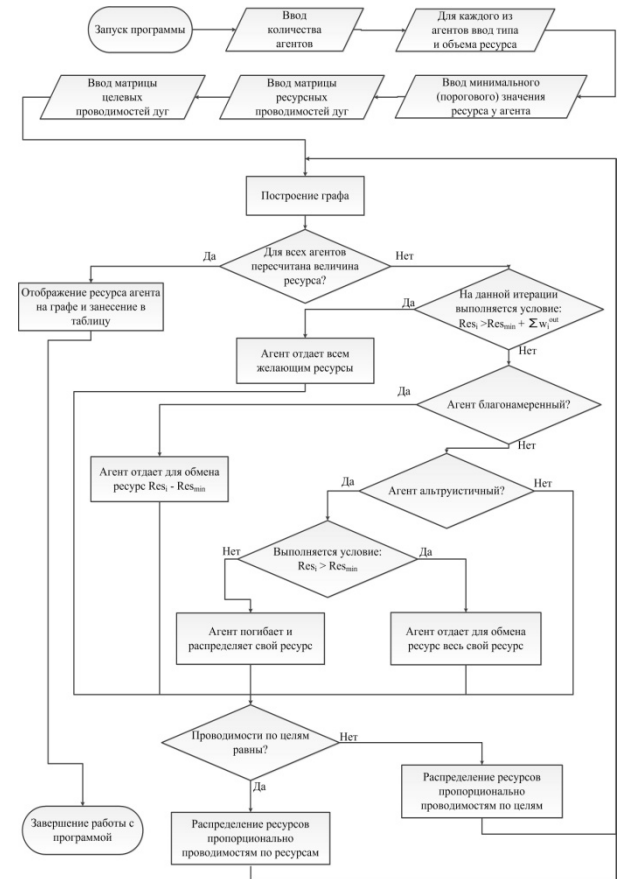


Рисунок 6 – Алгоритм программы поддержки построения ресурсно-целевых сетей

В случае организационной структуры типа «колесо» выбор в качестве центра эгоистичного агента приведет к ее быстрому превращению в структуру «звезда» и захвату ресурсов всех узлов центральным узлом.

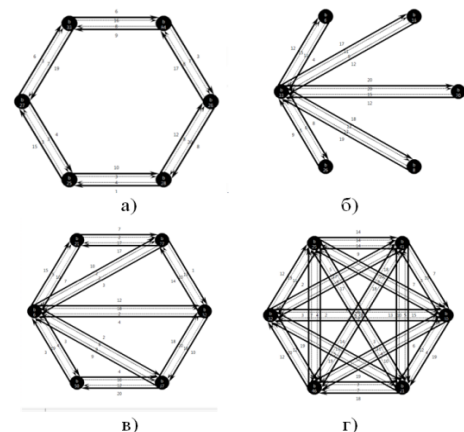


Рисунок 7 – Примеры структур ресурсно-целевых сетей: а) «кольцо»; б) «центр»; в) «колесо»; г) «полная сеть»

Наиболее благоприятными организационными структурами являются структура «полная сеть» и близкие к ней структуры, когда агент, стремясь максимизировать свои связи, имеет право выбора того, с кем обмениваться ресурсами в зависимости

от своих целей. Данная структура «горизонтальной организации» лучше всего работает при комбинации благонамеренных и альтруистичных агентов.

ЗАКЛЮЧЕНИЕ

В статье предложены варианты наглядного представления процессов взаимодействия агентов, формирования и развития многоагентных систем с помощью аппарата ресурсно-целевых сетей. Показано, что этот аппарат может стать удобным средством семантико-прагматического анализа и онтологического моделирования различных систем группового (коллективного, децентрализованного, централизованного) искусственного интеллекта.

Работа выполнена при финансовой поддержке РФФИ, проекты №10-01-00844 и №11-07-13165-офи-м-2011-РЖД

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Берж, 1962] Берж, К. Теория графов и ее применения: Перевод с франц. – М.: Изд-во иностранной литературы, 1962.

[Волков, 1986] Волков, А.М. Основы структурно-функционального анализа операторской деятельности. – М.: Изд-во МАИ, 1986.

[Дюндюков, 2010] Дюндюков, В.С. Моделирование взаимодействия интеллектуальных агентов: применение ресурсных графов/ В.С.Дюндюков// Труды международного конгресса по интеллектуальным системам и информационным технологиям (AIS-IT'2010, Дивноморское, 2-9 сентября 2010 г.). – М.: Физматлит, 2010. – Т.1. – С.204-210.

[Дюндюков и др., 2011] Дюндюков, В.С. Ресурсно-целевые сети: использование в многоагентных системах/ В.С.Дюндюков, В.Б. Тарасов// Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов VI-й Международной научно-практической конференции (Коломна, 16-19 мая 2011 г.). – М.: Физматлит, 2011. – Т.1. – С.483-495.

[Кузнецов, 2009] Кузнецов, О.П. Однородные ресурсные сети. I. Полные графы/ О.П.Кузнецов// Автоматика и телемеханика. – 2009. – № 11. – С.136-147.

[Кузнецов и др., 2010] Кузнецов, О.П. Двусторонние ресурсные сети – новая потоковая модель/ О.П.Кузнецов, Л.Ю.Жилиякова// Доклады Академии наук. – 2010. – Т.433, №5. – С.609-612.

[Тарасов, 2002] Тарасов, В.Б. От многоагентных систем к интеллектуальным организациям. – М.: Эдиториал УРСС, 2002.

[Тарасов В.Б., 2010] Тарасов, В.Б. Методы и модели алгебраической логики в анализе взаимоотношений и переговоров агентов в многоагентных системах/ В.Б.Тарасов// Сборник трудов X-й международной научной конференции им. Т.А.Таран «Интеллектуальный анализ информации» (ИАИ-2010, Киев, 18-21 мая 2010 г.). – Киев: Просвіта, 2010. – С.304-316.

[Форд и др., 1973] Форд, Л.Р., Фалкерсон, Д.Р. Потоки в сетях: Пер. с англ. – М.: Мир, 1973.

[Харари Ф., 1973] Харари, Ф. Теория графов: Пер. с англ. – М.: Мир, 1973.

GENERATION OF MULTI-AGENT SYSTEMS BY USING RESOURCE-GOAL NETWORKS

Dyundyukov V.S. , Tarassov V.B.

Moscow Bauman State Technical University

Vsd89@yandex.ru

Tarasov@rk9.bmstu.ru

The concept of resource-goal network is introduced for taking into account various agents types and theirs

goals while exchanging resources. Primarily main reasons and characteristics of agents interaction together with some conditions for generating multi-agent systems are stated. Basic cases of interactions between agents are analyzed by using bilateral signed graphs. Some resource definitions, classifications and properties are discussed. The definition of resource-goal network (RGN) is introduced on the basis of earlier specified agent classes; some key situations of exchanging resource by agents of various classes are considered. Two theorems concerning RGN boundary states are formulated and proved. Some appropriate illustrative examples are constructed. Finally the software tools to support the development of RGN with using Python language are presented.

Keywords: Artificial Intelligence, Agent, Multi-Agent System, Agents Interaction, Resource, Resource Exchange, Agent Influence, Bilateral Weighted Graph, Resource-Goal Network

INTRODUCTION

This paper faces the problems of building resource-goal networks and their using in Artificial Intelligence to construct multi-agent systems. Usually in graph theory network is defined as a weighted graph (multigraph, hypergraph) $G = \langle X, C, W \rangle$, where X is a vertex set, C is an arc set, and $W: C \rightarrow \mathbf{R}^+$, \mathbf{R}^+ is the set of all non-negative real numbers. For instance, transportation set is given as a finite graph without loops, where each arc is equipped with a non-negative integer w , called arc conductivity. A classical algorithm to solve the problems of calculating flows in the networks is Ford-Fulkerson's algorithm.

Recently O.P.Kuznetsov has suggested the formalism of resource networks defined as bilateral oriented graphs, in which vertices are equipped with non-negative numbers $q_i(t)$ called resources (t is discrete time) and arcs are characterized by non-negative numbers seen as conductivities. To differ from transportation networks and classical Ford-Fulkerson's flow model, where the flows are located in the arcs, the resources are placed in the vertices, and the resource exchange depends on arc conductivity.

The development of multi-agent systems (MAS) supposes organizing the interactions between agents. Here agent is viewed as an open active goal-directed object possessing its own behavior. The kernel of any agent is given by a quadruple «goal-resource-perception-action». The interaction between agents to create multi-agent system depends on the following basic parameters: 1) agent type; 2) compatibility of agent goals or intentions; 3) statute of agents and amount of available resources; 4) agents experience related to some problem area; 5) agents commitments.

Thus, the development of MAS requires distributed perception of external environment, joint goals generation, resource exchange and concerted actions. So the introduction of resource-goal networks with various types of vertices and two types of arcs denoted by continuous lines and dotted lines respectively allows us to take into account agents' goals while exchanging resources and reflect asymmetries in these exchanges.

MAIN PART

The paper considers interactions between agents from the viewpoint of purposeful resource sharing and exchanging. First of all agents interaction area is represented by a simple bi-polar scale $L_3 = \{-, 0, +\}$. Generally, three classes of relations between agents are revealed: a) mutual (symmetrical) relations; b) weakly contrasting (anti-symmetrical) relations; c) contrasting (oblique symmetrical) relations.

A crucial idea of RGN is based on the agents classification by two criteria: agent relation to himself and agent relation to other agents. Here these criteria are re-interpreted: first one is seen as agent's possibility (+) or impossibility (-) of generating his individual goal and second one as agent's will to exchanging resources or agent's possibility (+) – impossibility (-) to form collective goal. So four types of agents appear: 1) benevolent agent given by (+,+); 2) self-interested (egoist) agent denoted by (+, -); 3) altruistic agent (-, +); 4) destroyer (kamikaze) agent (-, -). In other words, benevolent agent has his own goal and is capable to participate in forming collective goal and mutual resource exchanges necessary for multi-agent system effective work. Self-interested agent is focused in his proper goal and neglects goals and interests of other agents; hence, in interaction processes he tends to impose his goal and appropriate all resources. It is obvious that multi-agent system cannot exist in case of many self-interested agents. Altruistic agent is ready to give his resources even in dangerous situations; on the contrary, he is not able to generate his own goal and easily accepts the goal of other agent. Like the selfish team including only self-interested agents, the society of altruistic agents cannot develop any collective goal. The properties of kamikaze agent are omitted in this paper.

This classification of agents underlies the definition of RGN below. By resource-goal network we mean a weighted directed multigraph (hypergraph)

$$G_{RO} = \langle A, C, K, RES, W, t \rangle,$$

where the set of vertices A is interpreted as the set of agents, the set of arcs C consists of goal connections C_O and resource connections C_{RES} , i.e. it is a partition $C = C_O \cup C_{RES}$, $C_O \cap C_{RES} = \emptyset$, t is the discrete time, $t = 0, 1, 2, \dots, n$. Each agent $a_i \in A$ being RGN-vertex is given by the following parameters – agent type $k_i \in K$ and the amount of available resources $res(a_i) \in RES$, whereas each arc $c_{ij} \in C$ is weighted by the value of conductivity $w_{ij} \in W$. Here two types of conductivity are specified: conductivity by goal $w_O(a_i, a_j)$ and conductivity by resources $w_{RES}(a_i, a_j)$. The resource exchange and goal exchange are modeled by directed graph arrows, where arc weights (input or output conductivity values) may be different.

The state of the resource-goal network for each discrete time t is given by a state vector $RES(t) = (res(a_1), \dots, res(a_m))$, where m is the number of agents in RGN. If this state vector does not change, then the network state is called stable.

In course of resource exchange process RGN is supposed to be a closed system, where the conservation law for material resources holds. Contrarily, in case of

information resources synergetic (superadditivity) phenomena arise.

Generally agent type may be viewed as a triple $K = \{a_b, a_e, a_a\}$, where a_b, a_e, a_a stand for degrees of benevolence, self-interest and altruism respectively.

Let us present basic cases of interactions between agents. The interaction between benevolent agents supposes individual goal information exchange to generate collective goal. Here mutually profitable resource exchange is performed. Multi-agent system constructed on the basis of such interacting agents seems to be very effective from the viewpoint of decentralized artificial intelligence strategies; the corresponding structure is a complete network.

In course of interaction between self-interested and altruistic agents the first one imposes his goal and tends to capture all resources. In fact, we observe resource transfer from a_a to a_e ; it can bring about the death of a_a , if $res(a_a(t)) < res_{min}$.

The interaction scenario for benevolent and self-interested agents is similar. Self-interested agent tends to appropriate all resources, but in response he mainly disseminates his beliefs and goal. So benevolent agent is not interested in such interactions and accepts them only in critical situation. In any case, he is provided with the mechanism of self-preservation: the interaction is interrupted in case of $res(a_b(t)) \rightarrow res_{min}$.

Finally, the interaction between benevolent and altruistic agents, when a_a shares the goal of a_b , supposes an efficient resource exchange. Here a_b does not accept the resource extinction for a_a due to his benevolence.

Another important agent characteristic in RGN is his influence degree. Here we need taking into account both vertex degree and total conductivity. Hence, agent influence in MAS is given by agent's own resource, the number of connections with other agents and sum values of resource conductivities and goal conductivities.

As a result, two theorems about resource distribution in MAS have been proved.

The software program to support the construction of resource-goal network has been developed with using Python language.

CONCLUSION

Some ways of visual representation to model interaction processes between agents, as well as multi-agent systems generation and development have been proposed in the paper with using resource-goal networks. It has been shown that this formalism may become a quite suitable tool for semantic-pragmatic analysis and ontological modeling while considering various systems of collective (decentralized or centralized) artificial intelligence.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8

ГРАНУЛЯРНЫЕ, НЕЧЕТКИЕ И ЛИНГВИСТИЧЕСКИЕ ОНТОЛОГИИ ДЛЯ ОБЕСПЕЧЕНИЯ ВЗАИМОПОНИМАНИЯ МЕЖДУ КОГНИТИВНЫМИ АГЕНТАМИ

Тарасов В.Б., Калуцкая А.П., Святкина М.Н.

*Московский государственный технический университет им.Н.Э.Баумана,
г. Москва, Россия*

tarasov@rk9.bmstu.ru

k_a_p@rbcmail.ru

maria.svyatkina@gmail.com

Рассмотрена система онтологий для когнитивных агентов. Введено и формализовано понятие гранулярной метаонтологии как алгебраической системы. Предложены варианты представления гранулярных множеств. Дана подборка определений нечетких онтологий как разновидностей гранулярных онтологий. Разработаны методика и алгоритм построения нечеткой онтологии для сложной, неоднородной проблемной области. Описан вариант построения лингвистических онтологий для эффективной коммуникации когнитивных агентов.

Ключевые слова: агент когнитивный, взаимодействие агентов, онтология, онтология нечеткая, гранула, метаонтология гранулярная, кооперация агентов, взаимопонимание

ВВЕДЕНИЕ

В последние годы в информатике и искусственном интеллекте все большее внимание уделяется проблемам построения онтологий – явных формализованных моделей предметных областей, служащих для реализации единого подхода к концептуальному моделированию этих областей и совместного использования знаний в профессиональном сообществе. Сегодня онтологии находят широкое применение в самых различных сферах прикладной информатики, включая управление знаниями в организационных сетях, семантический веб, компьютерную обработку естественного языка, интеллектуальный анализ данных и обнаружение знаний, электронное обучение, многоагентные системы.

При разработке онтологии требуются ответы на следующие вопросы. Какова природа проблемной области, которую следует описать с помощью онтологии? Кто является пользователем этой онтологии? Какова цель ее использования? Какие задачи решаются на основе данной онтологии? Какие средства нужны для ее разработки и поддержки?

В нашей работе главными пользователями онтологии (точнее, системы онтологий) полагаются естественные и искусственные когнитивные агенты, а основной целью построения онтологии является обеспечение взаимопонимания и совместной работы таких агентов. Здесь онтология предстает в различных ипостасях: и как метод представления информации, и как способ интеграции различных

моделей знаний и управления знаниями в сети, и как формальный инструмент семантического анализа предметной области, и как средство поддержки коммуникации между агентами, в частности, реализации диалога между человеком и искусственным агентом.

В настоящей статье рассмотрены проблемы построения системы онтологий для сообщества когнитивных агентов. Предварительно приводятся необходимые сведения из теории агентов, уточняются понятие когнитивного агента и его главные свойства, обсуждаются особенности информационного взаимодействия (кооперации) естественного и искусственного когнитивных агентов в процессе диалогового управления. С помощью ментальной карты строится общая картина онтологического подхода к инженерии взаимодействий между агентами и разработке многоагентных систем. На основе этой картины показаны трудности построения единственной онтологии и предложена система онтологий для когнитивных агентов.

Центральное место в работе занимает концепция формирования гранулярной метаонтологии и гранулярных онтологий, для обоснования которой изложены элементы единой теории грануляции информации когнитивными агентами, гранулярных вычислений и вычислений со словами на базе обобщенных ограничений. Предложен вариант формализации гранулярных метаонтологий как алгебраических систем, сделан краткий обзор существующих определений нечетких онтологий и введены новые определения, положенные в основу

разработанной авторами методики построения нечеткой онтологии. Также введено понятие нечеткой лингвистической онтологии, опирающееся на расширение лингвистической переменной Л.Заде.

1. Когнитивные агенты

Под *агентом* понимают открытую, активную, целенаправленную систему, которая способна сама формировать собственное поведение в неполностью определенной среде. Выделяются естественные и искусственные, физические и виртуальные, реактивные и интеллектуальные агенты [Тарасов 2002]. Так физические агенты работают в материальном мире (реальном физическом пространстве), а виртуальные агенты реализуются в некоторой программной среде. Примерами искусственных физических агентов могут служить автономные роботы, функционирующие в экстремальных условиях (в космосе, под землей или под водой), а среди программных агентов можно отметить инфоботов (т.е. информационных роботов, позволяющих снизить информационную нагрузку у человека). Поведение реактивных агентов определяется простейшими побуждениями и стимульно-реактивными связями, в то время как создание интеллектуальных агентов предполагает построение внутренней модели внешней среды на уровне мнений и знаний, а также организацию рассуждений в интересах планирования и осуществления действий.

В чисто базовых характеристиках агента обычно включают следующие свойства [Тарасов, 2002; Wooldridge, 2002; Wooldridge et al., 1995]: 1) активность; 2) реактивность; 3) автономность; 4) общительность; 5) потребностно-мотивационный потенциал, определяющий формирование целей, стремлений и предпочтений. Для интеллектуальных агентов в этот перечень следует добавить такие качества как формирование мнений, знаний и рассуждений, прогнозирование ситуации, принятие решений и планирование действий.

Если искусственные интеллектуальные агенты наделены собственными механизмами мотивации и способны формировать собственные цели (целеустремленные агенты), то их называют интенциональными. В противном случае, когда искусственные интеллектуальные агенты получают целеуказания от естественных агентов, т.е. являются целенаправленными, их именуют рефлекторными (понимающими и реализующими цели и интересы пользователей).

Интеллектуальные агенты подразделяются на когнитивные, делиберативные и коммуникативные (рисунок 1). В случае чисто коммуникативных агентов внутренняя модель мира превращается, главным образом, в модель общения, состоящую из моделей участников, процесса и желаемого результата общения [Попов, 2004]. Делиберативные (рассуждающие) агенты способны проводить достаточно сложные рассуждения различных типов (например, абдуктивные, дедуктивные, по аналогии)

и на их основе принимать самостоятельные решения или выполнять действия, изменяющие среду.



Рисунок 1 – Классы интеллектуальных агентов

В настоящей работе главное внимание уделяется искусственным когнитивным агентам, обладающим развитой функцией познания, которая обеспечивает построение внутренних моделей внешней среды (рисунок 2), в частности, моделей других агентов, а также моделей своего состояния.



Рисунок 2 – Архитектура искусственного когнитивного агента

Основными когнитивными процессами являются процессы восприятия среды, построения ее обобщенного представления, понимания закономерностей взаимодействия и поведения, обучения. Сюда же относятся процессы распределения ресурсов (в частности, внимания), прогнозирования и планирования поведения, формирования рассуждений о собственных состояниях и состояниях других объектов и агентов.

Ключевые особенности процессов познания, которые необходимо учитывать при разработке искусственных когнитивных агентов, таковы:

- 1) познание представляет собой открытую систему, которая базируется как на имеющемся знании, так и на восприятии текущих данных;
- 2) познание порождает гипотезы, а не выводы; эти гипотезы нуждаются в подтверждении или опровержении;
- 3) познание среды неотделимо от организации действия агента (как информационного процесса, локального изменения среды или физического перемещения).

Важнейшей особенностью когнитивных агентов является их способность эффективно работать при наличии таких факторов как: локальное *восприятие* среды, неточность и неполнота ее внутреннего *представления*; неоднозначность и изменчивость *мнений*, неясность и противоречивость *намерений* и *целей*, *подготовка* и *принятие решений* в условиях неопределенности, фрагментарность *планов* действий, принципиальная неполнота, неточность и ненадежность информации, получаемой от других агентов [Тарасов, 2004].

Итак, когнитивная система искусственного агента должна осуществлять мониторинг окружающей среды и получать оперативную информацию с помощью сенсорной подсистемы. В то же время, искусственные когнитивные агенты (ИКА) должны иметь возможность общения (диалога) с пользователями на ограниченном естественном языке. Любой диалог предполагает обмен сообщениями, связанный с изменением задач и состояний агентов (рисунок 3).

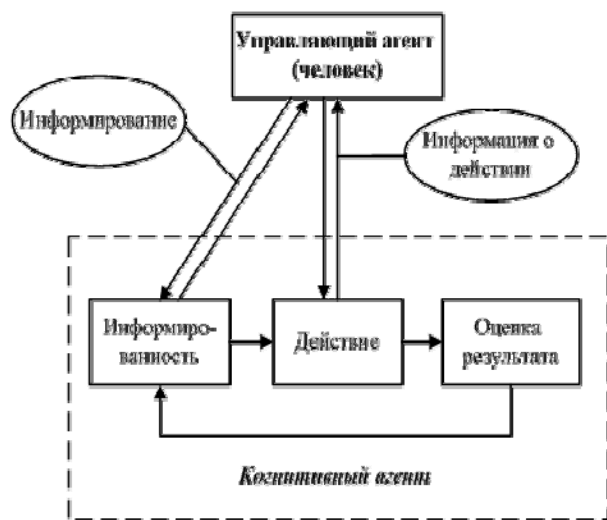


Рисунок 3 – Общая модель диалогового управления

Диалог человека с искусственным агентом включает как целеуказания и инструкции, передаваемые человеком агенту, так и обратную связь – сообщения человека с просьбой уточнить исходные инструкции, а также сведения о текущей ситуации или информацию о достижении поставленной цели [Ющенко, 2009].

Статус искусственного агента как когнитивной системы определяется получением и интеграцией разнородной информации, поступающей из разных источников, включая: 1) человека-пользователя; 2) собственную базу данных/знаний агента; 3) датчики сенсорной системы агента (рисунок 2).

2. Система онтологий для когнитивных агентов

Гибкое интеллектуальное управление и диалоговый интерфейс – отличительные черты когнитивных агентов. При этом главная проблема состоит не столько в техническом обеспечении диалога, сколько в создании общего языка и единого «поля знаний», необходимых для взаимопонимания, координации и совместной работы естественных и искусственных агентов, а также для синтеза многоагентных систем. Поэтому все большую популярность обретает онтологический подход, который предполагает формирование системы онтологий для когнитивных агентов.

Онтологию часто определяют, следуя Т.Груберу, как «спецификацию разделяемой разными людьми концептуализации» [Gruber, 1993] или, иначе, по Н.Гуарино, как «логическую теорию, основанную на концептуализации» [Guarino, 1995]. В последнем случае онтология состоит из словаря терминов, образующих таксономию, их определений и атрибутов, а также связанных с ними аксиом и правил вывода. Связи между простыми онтологиями и логикой Аристотеля показаны в статье [Плесневич, 2010]

Удобным и наглядным средством для представления понятия «онтология» являются ментальные карты [Гаврилова и др., 2008]. Пример ментальной карты онтологий дан на рисунке 4.

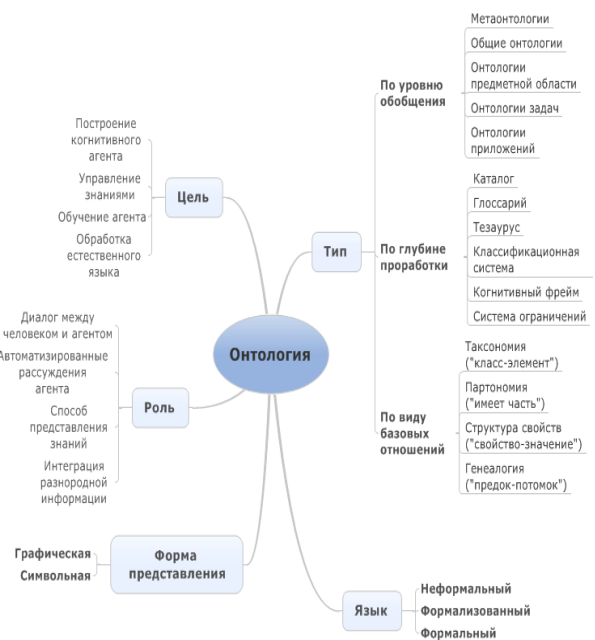


Рисунок 4 – Ментальная карта «Онтология когнитивного агента»

Из этого рисунка видно, что построение единственной понятной и согласованной предметной онтологии часто оказывается невозможным, поэтому на нижнем уровне наряду с предметной онтологией отдельно строятся онтологии задач и приложений, а на верхнем уровне – онтологии базовых категорий, встречающихся в разных предметных областях.

Согласно Дж.Сова [Sowa,1996], онтологии верхнего уровня описывают наиболее общие, парадигматические концептуализации систем, относительно независимые от задач предметной области и скорее характеризующие состояние профессионального сообщества. В отличие от этого онтологии нижнего уровня (онтология предметной области, онтология задач, онтология приложений) носят локальный характер и непосредственно зависят от типа и ролей агентов, для которых они используются.

Кроме того, выделяется метаонтология («онтология онтологий»), которая обеспечивает как точную, математическую спецификацию онтологий, так и формальный анализ их свойств. В частности, она включает методы и формы представления, интеграции и слияния различных онтологий. С помощью метаонтологии устанавливается соответствие между типом используемой информации (уровнем неопределенности) и выбираемым языком ее описания.

В работах [Смирнов и др., 2002; Калущкая и др., 2011] изложены варианты классификации и возможные схемы интеграции онтологий. Ниже рассмотрим модифицированную схему связей между онтологиями (рисунок 5), в которой главное место занимают гранулярные метаонтологии.

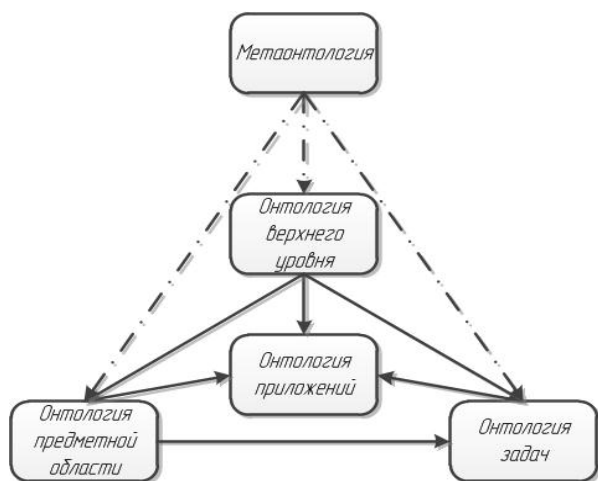


Рисунок 5 – Схема интеграции онтологий для интеллектуальных агентов

Из этой схемы видно, что на нижнем уровне онтология приложений получается путем интеграции онтологии предметной области и онтологии задач, онтология верхнего уровня обеспечивает инвариантные концептуально-реляционные конструкции для онтологий нижнего уровня, и все вышеперечисленные онтологии, а метаонтология позволяет выбирать адекватные модели и языки представления информации и знаний об онтологических категориях верхнего уровня (пространство, время, информация, и др.), а также языки описания предметных онтологий, онтологий задач и приложений. В следующем разделе будет обоснован выбор языков грануляции информации при построении онтологий для когнитивных агентов.

2.1. Гранулы, грануляция информации, гранулярные вычисления

Термин «гранула» происходит от латинского слова *granum*, что означает «зерно» и описывает мелкую частицу реального или идеального мира. Понятие «гранула» и термин «грануляция информации» ввел Л.Заде в 1979 г. [Zadeh, 1979]. Однако прямым предшественником теории грануляции может по праву считаться основатель мереологии Ст.Лесьневский.

Мереологией (партономией) называется учение о частях целого (теория частей и границ). Первоначально мереология понималась как вариант неклассической теории множеств. Как известно, в классической теории множеств используются два важных постулата – постулат принадлежности и постулат различимости элементов, – а также понятие пустого множества. В отличие от этого мереология: 1) делает акцент на целостности множества как «коллективного класса»; 2) основана на одном-единственном отношении «быть частью»; 3) обходится без пустого множества.

В настоящее время мереологию рассматривают, в первую очередь, как прототип «весомой» онтологии, которая опирается на следующие аксиомы.

1. Любой предмет есть часть самого себя (аксиома рефлексивности).

2. Две различные вещи не могут быть частями друг друга: если P – часть предмета Q , то Q не есть часть предмета P (аксиома антисимметричности).

3. Если P есть часть предмета Q , а Q – часть предмета R , то P есть часть предмета R (аксиома транзитивности).

Таким образом, отношение «часть-целое» есть отношение нестрогого порядка. Его построение является одним из главных способов грануляции информации.

Под *гранулой* (в смысле Л.Заде [Zadeh,1997]) понимается группа объектов, объединяемых отношениями неразличимости, эквивалентности, сходства, близости, т.е. отношениями, имеющими, по крайней мере, свойства симметричности и рефлексивности. В частности, информационные гранулы – это сложные единицы информации, которые образуются в процессе сжатия данных и извлечения знаний. По сути, термин «гранула» задает динамическую целостную информационную структуру, создаваемую когнитивным агентом для достижения своей цели. Типичные интерпретации гранул – это часть целого, подпроблема проблемы, разбиение, окрестность, кластер, множество с зоной неопределенности, обобщенное ограничение.

Известны различные классификации гранул: физические и концептуальные гранулы, четкие и нечеткие гранулы, одномерные и многомерные гранулы, гранулы данных и гранулы знаний, временные и пространственные (псевдофизические) гранулы и пр.

Простейшие примеры четких гранул приведены на рисунке 6 а,б,в, г, а нечетких гранул – на рисунке 6 д.

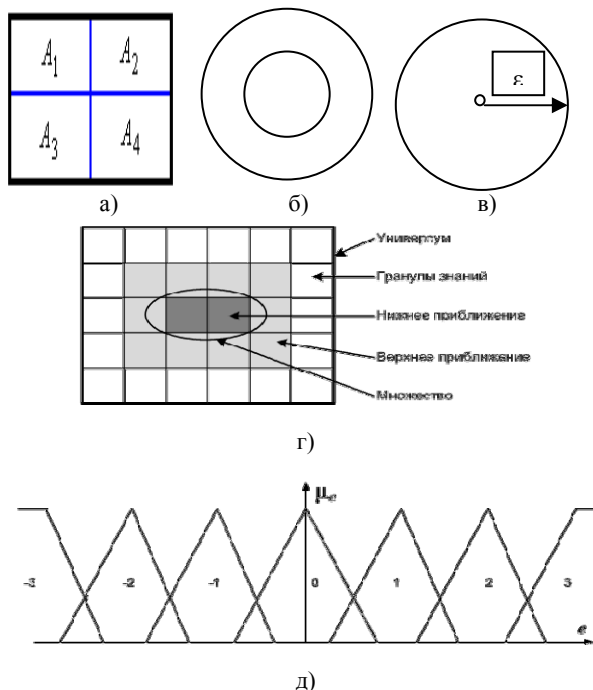


Рисунок 6 – Примеры четких и нечетких гранул:
 а) разбиение множества на подмножества;
 б) построение простой холархии (иерархии по вложенности);
 в) задание окрестности точки;
 г) представление приближенного множества;
 д) терм-множество лингвистической переменной, образующее когнитивный фрейм.

Грануляция информация выступает в качестве ключевой способности когнитивных агентов [Калуцкая и др., 2010]. Здесь следует выделить три важных положения.

1. Информационные гранулы играют ведущую роль в представлении и обработке знаний когнитивными агентами.
2. Уровень грануляции (размер гранул) имеет существенное значение для описания агентом проблемы и выбора стратегии ее решения.
3. Не существует универсального уровня информационной грануляции; размер гранулы является проблемно-ориентированным и зависящим от когнитивного агента.

Грануляцию информации можно осуществлять различными методами: а) на основе классификации и кластерного анализа; б) на базе мереологического подхода с помощью отношений вложенности и нестандартных множеств.

Отметим, что термин «Грануляция» охватывает процессы композиции (формирование более крупных гранул), и декомпозиции (формирование более мелких гранул). Гранулы отличаются друг от друга по своей природе, сложности, размеру, уровню абстрактности-детализации. Уровень грануляции можно задать как число объектов в грануле, поделенное на общее число гранул.

Одним из ведущих подходов к грануляции информации является построение обобщенных ограничений вида $X \text{ isr } C$, где X – переменная, C – гибкое ограничение на эту переменную, а значение r переменной связки isr определяет способ выражения (семантику) ограничения [Zadeh, 1997].

Общая схема грануляции информации когнитивным агентом имеет вид

$$G = \langle X, GR, M, R_{GR}, TR_v, TR_h \rangle$$

где X – проблемная область, GR – семейство информационных гранул, M – множество формальных методов грануляции, R_{GR} – множество соотношений между гранулами, TR_v – множество преобразований абстрактности-детализации гранул (вертикальных переходов между уровнями грануляции), TR_h – множество горизонтальных отображений гранул.

Гранулярные вычисления [Bargiela et al., 2003] – это новая концептуальная и компьютерная парадигма обработки информации, охватывающая различные методологии, теории, методы и программные средства, использующие гранулы при решении сложных задач. Ключевыми проблемами гранулярных вычислений являются построение, представление, интерпретация и использование гранул в соответствии с имеющимися знаниями. В частности, сам Л.Задэ связывает гранулярные вычисления с формированием, агрегированием и распространением обобщенных ограничений [Zadeh, 1997].

2.2. Определение и формальное представление гранулярных метаонтологий и онтологий

Когда говорят о метаонтологиях, речь идет непосредственно о моделях или языках формального представления онтологий (например, реляционные системы, семантические сети, деревья, ментальные карты, дескриптивные логики, унифицированный язык моделирования *UML*, язык веб-онтологий *OWL*, и пр.). В нашем случае метаонтология понимается как средство интеграции разнородных моделей представления знаний [Дюндюков и др., 2011; Калуцкая и др., 2011].

В общем случае следует выделять *сингулярные* и *гранулярные метаонтологии*. Сингулярная метаонтология задает либо один язык, либо семейство языков, ориентированных на работу с четкими, количественными данными.

Гранулярная метаонтология определяет конкретный набор взаимосвязанных моделей и языков представления информации, среди которых имеются средства, ориентированные на работу с качественной, неточной, нечеткой информацией.

Сам термин «метаонтология» имеет прямое отношение к наиболее универсальным, проблемно-независимым категориям, таким как *понятия, отношения, изменения*. Поэтому удобным средством формализации метаонтологий могут служить алгебраические системы А.И.Мальцева.

Определение 1. Формальная метаонтология есть тройка

$$MONT = \langle C, R, \Omega \rangle, \quad (1)$$

где C – множество понятий (объектов онтологии), R – множество отношений между понятиями, Ω – множество операций над понятиями и/ или отношениями.

Замечание. Следует отметить, что формальная спецификация онтологии или метаонтологии ранее

часто сводилась к реляционной системе, однако использование различных операций над понятиями и отношениями онтологии имеет очевидный практический смысл. Так в случае представления локальных онтологий нечеткими графами следует использовать операции пересечения, объединения, разности графов, как впрочем и специальные операции добавления/удаления вершин и дуг графа.

Определение 1*. Гранулярная метаонтология есть тройка

$$GMONT = \langle C_G, R_G, \Omega_G \rangle, \quad (1^*)$$

где C_G есть базовое гранулярное множество, понимаемое как основа онтологической грануляции R_G есть множество гранулярных отношений на C_G , Ω_G есть множество операций на C_G и/или R_G .

Ниже рассмотрим варианты задания C_G .

1. Гранулярное множество – универсум C вместе с фактор-множеством C/E , $C_{G1} = (C, C/E)$, где фактор-множество $C/E = \{[c]_E \mid c \in C\}$ индуцируется отношением эквивалентности E , $[c]_E = \{b \in C; cEb\}$ – класс эквивалентности, содержащий c .

Итак, в простейшем случае гранулярное множество формируется с помощью разбиения универсального множества C , а именно как $C = (C_1, \dots, C_n)$, где $C = C_1 \cup \dots \cup C_n$ (покрытие) и $C_i \cap C_j = \emptyset, \forall i, j = 1, \dots, n$.

2. Гранулярное множество как приближенное множество Липского-Павляка [Павляк, 1993].

Пусть C – множество понятий, а $E \subseteq C \times C$ есть отношение неразличимости (эквивалентности). Тогда пара $\wp = (C, E)$ образует пространство приближений. Классы эквивалентности по E называются элементарными множествами в пространстве приближений \wp , а любая совокупность элементарных множеств образует составное множество в пространстве \wp . Здесь $C_{G2} = (C, \wp)$. В результате любое подмножество $A \subseteq C$ аппроксимируется двумя приближениями: нижним приближением $\underline{E}C = \{c \mid [c]_E \subseteq A\}$ (наибольшее составное множество, содержащееся в A) и верхним приближением $\overline{E}C = \{c \mid [c]_E \cap A \neq \emptyset\}$ (наименьшее составное множество, содержащее A).

3. Гранулярное множество – универсум C вместе с семейством вложенных множеств (ensemble flou) F , $C_{G3} = (C, F)$, $F = \{A_0, \dots, A_n\}$, где $A_i \subseteq C, i=0, \dots, n$, $A_0 = X$, $A_0 \supseteq A_1 \supseteq \dots \supseteq A_n$ или в более общем случае как множество четких α -сечений, определенных на решетке L , $A_\alpha: L \rightarrow 2^X, \alpha \in L$.

4. Гранулярное множество – универсум C вместе с семейством нечетких множеств $[0,1]^C$ вида $\mu: C \rightarrow [0,1], C_{G4} = (C, [0,1]^C)$.

Существует немало вариантов грануляции информации с помощью нестандартных и гибридных нечетких множеств (см. [Тарасов, 2011]).

Ниже остановимся на онтологиях, построенных на основе нечетких и лингвистических гранул.

2.3. Гранулярные онтологии пространства

Среди онтологий верхнего уровня особое место занимают пространственные онтологии, тесно связанные с морфологическим анализом. Термин

«морфология системы» охватывает ее пространственную и функциональную организацию. Пространственные структуры образуются такими отношениями как: меререологические («часть-целое»), расстояние («близко–далеко»), окрестность, направление («вперед–назад»), размер («малый–большой») в физическом пространстве, а также отношениями типа «сходство–различие» в абстрактном пространстве свойств.

Для построения онтологии пространства здесь берется подход Лейбница. В отличие от Ньютона, предложившего теорию «пустого» пространства, Лейбниц ввел реляционную концепцию пространства, согласно которой пространство связывается с порядком взаимного расположения и сосуществования в нем различных тел. По Лейбницу, пространство представляется неявно, через отношения между объектами. Обычно в нем определяют некоторую метрику или топологию, чтобы оценивать размеры объектов и расстояния между ними.

В зависимости от выбора пространственных примитивов имеем два типа моделей пространственных объектов: а) точные, сингулярные модели, где пространственные объекты мыслятся (явно или неявно) как множества точек; б) приближенные, гранулярные модели, например, модели, связанные с интервалами и отношениями между ними.

Типичными примерами гранулярных моделей пространства и времени являются обычные и расширенные алгебры Аллена

В последние годы построение онтологий пространства часто идет по линии интеграции подходов меререологии и топологии: Меререология + Топология = Меререотопология. Здесь особую популярность приобретают меререотопологические «бесточечные» модели пространства, основанные на примитивах-областях и трех главных отношениях между ними – геометрических («конгруэнтность»), меререологических («быть частью») и топологических («связность»).

Конгруэнтность позволяет определить отношение сходства между областями. В геометрии две фигуры называются конгруэнтными, если одну из них можно перевести в другую с помощью движения. Понятие связности есть математическое выражение интуитивного представления о целостности геометрических фигур. Отношение связности в топологии рефлексивно, симметрично и монотонно.

Полезный подход к построению гранулярных онтологий пространства связан с обобщенными ограничениями вида $X \text{ is } r C$, где X – переменная, C – гибкое ограничение на эту переменную, а значение r переменной связки $\text{is } r$ определяет способ выражения (семантику) ограничения [Zadeh, 1997]. В частности, речь идет о лингвистических ограничениях на *расстояния* между объектами a и b , например, «Расстояние между a и b – очень близкое», а также о лингвистических ограничениях на их *взаимное положение*, например, «объект b находится впереди и немного левее объекта a ».

Наборы значений лингвистических переменных Расстояние и Положение образуют когнитивные протофреймы, для которых можно определить соответствующие экзофреймы с помощью семейств означенных, нормальных нечетких множеств $\Phi = \{A_1, \dots, A_n\}$, где любые два соседних A_i и A_j имеют область перекрытия (рисунок 6 д).

3. Нечеткие и лингвистические онтологии

Переход от обычных онтологий к нечетким и лингвистическим онтологиям выглядит вполне естественным, поскольку понятия и отношения естественного языка, представляющие собой исходный материал для построения онтологии неоднозначны, неточны и не имеют жестких границ. В то же время при построении графа онтологии в условиях коллектива экспертов при наличии противоречивых мнений даже бинарные экспертные оценки типа «да-нет» приводят к взвешиванию вершин и дуг соответствующего графа. Поэтому адекватным средством формализации онтологий могут служить модели на базе лингвистических переменных, нечеткие множества, нечеткие отношения, нечеткие графы и нечеткие деревья, нечеткие ограничения, нечеткие реляционные и алгебраические системы (см. [НГС, 2007]).

3.1. Краткий обзор нечетких онтологий

Понятие нечеткой онтологии не является новым, но большинство определений и подходов в этой области появились сравнительно недавно, в середине или в конце 2000-х годов. Как правило, строятся нечеткие онтологии нижнего уровня, где вводится минимальное расширение, достаточное для конкретного приложения (например, либо нечеткие отношения, либо, реже, нечеткие понятия и их нечеткие атрибуты). Типичным примером является определение легкой нечеткой онтологии в виде $FONT_{DA} = \langle C, R_F \rangle$, где C – множество понятий, R_F – множество нечетких отношений [Dey et al., 2008]. Из этого определения видно, что нечеткую онтологию можно представить обычным нечетким графом с взвешенными дугами. В свою очередь, минималистское определение весомой нечеткой онтологии может иметь вид $FONT_{QHFC} = \langle C, AT_F, R, AX \rangle$, где C – множество понятий, AT_F – семейство множеств нечетких атрибутов (каждое понятие может описываться своим множеством атрибутов), R – множество отношений между понятиями и AX – множество аксиом [Quan et al., 2006].

С целью явного представления весомой нечеткой онтологии в виде нечеткого дерева на множестве понятий определяют иерархию, т.е. $FONT_{SY} = \langle C, R_F, H, AX \rangle$, где C – множество понятий (возможно с нечеткими атрибутами), R_F – множество нечетких отношений, H – иерархия, AX – множество аксиом [Sanchez et al., 2006].

В [Lee et al., 2005] под нечеткой онтологией понимается расширенная онтология проблемной области с нечеткими понятиями и нечеткими отношениями. Более интересным представляется

тотальное нечеткое расширение обычной онтологии с введением множества нечетких понятий C_F , семейств множеств нечетких атрибутов (свойств) понятий AT_F и множества нечетких бинарных отношений R_F , а также множества индивидов I . Тогда получаем $FONT_{CL} = \langle C_F, R_F^2, AT_F, I \rangle$ [Cai et al., 2008].

Некоторые определения нечеткой онтологии тесно связаны с конкретными формализмами или языками программирования. Так в [Stoilos et al., 2005] предложено определение нечеткой онтологии как подмножества аксиом языка «Fuzzy OWL».

В контексте развития нечетких гранулярных подходов в инженерии онтологий особое внимание привлекают работы [Calegari et al., 2006; Calegari et al., 2007], в которых нечеткая онтология задается четверкой $FONT_{CC} = \langle C_F, R_F, R_{Fc}, AX \rangle$ или пятеркой $FONT_{CC} = \langle I, C_F, R_F, R_{Fc}, AX \rangle$, где I – множество индивидов (агентов), C_F – множество нечетких понятий, R_F – множество базовых нечетких n -арных отношений (включая нечеткие таксономические отношения), R_{Fc} – множество конкретных нечетких отношений проблемной области, AX – множество аксиом.

Ниже нами предлагается вариант построения полностью нечетких онтологий на основе нечетких алгебраических систем.

Определение 2. Полностью нечеткая онтология есть пятерка

$$FONT_{TKC} = \langle I, C_F, R_F^k, \Omega_F^j, AX \rangle, \quad (2)$$

где I – множество индивидов (агентов), C_F – множество нечетких понятий, R_F^k – семейство множеств нечетких отношений, $k = 1, 2, \dots, s$; Ω_F^j – множество конечных операций над нечеткими понятиями и/или нечеткими отношениями, $j = 0, \dots, n$, AX – множество (возможно нечетких) аксиом.

Примеры нечеткой аксиоматизации изложены в [Turksen, 2007].

3.2. Методика построения нечеткой онтологии

Ниже представлена методика построения нечетких онтологий. В ее основе лежит идея построения конкретной иерархической структуры онтологии группой экспертов (агентов)

$$FONT_{KT} = \langle I, C_F, H, R_F \rangle, \quad (3)$$

где I – множество индивидов (агентов), C_F – множество нечетких понятий, H – иерархическая модель, R_F – семейство множеств нечетких отношений.

Это формальное представление онтологии будет использовано при создании методики построения нечетких онтологий на основе пятиуровневой онтологической модели (рисунок 7).

Здесь на первом (верхнем) уровне расположена комплексная проблемная область D , которая предполагает слияние источников информации (экспертов, коллекций текстов) из разных областей. Эта проблемная область (ПрО) разбивается на соответствующие подобласти SD (второй уровень), причем каждая подобласть характеризуется своими источниками информации (третий уровень). С их помощью строится иерархия понятий ПрО, где на

четвертом уровне находятся базовые понятия или категории онтологии $C_i, i = 1; \dots, n$, а на пятом (нижнем) уровне – ключевые слова, относящиеся к именам категорий $k_j, j = 1; \dots, m$.

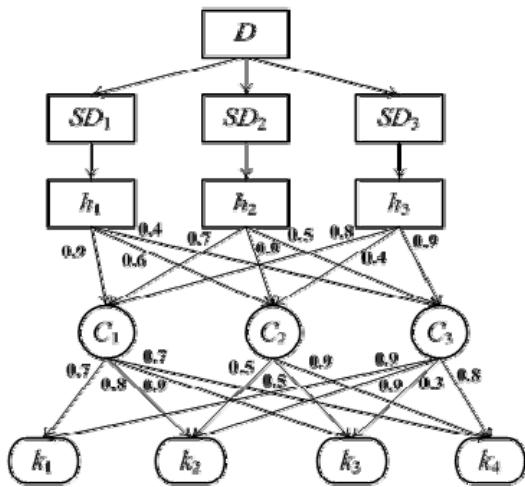


Рисунок 7 –Пример нечеткой онтологии

Рассмотрим иллюстративный пример нечеткой реляционной онтологии «Интеллектуальная среда» (Ambient Intelligence или Smart Environment) для железнодорожного транспорта которая трактуется как коллективный мета-агент. Здесь проблемная область распадается на три главных подобласти «Распределенные системы», «Исполнительные устройства», «Интеллектуальное ядро». Базовыми понятиями онтологии являются « C_1 – «мета-агент», C_2 – «восприятие», C_3 – «действие» вместе с четырьмя ключевыми словами (k_1 – «делиберативный агент», k_2 – «когнитивный агент», k_3 – «коммуникативный агент», k_4 – «ресурсный агент»).

Таким образом, ядро нечеткой реляционной онтологии задается нечетким отношением R , определенном на декартовом произведении $k \times C$ ключевых слов и категорий.

$$R = \begin{matrix} & C_1 & C_2 & \dots & C_n \\ \begin{matrix} k_1 \\ k_2 \\ \dots \\ k_m \end{matrix} & \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \dots & \dots & \dots & \dots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix} \end{matrix} \quad (4)$$

Рассмотрим в зависимости от выбора источников информации два возможных сценария построения онтологии. Первый сценарий связан с созданием автономной междисциплинарной рабочей группы экспертов $h_l, l=1, \dots, p$, которые вначале формируют термины и терминологические словосочетания проблемной области, а затем составляют простую иерархию «категории – ключевые слова». При разработке онтологической системы эксперты производят выбор и соотнесение имен категорий с ключевыми словами. Поскольку каждый эксперт есть специалист в определенной

подобласти ПрО, а уровень его компетентности в других подобластях ниже, предварительно определяется вес эксперта по разделам (подобластям) ПрО, выражаемый числом из интервала $[0, 1]$. Далее эксперты определяют степень связи между категорией C_i и ключевым словом k_j , которая также задается числом из интервала $[0, 1]$. Сближение значений весов может характеризовать «степень разделения» онтологии разными специалистами.

Нечеткую онтологию можно также получить на основе четких онтологий, построенных разными экспертами. А именно, нечеткие значения связей в онтологии (рисунок 7) можно легко получить, исходя из четких экспертных оценок в ситуации групповой экспертизы. Пусть, например, имеется 10 экспертов, которые оценивают наличие связи категории C_i с ключевым словом k_j . Предположим, семь экспертов указали, что связь есть, а трое – что связи нет. Тогда имеем степень связи $r_{ij} = 0,7$. В общем случае значение степени связи определяется формулой $r_{ij} = m/n$, где m – число экспертов, указавших, что связь между C_i и k_j есть, а n – общее число экспертов. Если необходимо сравнить между собой различные онтологии, то можно использовать показатели сходства графов.

Второй сценарий связан с формированием (возможно с помощью экспертов) представительной коллекции текстовых документов (учебники, монографии, научные статьи, техническая документация, материалы с сайтов и пр.). Сборка терминологических словосочетаний из документов производится автоматически с использованием синтаксических признаков. Для выделения словосочетаний укажем их лексико-грамматические типы. Известны различные лексико-грамматические типы словосочетаний: глагольные, именные, наречные [Валгина, 2000].

Глагольные словосочетания имеют следующие варианты:

- глагол + существительное или местоимение (с предлогом или без предлога): приближаться к станции, определить местоположение;
- глагол + инфинитив или деепричастие: требует остановиться, двигаться ускоряясь;
- глагол + наречие: повернуть направо, запросить повторно.

Именные словосочетания делятся на субстантивные, адекативные, с главным словом числительным и с главным словом местоимением [Валгина, 2000]. Остановимся на субстантивных словосочетаниях:

- одиночные существительные, аббревиатуры: агент, мета-агент, когнитивный агент (КА);
- согласуемое слово + существительное: мобильный агент;
- существительное + существительное: взаимодействие со средой;
- существительное + инфинитив: команда остановиться.

Также будем использовать модели адекативных словосочетаний:

- прилагательное + существительное: коллективное восприятие;
- прилагательное + прилагательное + существительное: физический когнитивный агент;
- прилагательное + инфинитив: способный общаться, готовый действовать.

После построения текстовой коллекции определяются веса связей, например, по частотному признаку встречаемости ключевых слов и словосочетаний в соответствующих текстовых документах.

Алгоритм формирования нечеткой онтологии представлен на рисунке 8.

3.3. Лингвистические онтологии

В отличие от значительного числа зарубежных работ по нечетким онтологиям, публикации, связанные с формальными лингвистическими онтологиями остаются достаточно редкими (см. [Zhai et al., 2008]). В принципе, построение нечеткой лингвистической онтологии может опираться на понятие лингвистической переменной (ЛП) Л.Заде, значениями которой являются термы или понятия естественного языка. Напомним соответствующее определение

Определение 3 [Заде, 1976]. Лингвистической переменной называется пятерка

$$LV = \langle L, T, U, G, M \rangle, \quad (5)$$

где L – название лингвистической переменной, T – ее терм-множество (совокупность лингвистических значений), U – универсальное множество числовых значений, G – множество синтаксических правил (грамматика), служащее для образования составных термов из простых, M – множество семантических правил, ставящих в соответствие каждому терму из T нечеткое множество в U .

Это определение Л.Заде не совсем подходит для построения лингвистических онтологий, поскольку в нем отсутствуют отношения между объектами из множеств T и U . Дадим расширение определения (5), в котором явно учитываются эти отношения.

Определение 3*. Расширенная лингвистическая переменная есть набор

$$LV_{ex} = \langle L, T, U, G, M, R_T, R_U, O_g, TR_U \rangle, \quad (5^*)$$

где сохранены обозначения формулы (5), а также определены R_T и R_U – множество отношений на T и U соответственно, O_g – множество операций грануляции, TR_U – множество преобразований универсума.

Тогда онтологию на базе ЛП можно задать следующим образом.

Определение 4. Онтология на основе расширенной лингвистической переменной есть набор

$$LVONT = \langle I, C_A, C_F, R, U, [0,1]^U, R_F \rangle, \quad (6)$$

где I – множество индивидов (агентов), $C_A = \{c_A\}$ – абстрактное понятие (синглетон), соответствующее имени лингвистической переменной, C_F – множество нечетких понятий (терм-множество ЛП), $R = \{r \mid r \subseteq C_F \times C_F\}$ – множество бинарных отношений между понятиями. Здесь главное место занимает отношение

строого порядка $<$. Пара $\langle C_F, < \rangle$ порождает упорядоченную структуру. U есть (количественный) универсум, а $[0,1]^U$ – множество всех нечетких подмножеств на U . R_F обозначает множество отношений на $[0,1]^U$.

Формулу (6) можно обобщить на случай составной лингвистической переменной или иерархии лингвистических переменных.

ЗАКЛЮЧЕНИЕ

В настоящей статье предложены варианты расширения понятия онтологии с целью учета неоднозначности и нечеткости естественно-языковых понятий, отношений, ограничений, связанных с нею понятий. Введено общее представление метаонтологии как алгебраической системы. Разработаны гранулярные, нечеткие и лингвистические онтологии для обеспечения диалога и взаимопонимания когнитивных агентов. Описана предложенная авторами методика построения нечетких онтологий, которая предусматривает два возможных сценария в зависимости от выбора источников информации: группы экспертов или коллекции текстов.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проекты №11-07-00738 и №11-07-13165-офи-м-2011-РЖД).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Валгина, 2000] Валгина Н.С. Синтаксис современного русского языка. Учебник. – М.: Агар, 2000.

[Гаврилова и др., 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.

[Дюндюков и др., 2011] Дюндюков, В.С. Онтология ресурсов: теоретический анализ и приложения/ В.С.Дюндюков, А.П.Калуцкая, М.Н.Святкина//Сборник трудов 2-й Всероссийской конференции «Инженерия знаний и технологии семантического веба-2010» (KESW-2011, Санкт-Петербург, НИУ ИТМО, 7-9 ноября 2011 г.). – СПб: НИУ ИТМО, 2011. – С. 137-149.

[Заде, 1976] Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений: Пер. с англ. – М.: Мир, 1976.

[Калуцкая и др., 2010] Калуцкая А.П. Информационные гранулы и методы их построения: применение при разработке интеллектуальных агентов/ А.П.Калуцкая, В.Б.Тарасов // Интеллектуальный анализ информации. Сборник трудов X-й международной научной конференции им. Т.А.Таран (ИАИ-2010, Киев, 18-21 мая 2010 г.). – Киев: Прогресс, 2010. – С.291-297.

[Калуцкая и др., 2011] Калуцкая А.П. Гранулярные метаонтологии и онтологии пространства/ А.П.Калуцкая, В.Б.Тарасов// Реинжиниринг бизнес-процессов на основе современных информационных технологий. Системы управления знаниями. Сборник научных трудов XIV-й научно-практической конференции (РБП-СУЗ-2011, Москва, МЭСИ, 28-29 апреля 2011 г.). – М.: МЭСИ, 2011. – С. 136-145.

[НГС, 2007] Нечеткие гибридные системы/ И.З.Батыршин, А.О.Недосекин, А.А.Стецко, В.Б.Тарасов, А.В.Язенин, Н.Г.Ярушкіна. – М.: Физматлит, 2007.

[Павляк, 1993] Павляк З. Приближенные множества – основные понятия/ З.Павляк// Логические исследования. Вып.1. – М.: Наука, 1993. – С.6-19.

[Плесневич, 2011] Плесневич, Г.С. Анализ простых онтологий / Г.С.Плесневич// Интеллектуальные системы. Коллективная монография/ Под ред. В.М. Курейчика. – М.: Физматлит, 2010. – С.206-221/.

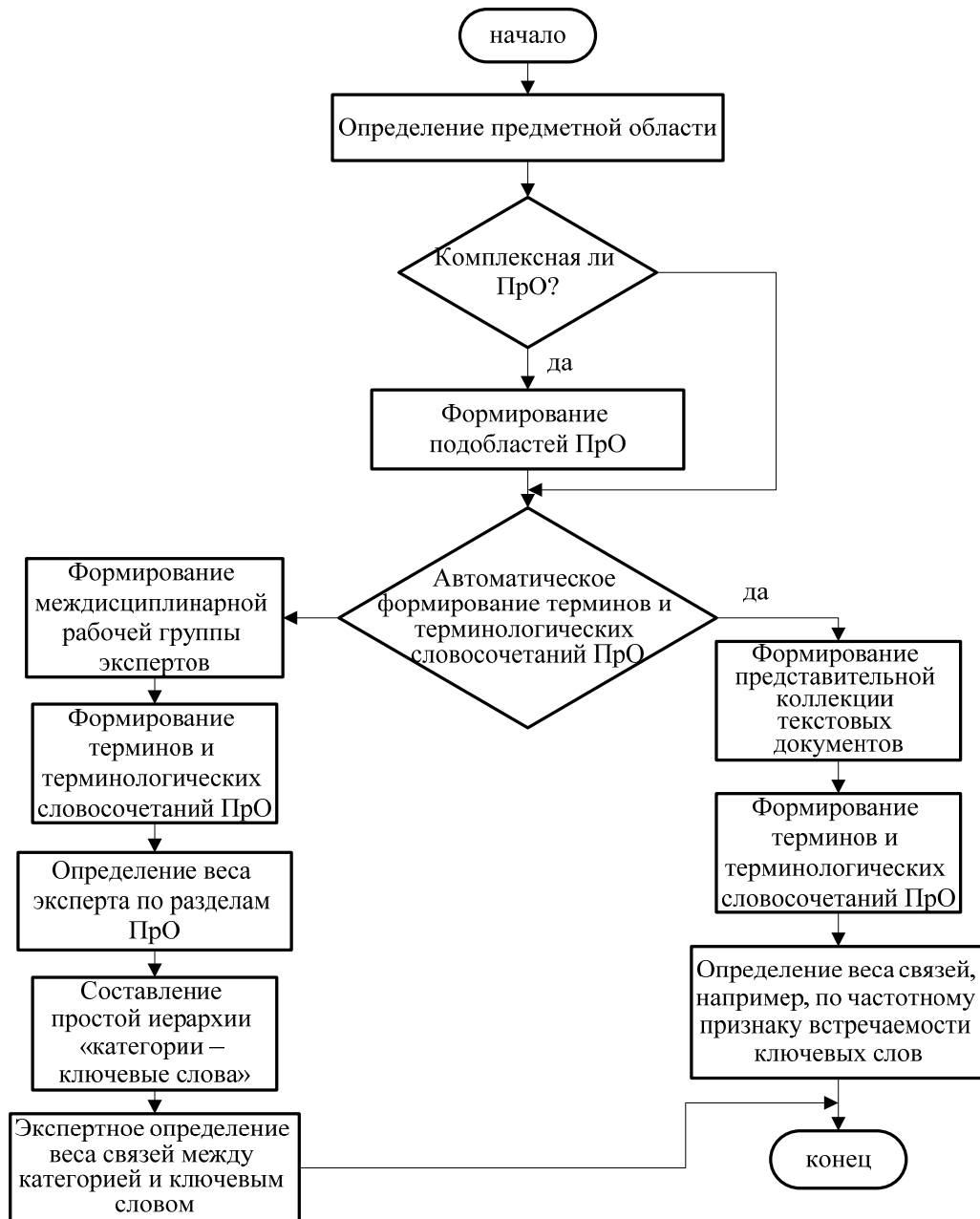


Рисунок 8 – Алгоритм построения нечетких онтологий

[Попов, 2004] Попов, Э.В. Общение с ЭВМ на естественном языке. 2-е изд. – М.: Едиториал УРСС, 2004.

[Смирнов и др., 2002] Смирнов, А.В. Онтологии в системах искусственного интеллекта: способы построения и организации (часть 1) / А.В.Смирнов, М.П.Пашкин, Н.Г.Шилов, Т.В.Левашова // Новости искусственного интеллекта. – 2002. – №1. – С.3-13.

[Тарасов, 2002] Тарасов, В.Б. От многоагентных систем к интеллектуальным организациям. – М.: Едиториал УРСС, 2002.

[Тарасов, 2004] Тарасов, В.Б. НЕ-факторы: от семиотического анализа к методам формализации / В.Б.Тарасов // Новости искусственного интеллекта. – 2004. – №2. – С. 95-114.

[Тарасов, 2011] Тарасов, В.Б. Грануляция информации, нестандартные и гибридные нечеткие множества / В.Б.Тарасов // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов VI-й Международной научно-практической конференции (Коломна, 16-19 мая 2011 г.). – М.: Физматлит, 2011. – Т.1. – С.35-49.

[Ющенко, 2009] Ющенко, А.С. Диалоговое управление роботами с использованием нечетких моделей / А.С.Ющенко // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов V-й Международной научно-практической конференции (Коломна, 28-30 мая 2009 г.). – М.:

Физматлит, 2009. – Т.1. – С. 97-108.

[Bargiela et al., 2003] Bargiela, A., Pedrycz, W. Granular Computing: an Introduction. – Dordrecht: Kluwer Academic Publishers, 2003.

[Cai et al., 2008] Cai, Y. A Formal Model of Fuzzy Ontology with Property Hierarchy and Object Membership / Y.Cai, H F. Leung / Conceptual Modeling. Lecture Notes in Computer Science. Vol.5231. Ed. by Q.Li et al. – Berlin: Springer-Verlag, 2008. – P.69-82.

[Calegari et al., 2006] Calegari, S. Towards a Fuzzy Ontology Definition and a Fuzzy Extension of Ontology Editor / S.Calegari, D.Ciucci D // 8th International Conference on Enterprise Information Systems (ICEIS-2006, Paphos, Cyprus, 2006). Selected Papers. Lecture Notes in Business Information Processing. Vol.3 / Ed. by Y.Manolopoulos et al. – Berlin: Springer-Verlag, 2006. – P. 147-158.

[Calegari et al., 2007] Calegari, S. Fuzzy Ontologies and Scale-Free Network Analysis / S.Calegari, F.Farina // International Journal of Computer Science and Applications. – 2007. – Vol.IV. – P. 125-144.

[Dey, 2008] Dey, L. Fuzzy Ontologies for Handling Uncertainties and Inconsistencies in Domain Knowledge Description / L.Dey, M.Abulaish // Proceedings of the 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'2008, Hong Kong, China). – IEEE Computer Society, 2008, p.1366-1373.

[Gruber, 1993] Gruber, T.R. A Translation Approach to Portable Ontologies/ T.R.Gruber// Knowledge Acquisition. – 1993. –Vol.5, №2. – P. 199-220.

[Guarino, 1995] Guarino, N. Formal Ontology, Conceptual Analysis and Knowledge Representation/ N.Guarino// International Journal of Human-Computer Studies. – 1995. – Vol.43. – №5-6. – P. 625-640.

[Lee et al., 2005] Lee, Ch.-S. A Fuzzy Ontology and its Application to News Summarization/ Ch.-S. Lee, Zh.-W. Jian, L.-K. Huang// IEEE Transactions on Systems, Man and Cybernetics. – 2005. – Vol.35, №5. – P.859-880.

[Quan et al., 2006] Quan, T.T. Automatic Fuzzy Ontology Generation for Semantic Web/ T.T.Quan, S.Ch.Hui, A.Ch.M. Fong, T.H.Cao// IEEE Transactions on Knowledge and Data Engineering. – 2006. – Vol.18, №6. – P.842-856.

[Sanchez et al., 2006] Sanchez, E. Fuzzy Ontologies for the Semantic Web/ E.Sanchez, T.Yamanoi// Proceedings of the 7th International Conference on Flexible Query Answering Systems (FQAS'2006)/ Ed. by H.L.Larsen et al. Lecture Notes in Computer Science. Vol.4027. – Berlin: Springer-Verlag, 2006. – P.691-699.

[Sowa, 1995] Sowa, J.F. Top-Level Ontological Categories/ J.F.Sowa// International Journal of Human-Computer Studies. –1995. – Vol.43, №5-6. – P. 669–685.

[Stoilos et al., 2005] Stoilos, G. Fuzzy OWL: Uncertainty and the Semantic Web/ G.Stoilos, G.Stamou, V.Tsouvaras, J.Z.Pan, I.Horrocks // Proceedings of the 1st International Workshop on OWL: Experience and Directions (OWLED'2005, Gailway, Ireland), 2005.

[Turksen,2006] Turksen,I.B.An Ontological and Epistemological Perspective of Fuzzy Set Theory. – Amsterdam: Elsevier Science Publishers, 2006.

[Wooldridge, 2002] Wooldridge, M. An Introduction to Multi-Agent Systems. – New York: Wiley and Sons, 2002.

[Wooldridge et al., 1995] Wooldridge, M. Intelligent Agents: Theory and Practice/ M.Wooldridge, N.Jennings// The Knowledge Engineering Review. – 1995. – Vol.10, №2. – P. 115-152.

[Zadeh, 1979] Zadeh, L.A. Fuzzy Sets and Information Granularity/ L.A.Zadeh// Advances in Fuzzy Sets Theory and Applications/ Ed. by M.M.Gupta, R.K.Ragade, R.R.Yager.– Amsterdam: North-Holland Publishing Company, 1979. – P.3-20.

[Zadeh, 1997] Zadeh, L.A. Toward a Theory of Fuzzy Information Granulation and Its Centrality in Human Reasoning and Fuzzy Logic/ L.A.Zadeh // Fuzzy Sets and Systems. – 1997. – Vol.90. – P.111-127.

[Zhai et al., 2010] Zhai J. Linguistic Variable Ontology and Its Application to Fuzzy Semantic Retrieval/ J.Zhai, V.Li, K.Zhou// Communication in Computer and Information Science. – 2010. – Vol.106, Part 4. – P.188-195.

GRANULAR, FUZZY AND LINGUISTIC ONTOLOGIES TO ENABLE MUTUAL UNDERSTANDING BETWEEN COGNITIVE AGENTS

Tarasov V.B., Kalutskaya A.P., Svyatkina M.N.

*Bauman Moscow State Technical University,
Moscow, Russia*

Tarasov@rk9.bmstu.ru

k_a_p@rbcmail.ru

maria.svyatkina@gmail.com

A system of ontologies for cognitive agents is considered. The concept of granular meta-ontology is introduced and formalized as algebraic system. Some ways of representing granular sets are suggested. Definitions of fuzzy ontologies as a kind of granular ontologies are given. The methodology and algorithm of constructing fuzzy ontology to deal with a complex heterogeneous problem domain are developed. A model of linguistic ontologies based on extended linguistic variable is proposed to provide an effective communication between cognitive agents.

Keywords: cognitive agent, agents interaction, ontology, fuzzy ontology, granule, granular meta-ontology, agents co-operation, mutual understanding

INTRODUCTION

Basic problems of constructing a system of ontologies for cognitive agents team are considered. Primarily some necessary information from agents theory is given, the concept of cognitive agent and his basic features are detailed, the peculiarities of informational interaction between natural and artificial cognitive agents in course of dialogue control are discussed. A general picture of ontological approach to agents communication engineering and MAS development is drawn by using mind map. Some difficulties of building a unique ontology are shown on the basis of this map and a system of ontologies for cognitive agents is proposed.

The concepts of granular meta-ontology and granular ontologies play a leading part in this paper. To justify the use of granularity in ontologies we give the fundamentals of information granulation by cognitive agents, granular computing and computing with words based on generalized constraints. A way of formalizing granular meta-ontologies by using algebraic systems is proposed. A short review of existing definitions for fuzzy ontologies is presented and two new definitions of fuzzy ontologies are suggested. These definitions underlay authors's methodology of constructing fuzzy ontology. The concept of fuzzy linguistic ontology based on an extension of Zadeh's linguistic variable is introduced.

MAIN PART

The main feature of cognitive agent is the construction of internal modal of his environment. Moreover cognitive agents are able to efficiently work in the presence of such factors as local environment perception, imprecision and incompleteness of its internal representation, vagueness and changes in beliefs, fuzziness and contradictoriness of goals and intentions, decision-making under uncertainty, fragmented planning, incompleteness, inaccuracy and unreliability of information issued from other agents, and so on.

To overcome these difficulties cognitive agent obtains the information from various sources and performs information granulation. Here main information sources for artificial cognitive agent are: 1) human agent; 2) proper database/ knowledge base; 3) artificial sensors. Both to establish a dialogue with human agent and discover knowledge from sensor data artificial cognitive agent ought to form granules.

Information granulation as a basic capacity of cognitive agent consists in processing information on such level of abstraction which is consistent with the allowable level of imprecision. The term «granule» is originated from Latin word *granum*, that means grain, to denote a small particle in the real or imaginary world. According to L.Zadeh, granule is seen as a collection of objects which are drawn together by equivalence,

indistinguishability, similarity or functionality. Typical interpretations of granules are: part of the whole, sub-problem of the problem, cluster, variable constraints. Information granules are complex dynamic information entities which are formed to achieve some goal. By selecting different levels of granulation one can obtain different levels of knowledge.

Granular computing is an emergent conceptual and computational paradigm for information processing and knowledge representation to develop integrated, hybrid and synergetic AI Systems. Granular computing is often used as an umbrella term to cover any methodologies, theories, techniques and tools that make use of granules in complex problem solving.

The system of ontologies to provide cognitive agents mutual understanding and joint actions includes domain ontology, task ontology, application ontology top-level ontology and meta-ontology. Here the concept of meta-ontology is of primary concern. Meta-ontology enables both mathematical specification of ontology and formal analysis of its properties. In particular, it includes techniques and tools of representing, integrating and merging various ontologies.

The term meta-ontology is tightly related to some universal, problem-independent categories such as concept, relation, changes. Therefore, a natural formalism to generally describe meta-ontology is Maltsev's algebraic system. It is worth noticing that formal specification of meta-ontology (ontology) by relational system is not sufficient, the use of various operations over ontology concepts and relations allows to introduce dynamics. For instance, in case of representing ontologies by fuzzy graphs the intersection, union, difference operations, as well as special vertex/arc addition or elimination operations have to be used.

Below the definition of granular algebraic system is suggested.

Definition 1*. Granular meta-ontology is given by a triple

$$GMONT = \langle C_G, R_G, \Omega_G \rangle, \quad (1^*)$$

where C_G is a basic granular set seen as a kernel of ontological granulation, R_G is a set of granular relations on C_G , Ω_G is a set of operations on C_G . and/ or R_G .

The following ways of giving concepts granular sets are considered: 1) universal set C with a quotient set C/E , denoted by $C_{G1}=(C,C/E)$, where E is an equivalence relation; 2) universal set C with rough set given by lower and upper approximation; 3) universal set C with a family of nested sets $F = \{A_0, \dots, A_n\}$, $C_{G3} = (C, F)$, $F = \{A_0, \dots, A_n\}$, where $A_i \subseteq C$, $i=0, \dots, n$, $A_0 = X$, $A_0 \supseteq A_1 \supseteq \dots \supseteq A_n$ or more generally as a set of α -cuts defined on the L , $A_\alpha: L \rightarrow 2^X$, $\alpha \in L$; 4) universal set C with a family of fuzzy sets $[0,1]^C$, $C_{G4} = (C, [0,1]^C)$.

A special case of granular ontology is fuzzy ontology, where fuzzy concepts and/or fuzzy relations and/or fuzzy attributes are considered. Two versions of fuzzy ontologies – heavy-weight and light-weight-ones are suggested below.

Definition 2. Completely fuzzy ontology is a quintuple

$$FONT_{TKC} = \langle I, C_F, R_F^k, \Omega_F^j, AX \rangle, \quad (2)$$

where I is the set of individuals (agents), C_F is the set

of fuzzy concepts, R_F^k is the family of fuzzy relations sets, $k = 1, 2, \dots, s$; Ω_F^j is the set of finite operations over fuzzy concepts and/or fuzzy relations, $j=0, \dots, n$, AX is the set of axioms.

Definition 3. Fuzzy ontology is a quadruple

$$FONT_{KT} = \langle I, C_F, H, R_F \rangle, \quad (3)$$

where I is the set of individuals (agents), C_F is the set of fuzzy concepts, H is the hierarchy, R_F is the family of fuzzy relations sets.

This formal representation is used to create a methodology of constructing fuzzy ontology on the basis of five-layer model.

The first top-layer corresponds to a complex problem domain D that supposes merging of information sources (experts, text collections) from various areas. So problem domain is decomposed into sub-domains SD (the second layer), where each SD_l , $l = 1, \dots, u$ has its own information sources (the third layer). They enable the generation of basic concepts hierarchy, where main ontology categories C_i , $i = 1, \dots, n$ are situated on the fourth layer, and related keywords k_j , $j = 1, \dots, m$ are placed on the fifth layer.

Finally, fuzzy linguistic ontology is introduced on the basis of extended linguistic variable.

Definition 3*. An extended linguistic variable is given by a tuple

$$LV_{ex} = \langle L, T, U, G, M, R_T, R_U, O_g, TR_U \rangle, \quad (5^*)$$

where L is the name of linguistic variable, T is its term set, U is the universal set (numerical scale), G is the set of syntactic rules (grammar), M is the set of semantic rules, R_T is the set of relations on T , R_U is the set of relations on U , O_g is the set of granulation operations, TR_U is the set of universe transformations.

Definition 4. A fuzzy linguistic ontology based on extended linguistic variable is a tuple

$$LVONT = \langle I, C_A, C_F, R, U, [0,1]^U, R_F \rangle, \quad (6)$$

where I is the set of individuals (agents), $C_A = \{c_A\}$ is an abstract concept (singleton) that corresponds to the name of linguistic variable, C_F is the set of fuzzy concepts (the term set of linguistic variable), $R = \{r \mid r \subseteq C_F \times C_F\}$ is the set of binary relations between fuzzy concepts. Here the strict order relation $<$ is of special concern. The pair $\langle C_F, < \rangle$ generates an ordered structure. Here U is the universal set, $[0,1]^U$ is the set of fuzzy subsets on U , R_F is the set of fuzzy relations on $[0,1]^U$.

CONCLUSION

Some extensions of ontology formalism are proposed in order to take into account imprecision and fuzziness of natural-language concepts, relations, constraints. A general representation of meta-ontology as algebraic system is introduced to provide a suitable background for dynamic ontology specification. Granular, fuzzy and linguistic ontologies are developed to ensure dialogue and mutual understanding of cognitive agents. The methodology of constructing fuzzy ontologies is suggested that supposes two possible scenarios depending on the selection of information source: expert team or texts collection.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8

МИВАРНЫЕ ТЕХНОЛОГИИ СОЗДАНИЯ ЛОГИЧЕСКОГО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА: ЛОГИЧЕСКАЯ ОБРАБОТКА С ЛИНЕЙНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТЬЮ БОЛЕЕ 3 МИЛЛИОНОВ ПРОДУКЦИЙ И ВОЗМОЖНОСТЬ ПОНИМАНИЯ СМЫСЛА ЧЕРЕЗ МОДЕЛИРОВАНИЕ БОЛЬШОГО КОНТЕКСТА

Варламов О.О.

*Московский автомобильно-дорожный государственный технический университет (МАДИ),
Московский физико-технический институт (государственный университет), г. Москва, Россия*

Ovar@narod.ru, OVarlamov@gmail.ru

Выделены 3 уровня научных исследований в области искусственного интеллекта (ИИ). Приведены результаты практических расчетов и решений задач, которые экспериментально подтвердили линейную вычислительную сложность логического вывода и/или автоматического конструирования алгоритмов в формализме миварных сетей (MIVAR net's). Программа УДАВ обрабатывает более 1,17 млн объектов и более 3,5 млн правил. Обоснован путь создания ИИ: Мивары => БД + Логика => Контекст => Смысл => Искусственный интеллект! Мивары позволят компьютерам обучаться и понимать смысл, что создаст логический искусственный интеллект. **Ключевые слова:** искусственный интеллект, логический вывод, мивар, миварные сети, экспертные системы.

ВВЕДЕНИЕ

Проблема создания интеллектуальных систем и логического искусственного интеллекта (ИИ) является актуальной и важной. Этой тематике посвящено большое количество научных работ [Поспелов, 1989], [Люгер, 2005], [Джарратано и др., 2007], [Варламов, 2002], [Когаловский, 2005], [Кузнецов, 2009], [Санду и др., 2010]. Миварный подход позволяет предложить новые модели и методы обработки информации и управления [Варламов и др., 2010], [Варламов, 2002], [Варламов, 2011], [Мивар, 2011], [Санду и др., 2010]. Под системами искусственного интеллекта будем понимать активные самообучающиеся логически рассуждающие системы. Ранее были разработаны технологии создания экспертных систем по узконаправленным предметным областям. Это обусловлено сложностями формализованного описания предметных областей и тем, что из-за факториальной вычислительной сложности системы логического вывода не могли обрабатывать много объектов/правил [Кузнецов, 2009], [Поспелов, 1989], [Люгер, 2005]. Для реализации экспертных систем, основанных на знаниях, используются продукции, логика предикатов и др. [Джарратано и др., 2007]. В то же время, получили развитие "интеллектуальные пакеты прикладных программ" (ИППП), которые позволяли решать в автоматизированном режиме задачи в разных областях, где требовались

вычисления и конструирование алгоритмов решения задач. Технологии ИППП развиваются в миварах, сервисно-ориентированных архитектурах.

Российский фундаментальный инновационный миварный подход [Варламов, 2002], [Варламов, 2003], [Варламов, 2011], [Мивар, 2011], [Санду и др., 2010] позволяет использовать эволюционные базы данных и правил для формирования единого глобального пространства накопления и обработки информации. Миварный подход позволит создать мультипредметные экспертные системы и перейти к реальной обработке больших контекстов, представляющих собой информационные модели сложных предметных областей. Уже сейчас мивары обрабатывают более 3,5 миллионов продукции. На основе миварных сетей можно создавать сложные и сверхбольшие информационные модели. Это позволит работать с адекватными контекстами и, в ближайшей перспективе позволит компьютерам "понимать смысл информации". Следовательно, тема данной работы актуальна и перспективна.

1. Миварный подход

В миварном подходе объединяются базы данных, вычисления, сервисы и логика. Мивары изначально были рассчитаны на многомерное и эволюционное представление информации для создания различных интеллектуальных систем, включая глобальные системы обучения, обработки информации и тренажеры [Варламов, 2002], [Варламов, 2003],

[Варламов, 2011], [Мивар, 2011], [Санду и др., 2010]. Информатика занимается 5 основными процессами: сбор, передача, накопление, обработка и представление информации. В настоящее время мивары из этих 5 процессов реализует два: накопление и обработку информации, а миварный подход включает две основные технологии.

1) *Миварная технология накопления информации* - это способ создания глобальных эволюционных баз данных и правил (знаний) с изменяемой структурой на основе адаптивного дискретного миварного информационного пространства унифицированного представления данных и правил, базирующегося на трех основных понятиях "вещь, свойство, отношение" (полученных на основе гносеологии и работ Райбекаса А.Я.).

2) *Миварная технология обработки информации* - это способ решения логико-вычислительных задач путем создания системы логического вывода или "конструирования алгоритмов из модулей, сервисов или процедур" на основе активной обучаемой миварной логико-вычислительной сети правил ("миварная сеть" - "MIVAR net") с линейной вычислительной сложностью.

Миварная технология накопления информации предназначена для хранения любой информации с возможным эволюционным изменением структуры и без ограничений по объему и формам представления. Миварная технология обработки информации предназначена для обработки информации, включая логический вывод, вычислительные процедуры и сервисы. Фактически, миварные сети позволяют развить продукционный подход и создать автоматическую обучаемую логически рассуждающую систему реального времени. Миварное представление логики позволило перейти от однодольных графов продукций к многодольным сетям и выполнять все операции с линейной сложностью.

Мивары быстрее, чем продукции, потому что:

1) в продукциях за основу поиска были взяты правила, которые перебирались для поиска решения, что порождало полный перебор, факториальную сложность и циклы (рисунок 1);

2) в миварных сетях явно выделены две доли: "правила" и "объекты" ("переменные"), а за основу поиска алгоритма логического вывода взяты именно "объекты", которые могут иметь только одно значение и их можно найти только один раз, что исключает циклы и полный перебор (рисунок 1);

3) Миварная сеть может быть задана двухмерной матрицей (рисунок 2), в которой каждое правило знает все свои входные и выходные объекты, а каждый объект, соответственно, знает все свои правила и свои роли в них ("вход" или "выход"), что позволяет избежать перебора и постепенно выявляя новые известные объекты через соответствующие правила, постоянно сокращать размерность исходной миварной матрицы обеспечивая линейную

вычислительную сложность логического вывода относительно общего количества правил в матрице.

При поиске логического вывода в миварной сети, как показано на рисунке 3, заданные объекты как бы "вытягивают" на следующем уровне свои соответствующие правила, которые затем "вытягивают" соответствующие полученные объекты.

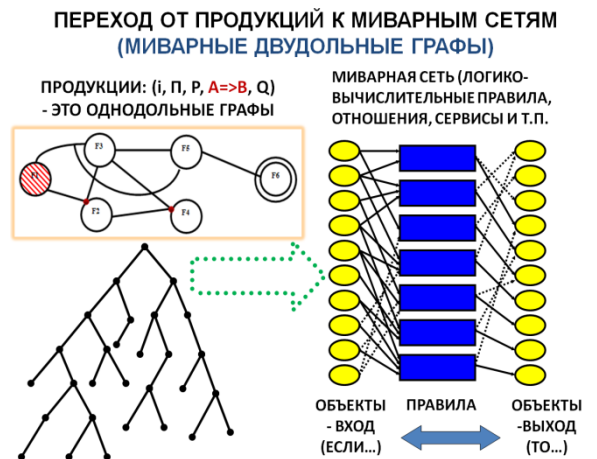


Рисунок 1 – Переход от однодольных графов продукций ("Правила") к двудольным графам миварных сетей ("Объекты; Правила")

ФОРМИРОВАНИЕ ДВУХМЕРНОЙ МАТРИЦЫ МИВАРНОЙ СЕТИ

ОБОЗНАЧЕНИЯ:
M – КОЛИЧЕСТВО ПРАВИЛ В ОПИСАНИИ ЗАДАЧИ;
N – КОЛИЧЕСТВО ВСЕХ ОБЪЕКТОВ - ПЕРЕМЕННЫХ В ПРАВИЛАХ;
 ПРЯМОУГОЛЬНАЯ МАТРИЦА РАЗМЕРОМ (M+1; N+1) – СОДЕРЖИТ ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ, В КОТОРОЙ ПО СТРОКАМ РАСПОЛОЖЕНЫ ВСЕ ПРАВИЛА, А ПО СТОЛБЦАМ – ВСЕ ПЕРЕМЕННЫЕ;
X – ВХОДНЫЕ ПЕРЕМЕННЫЕ В КОНКРЕТНОМ ПРАВИЛЕ;
Y – ВЫХОДНЫЕ ПЕРЕМЕННЫЕ В КОНКРЕТНОМ ПРАВИЛЕ;
Z – ПРИЗНАК ИЗВЕСТНОСТИ ПЕРЕМЕННОЙ, КОТОРЫЙ ПРОСТАВЛЯЕТСЯ В (M+1)-ОЙ СТРОКЕ. ЭТО ЛИБО ЗАДАНЫЕ ВХОДНЫЕ ПЕРЕМЕННЫЕ («ДАНО»), ЛИБО ПОЛУЧЕННЫЕ В ПРОЦЕССЕ РЕШЕНИЯ ЗАДАЧИ;
W – ИСКАОМАЯ ПЕРЕМЕННАЯ, ЗНАЧЕНИЕ КОТОРОЙ НАДО НАЙТИ («НАЙТИ»).

	1	2	3	4	5	...	N-2	N-1	N	N+1
1	X	X		X						
2			X	Y	Y				Y	Y
...						...				
M		X		X	X			Y		
M+1			Z	Z			W	W		

Рисунок 2 – Формирование матрицы миварной сети

ФОРМИРОВАНИЕ ДВУДОЛЬНОЙ МИВАРНОЙ СЕТИ

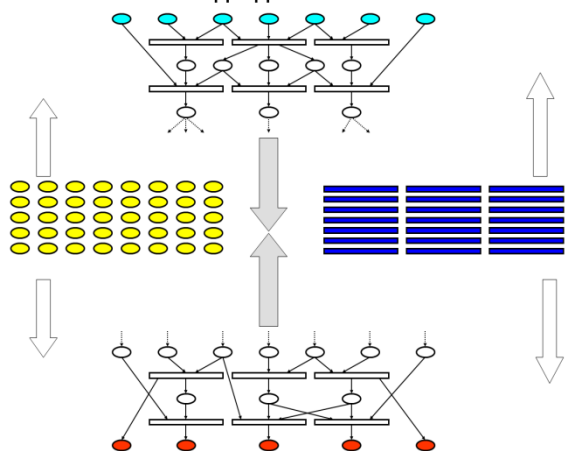


Рисунок 3 – Формирование двудольной миварной сети на основе потока входных данных объектов, типа "Дано" (сверху) и "Найти" (снизу)

Такое "вытягивание" миварами может быть реализовано и "сверху-вниз", и "снизу-вверх", и одновременно от "дано" и от "найти". При этом нет никакого перебора и каждое правило, и каждый объект используется и ищется только один раз. В худшем случае, если есть решение, то будет задействована вся логическая сеть путем такого "вытягивания" с линейной вычислительной сложностью. Если в процессе логического вывода входных данных не будет хватать, то по мере исчерпания всех "запускаемых" правил (т.е. правил у которых известны все входные объекты) процесс остановится и сообщит о нехватке входных данных.

В отличие от традиционных подходов, разделяющих хранение в базах данных, логический вывод и вычислительную обработку, миварный подход позволяет создавать многомерные и эволюционные системы, обрабатывающие информацию в реальном масштабе времени с совмещением логических выводов и вычислительной обработки. Миварный подход - это новый подход для разработки интеллектуальных систем и, в ближайшей перспективе, для создания систем логического искусственного интеллекта [Варламов, 2002], [Варламов, 2003], [Варламов, 2011], [Мивар, 2011], [Санду и др., 2010].

2. Выделение трех уровней исследований в области ИИ

На основе системного анализа выделены 3 основных уровня исследований в области создания искусственного интеллекта.

1. *Рефлексивный ДО-интеллектуальный уровень* - "инстинкты, рефлексы". Нейросети и им подобные методы подобны инстинктивным реакциям, рефлексам и т.п., когда надо мгновенно выполнить некое действие. Образно говоря, если представить систему принятия решения в виде "черного ящика", то на рефлексивном уровне нас интересуют только "входы и выходы" такого ящика - главное, чтобы все эти реакции были правильными и быстрыми. Получается, что мы не пытаемся понять устройство "черного ящика", а только моделируем его работу с целью правильного определения внешних воздействий и поведения системы принятия решений. В основе подобных исследований лежат различные методы статистики, которые имеют ограничения по их применению. Отметим, что этот уровень встречается у животных.

2. *Логический интеллектуальный уровень*. Под этим уровнем будем понимать традиционные логические системы, реализованные в экспертных системах и т.д. Важным отличием от рефлексивного уровня является то, что изучается не поведение, а само устройство "черного ящика". Мы пытаемся "залезть внутрь черного ящика" и смоделировать его, т.е. раскрыть секрет черного ящика и понять его логическую структуру. Термин "интеллект" ближе к логическому осознанному мышлению, когда человек понимает что и как он делает. Это гораздо

медленнее, чем рефлексы и реакции, но зато больше универсальности. Здесь - познание, а нейросети - это диагностика и непосредственное управление. Отметим, что миварный подход работает именно на логическом уровне исследований.

3. *Неформализуемый НАД-интеллектуальный уровень*. В последнее время уделяется внимание вопросам сознания, совести и т.п. свойствам интеллектуальных систем. Эти проблемы явно выходят за пределы "логического осознанного мышления", т.к. даже человек эти процессы не осознает и не всегда может формализовать. Сюда можно отнести и разнообразные эмоции, чувства, мораль и подобные вопросы. Есть позиция: "давайте сначала сделаем хотя бы "логический интеллект", а уж потом займемся вопросами его совести и эмоций". Но, если есть ученые, которым важны и интересны проблемы, которые возникнут сразу после создания ИИ, то этим тоже нужно заниматься. На данном уровне исследований в настоящее время исследуется много неформализуемых задач. Причем, по некоторым предпосылкам, вполне может оказаться, что часть задач принципиально нельзя формализовать или придется работать с бесконечными множествами описания различных предметных областей. В настоящий момент этот уровень изучен меньше других, он является наиболее абстрактным. Учитывая успехи на других уровнях исследований ИИ, вопросы взаимодействия, защиты человечества от роботов и ИИ надо решать заранее, ибо делать это потом будет или уже поздно, или некому. Актуальность и важность этому уровню добавляет и то, что миварный подход уже позволил создать работающие программные комплексы, которые автоматически создают алгоритмы [Варламов и др., 2010], [Варламов, 2002], [Варламов, 2011], [Мивар, 2011], [Санду и др., 2010] и являются прототипом обучаемого логического искусственного интеллекта.

3. Практические реализации миваров

В настоящее время разработано уже несколько программных комплексов на основе миварного подхода [Варламов, 2011], [Мивар, 2011], [Санду и др., 2010]. Прежде всего, развивается проект создания мультипредметных экспертных систем на основе программного комплекса УДАВ с переходом к глобальной миварной активной Интернет-энциклопедии. УДАВ (Универсальный делатель алгоритмов Варламова) предназначен для решения сложных логико-вычислительных задач в реальном времени путем "конструирования алгоритмов" или "поиска маршрута логического вывода" с линейной вычислительной сложностью. Это не простой поиск информации или выполнение жестких алгоритмов, а именно интеллектуальное решение задач с построением алгоритмов, активными запросами недостающей информации и поиском решений.

Результаты практических исследований и экспериментов. В настоящее время, наибольшую трудность представляет описание предметной

области в виде миварной сети. Поэтому для проведения вычислительных экспериментов был создан специальный программный комплекс, основанный на программе УДАВ. В тестовом комплексе реализован генератор миварных матриц описания предметных областей. Например, в процессе экспериментов были созданы миварные матрицы размерности 1 170 007 объектов на 3 510 015 правил. Программе задается количество объектов - переменных. Далее генератор матриц сам генерирует простые арифметические правила для введенного количества объектов. Например, следующие: $a+b=c$ и/или $c-b=a$ и/или $c-a=b$.

Генератор матриц выполняет формирование матрицы и сохраняет ее в специальный файл. Затем тестовая программа считывает тестовую матрицу и начинает обработку. Для однозначности и адекватности проводимых сравнений все генерируемые матрицы в определенном смысле являются подобными. Пользователь может сам сгенерировать матрицы или изменить созданные. Такие эксперименты тоже были проведены. В конце своей работы тестовая программа строит граф решения: наверху рисуются исходные данные, а в внизу объект, который искали. Генератор настроен таким образом, чтобы решение всегда существовало, а его нахождение требовало пропорционального увеличения числа шагов искомого алгоритма решения задачи. Время работы программы определяется по внутренним часам компьютера и может сильно зависеть от общей нагрузки и параллельно выполняемых задач. Время решения задач выводится в миллисекундах (ms). Было рекомендовано запускать тестовую программу с определенным шагом увеличения объектов и в относительно равнозначных условиях внешней загрузки компьютера. Выявлено, что основным ограничением является ресурс оперативной памяти компьютера и корректность работы с ней операционной системы.

Для проведения тестов были задействованы различные компьютеры, начиная от простейших нетбуков с оперативной памятью 512 Мб и до небольших серверов с процессорами Intel 3,8 ГГц и оперативной памятью 4 Гб. Количество ядер не влияло на скорость работы, т.к. основные тесты проводились с использованием однопоточной программы. Использовались два семейства операционных систем: Windows и Mac OS. На семействе ОС Windows получены следующие результаты: до 150 000 объектов и 450 000 правил. Время решения составляло от нескольких миллисекунд до 47 минут.

Наибольший интерес представляют эксперименты, проведенные на ноутбуке MacBook, операционная система Mac OS X версии 10.6.7, процессор Intel Core 2DUO с частотой 2 ГГц и с оперативной памятью DDR3 объемом 4 Гб. На этом ноутбуке в различные периоды времени были проведены тестовые испытания двух версий тестовой программы: В1 и В2. Результаты

испытаний приведены на рисунке 4. Итак, на обычном серийном ноутбуке MacBook неоптимизированная тестовая программа УДАВ, написанная на языке ЯВА (не самом быстром языке), обрабатывает миварные матрицы размерности 1 миллион объектов на 3,5 миллиона правил примерно за 3 часа.

Испытания на максимальное значение получили следующий результат: 1 170 007 объектов - переменных, 3 510 015 правил и время решения 12 239 183 миллисекунды, что примерно равно 200 минутам. Для сравнения приведем следующие оценки: система управления атомными станциями оперирует всего 20 000 правил; описание всей программы средней школы и технического ВУЗа потребует около 300 правил на каждую научную область и в сумме не превысит 100 000 правил. Напомним, что продукции и предикатные системы не могли работать и с сотней правил.

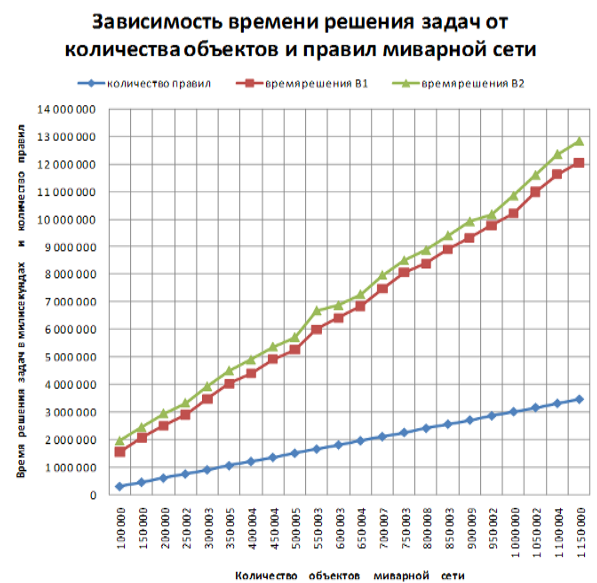


Рисунок 4 – Результаты экспериментов на MacBook

Коботаевым Н.С. была создана многопоточная версия УДАВ на языке C++ и с MPI для систем с распределенной памятью. На рисунке 5 приведены результаты при суммарном количестве процессов $N=8$, $NS=3$, $NC=4$, $C=3$. Время выполнения задач прямо пропорционально количеству правил.

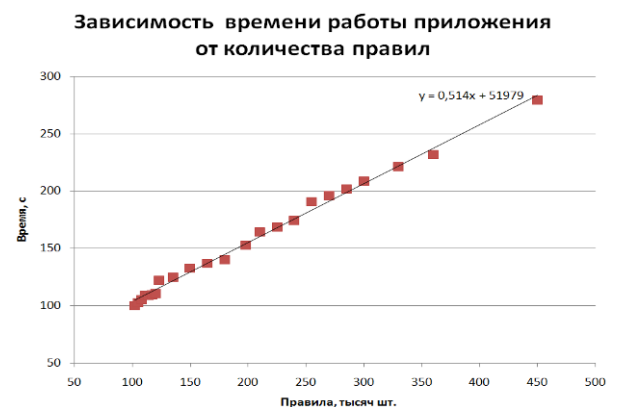


Рисунок 5 – Результаты экспериментов на MPI

Эксперименты проведены на 15 компьютерах. Для каждого значения количества объектов проводилось более 10 однотипных испытания в примерно одинаковых условиях. Расчеты показали, что ошибка измерения времени работы программы не превышала 3 %. Все проведенные эксперименты подтвердили линейную вычислительную сложность логического вывода и/или автоматического конструирования алгоритмов на основе миваров.

Миварный подход не отвергает существующие научные достижения и различные модели, а позволяет объединить их в единую систему для достижения принципиально новых возможностей по обработке и накоплению информации.

Современные "технические средства" в виде ГРИД и облачных вычислений позволяют проводить параллельную обработку и решать требуемые задачи с одновременным использованием различных моделей и программных комплексов. Миварный подход позволяет параллельно решать эти задачи, объединять полученные результаты и обеспечивает реальное время логической обработки и автоматического конструирования алгоритмов в сложных случаях.

4. Этапы обработки информации

Выделяют три основных этапа миварной обработки информации:

- 1) формирование миварной матрицы описания предметной области;
- 2) работа с матрицей и конструирование алгоритма решения заданной задачи;
- 3) по полученному алгоритму выполнение всех вычислений и нахождение ответа.

Первый этап - формирование матрицы является по существу этапом синтеза концептуальной модели предметной области и ее формализации в виде продукционных правил с переходом к миварным сетям формата:

"входные объекты => правила+процедуры => выходные объекты".

В настоящее время именно этот этап является наиболее сложным и требует участия человека-специалиста для создания миварной модели предметной области. С точки зрения обучения важно, что на основе второго этапа УДАВ выполняет "объяснения" для обучаемого, показывает и обосновывает ход решения. Преимущества миварного подхода:

- 1) скорость: линейная вычислительная сложность и реальное время;
- 2) универсальность: решение логических и вычислительных задач;
- 3) эволюция: адаптивное описание и непрерывное решение задач;

4) управление потоком входных данных и оперативная диагностика;

5) активная работа с запросами или уточнениями входных данных на эволюционной сети правил и объектов (обучение).

Рассмотрим этапы миварной обработки на практическом примере. Решение задач треугольников в области геометрии является хорошим тестовым примером для экспертных систем с логическим выводом. Считается, что логический вывод это полно-переборная задача с факториальной вычислительной сложностью, которая зависит от количества правил.

Область "Геометрия. Треугольники" хорошо изучена и описана в учебниках. В нашем случае мы взяли 39 объектов-переменных (углы, стороны, высоты и т.д.), которые описываются 177 правилами - зависимостями одних объектов от других. Решение задач состоит в том, что по заданному набору входных параметров ("дано") надо найти выходные параметры ("найти"). Ученику известны все уравнения и зависимости в этой области. Тогда под каждую "задачу" надо составить алгоритм ее решения, затем вычислить и получить из "дано" требуемые "найти". Известно 2 метода решения:

- 1) логическим выводом строить алгоритмы и
- 2) сформировать матрицу всех алгоритмов для любых наборов "дано" и "найти".

В первом случае вычислительная сложность определяется количеством правил, т.е. $177!$ (факториал). Для второго случая под каждый набор входных и выходных данных требуется до $39!$ строк матрицы. Оба варианта не осуществимы на практике.

Есть третий вариант: миварные сети, которые управляются потоком входных данных и могут автоматически конструировать алгоритмы с линейной вычислительной сложностью логического вывода. Носов А.В. реализовал миварный метод в программе "УДАВ", которая решает все задачи для 39 объектов и 177 правил в реальном времени.

На основе опроса преподавателей средней школы был выявлен наиболее удобный интерфейс проекта, показанный на рисунках 6-10 сначала задают "дано" и "найти", затем программа формирует (автоматически конструирует) алгоритм решения, вычисляет переменные и рисует алгоритм решения задачи. На рисунке 6 показано задание исходных данных цифрами в столбце "значение" и выделение "галочками" требуемых значений в столбце "найти".

В нашем примере заданы 3 высоты - строки P15, P16 и P17, а найти надо параметры в строках: P1 (угол A), P4 (сторона A), P24 (площадь треугольника), P25 (расстояние от точки пересечения медианой стороны а до точки пересечения медиан Ma0) и P39 (расстояние от точки пересечения медиан до стороны C).

Таблица исходных данных:			
id	описание переменных	значение	найти
P1	Угол A, противолежащий стороне a (град)	0	<input checked="" type="checkbox"/>
P2	Угол B, противолежащий стороне b (град)	0	<input type="checkbox"/>
P3	Угол C, противолежащий стороне c (град)	0	<input type="checkbox"/>
P4	Сторона a	0	<input checked="" type="checkbox"/>
P5	Сторона b	0	<input type="checkbox"/>
P6	Сторона c	0	<input type="checkbox"/>
P7	Радиус описанной окружности R	0	<input type="checkbox"/>
P8	Радиус вписанной окружности r	0	<input type="checkbox"/>
P9	Расстояние между центрами вписанной и описанной окружностями d	0	<input type="checkbox"/>
P10	Периметр треугольника P	0	<input type="checkbox"/>
P11	Полупериметр треугольника p	0	<input type="checkbox"/>
P12	Средняя линия Sa, параллельная стороне a	0	<input type="checkbox"/>
P13	Средняя линия Sb, параллельная стороне b	0	<input type="checkbox"/>
P14	Средняя линия Sc, параллельная стороне c	0	<input type="checkbox"/>
P15	Высота Ha, опущенная из угла A к стороне a	157	<input type="checkbox"/>
P16	Высота Hb, опущенная из угла B к стороне b	179	<input type="checkbox"/>
P17	Высота Hc, опущенная из угла C к стороне c	195	<input type="checkbox"/>
P18	Медиана Ma, опущенная из угла A к стороне a	0	<input type="checkbox"/>
P19	Медиана Mb, опущенная из угла B к стороне b	0	<input type="checkbox"/>
P20	Медиана Mc, опущенная из угла C к стороне c	0	<input type="checkbox"/>
P21	Биссектриса La угла A	0	<input type="checkbox"/>
P22	Биссектриса Lb угла B	0	<input type="checkbox"/>
P23	Биссектриса Lc угла C	0	<input type="checkbox"/>
P24	Площадь треугольника	0	<input checked="" type="checkbox"/>
P25	Расстояние от точки пересечения медианой стороны a до точки пересечения медиан MaO	0	<input checked="" type="checkbox"/>
P26	Расстояние от точки пересечения медианой стороны b до точки пересечения медиан MbO	0	<input type="checkbox"/>
P27	Расстояние от точки пересечения медианой стороны c до точки пересечения медиан McO	0	<input type="checkbox"/>
P28	Отрезок, на который биссектриса La делит сторону a - aC	0	<input type="checkbox"/>
P29	Отрезок, на который биссектриса La делит сторону a - aB	0	<input type="checkbox"/>
P30	Отрезок, на который биссектриса Lb делит сторону b - bA	0	<input type="checkbox"/>
P31	Отрезок, на который биссектриса Lb делит сторону b - bC	0	<input type="checkbox"/>
P32	Отрезок, на который биссектриса Lc делит сторону c - cA	0	<input type="checkbox"/>
P33	Отрезок, на который биссектриса Lc делит сторону c - cB	0	<input type="checkbox"/>
P34	Расстояние от вершины A до точки пересечения медиан	0	<input type="checkbox"/>
P35	Расстояние от вершины B до точки пересечения медиан	0	<input type="checkbox"/>
P36	Расстояние от вершины C до точки пересечения медиан	0	<input type="checkbox"/>
P37	Расстояние от точки пересечения медиан до стороны a	0	<input type="checkbox"/>
P38	Расстояние от точки пересечения медиан до стороны b	0	<input type="checkbox"/>
P39	Расстояние от точки пересечения медиан до стороны c	0	<input checked="" type="checkbox"/>

Рисунок 6 – Задание исходных данных цифрами в столбце "значение" и выделение "галочками" требуемых значений в столбце "найти"

На рисунке 7 показаны результаты работы программы по решению заданной задачи: все полученные значения показаны в столбце "значение". Далее, на рисунках 8, 9 и 10 показаны результаты работы программы по автоматическому конструированию алгоритма решения заданной задачи: в окне "Графический вывод работы логики" показан алгоритм решения задачи, который разбит на 3 фрагмента.

На рисунке 8 показаны результаты работы программы по решению заданной задачи: в окне "Графический вывод работы логики" показан алгоритм решения задачи - фрагмент номер 1 (начало алгоритма).

На рисунке 9 показаны результаты работы программы по решению заданной задачи: в окне "Графический вывод работы логики" показан алгоритм решения задачи - фрагмент номер 2 (середина алгоритма).

Таблица исходных данных:			
id	описание переменных	значение	найти
P1	Угол A, противолежащий стороне a (град)	72,805	<input type="checkbox"/>
P2	Угол B, противолежащий стороне b (град)	0	<input type="checkbox"/>
P3	Угол C, противолежащий стороне c (град)	0	<input type="checkbox"/>
P4	Сторона a	232,727	<input type="checkbox"/>
P5	Сторона b	204,124	<input type="checkbox"/>
P6	Сторона c	187,376	<input type="checkbox"/>
P7	Радиус описанной окружности R	0	<input type="checkbox"/>
P8	Радиус вписанной окружности r	0	<input type="checkbox"/>
P9	Расстояние между центрами вписанной и описанной окружностями d	0	<input type="checkbox"/>
P10	Периметр треугольника P	0	<input type="checkbox"/>
P11	Полупериметр треугольника p	0	<input type="checkbox"/>
P12	Средняя линия Sa, параллельная стороне a	0	<input type="checkbox"/>
P13	Средняя линия Sb, параллельная стороне b	0	<input type="checkbox"/>
P14	Средняя линия Sc, параллельная стороне c	0	<input type="checkbox"/>
P15	Высота Ha, опущенная из угла A к стороне a	157	<input type="checkbox"/>
P16	Высота Hb, опущенная из угла B к стороне b	179	<input type="checkbox"/>
P17	Высота Hc, опущенная из угла C к стороне c	195	<input type="checkbox"/>
P18	Медиана Ma, опущенная из угла A к стороне a	157,632	<input type="checkbox"/>
P19	Медиана Mb, опущенная из угла B к стороне b	0	<input type="checkbox"/>
P20	Медиана Mc, опущенная из угла C к стороне c	197,831	<input type="checkbox"/>
P21	Биссектриса La угла A	0	<input type="checkbox"/>
P22	Биссектриса Lb угла B	0	<input type="checkbox"/>
P23	Биссектриса Lc угла C	0	<input type="checkbox"/>
P24	Площадь треугольника	18 269,07	<input type="checkbox"/>
P25	Расстояние от точки пересечения медианой стороны a до точки пересечения медиан MaO	52,545	<input type="checkbox"/>
P26	Расстояние от точки пересечения медианой стороны b до точки пересечения медиан MbO	0	<input type="checkbox"/>
P27	Расстояние от точки пересечения медианой стороны c до точки пересечения медиан McO	65,944	<input type="checkbox"/>
P28	Отрезок, на который биссектриса La делит сторону a - aC	0	<input type="checkbox"/>
P29	Отрезок, на который биссектриса La делит сторону a - aB	0	<input type="checkbox"/>
P30	Отрезок, на который биссектриса Lb делит сторону b - bA	0	<input type="checkbox"/>
P31	Отрезок, на который биссектриса Lb делит сторону b - bC	0	<input type="checkbox"/>
P32	Отрезок, на который биссектриса Lc делит сторону c - cA	0	<input type="checkbox"/>
P33	Отрезок, на который биссектриса Lc делит сторону c - cB	0	<input type="checkbox"/>
P34	Расстояние от вершины A до точки пересечения медиан	0	<input type="checkbox"/>
P35	Расстояние от вершины B до точки пересечения медиан	0	<input type="checkbox"/>
P36	Расстояние от вершины C до точки пересечения медиан	131,898	<input type="checkbox"/>
P37	Расстояние от точки пересечения медиан до стороны a	0	<input type="checkbox"/>
P38	Расстояние от точки пересечения медиан до стороны b	0	<input type="checkbox"/>
P39	Расстояние от точки пересечения медиан до стороны c	92,828	<input type="checkbox"/>

Рисунок 7 – Результаты работы программы по решению заданной задачи: все полученные значения показаны в столбце "значение"

На рисунке 10 показаны результаты работы программы по решению заданной задачи: в окне "Графический вывод работы логики" показан алгоритм решения задачи - фрагмент номер 3 (конец алгоритма).

Практическая польза: преподаватель может готовить индивидуальные задания и проверять контрольные задания, а ученики сами получают объяснения в программе.

Таким образом, Миварная экспертная система "УДАВ. Геометрия" в реальном времени решает все задачи для 39 объектов и 177 правил, что помогает и учителям, и ученикам.

В настоящее время ведутся работы по добавлению новых объектов и правил из различных математических разделов. Кроме того, будут добавлены объекты и правила, описывающие физические процессы.

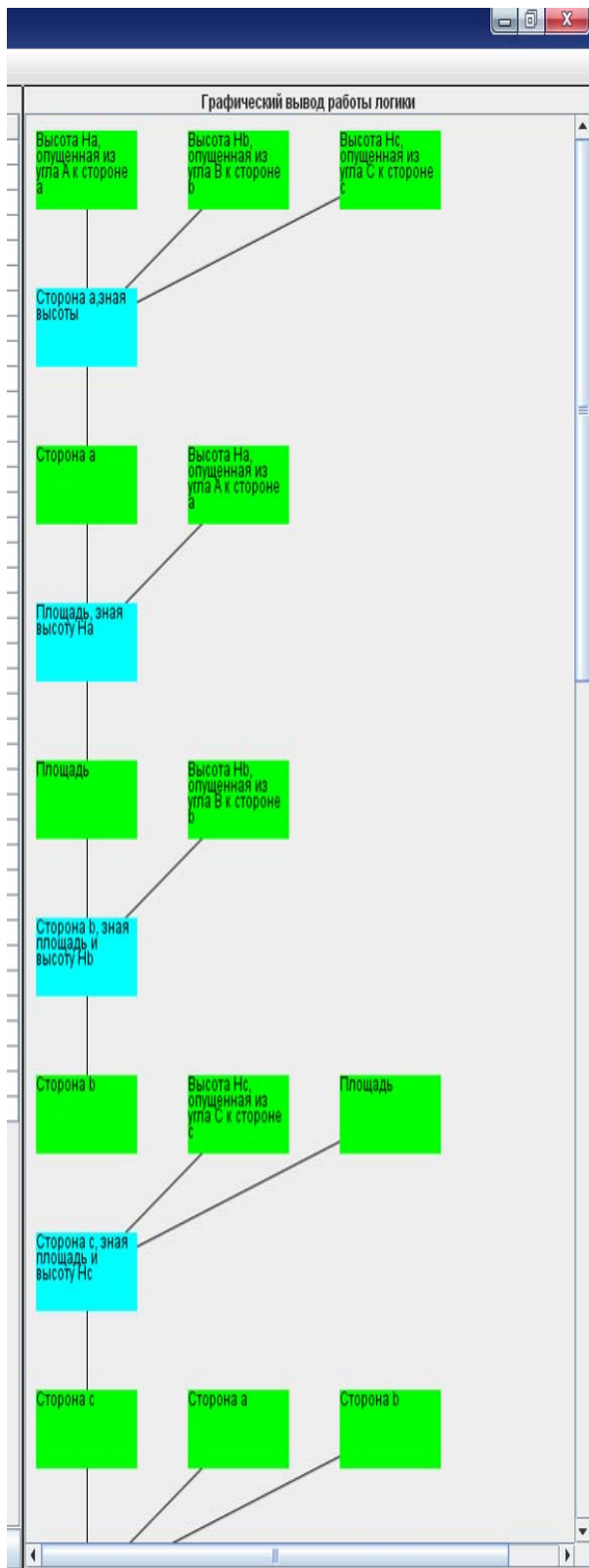


Рисунок 8 – Результаты работы (начало алгоритма)

На основе миварных технологий реализуются инновационные проекты [Варламов, 2011], [Мивар, 2011]. На Конкурсе русских инноваций наши проекты принимают участие в номинациях "Белая книга", "Перспективный проект" и "Инновационный проект".

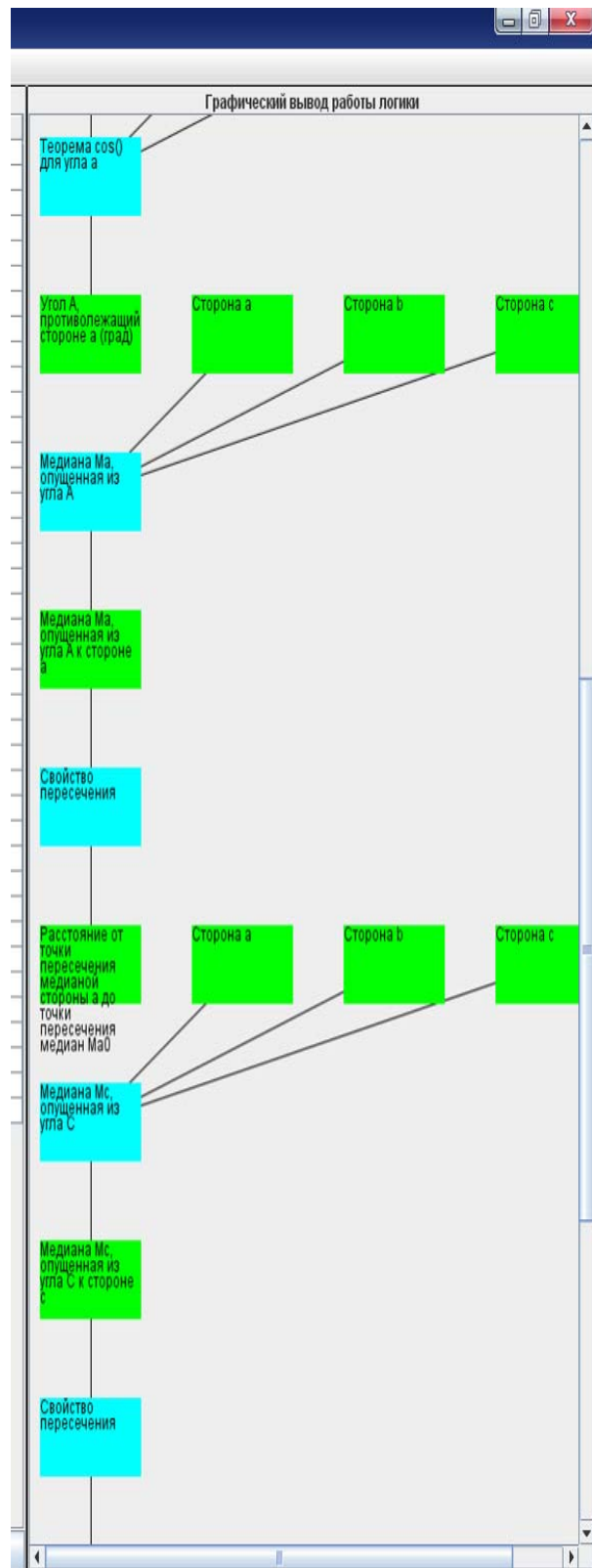


Рисунок 9 – Результаты работы (середина алгоритма)

Наши проекты в области создания искусственного интеллекта, разработки миварной информационной инфраструктуры российской электроэнергетики и другие доходили до Второго тура и становились финалистами Конкурса русских инноваций в разные годы.

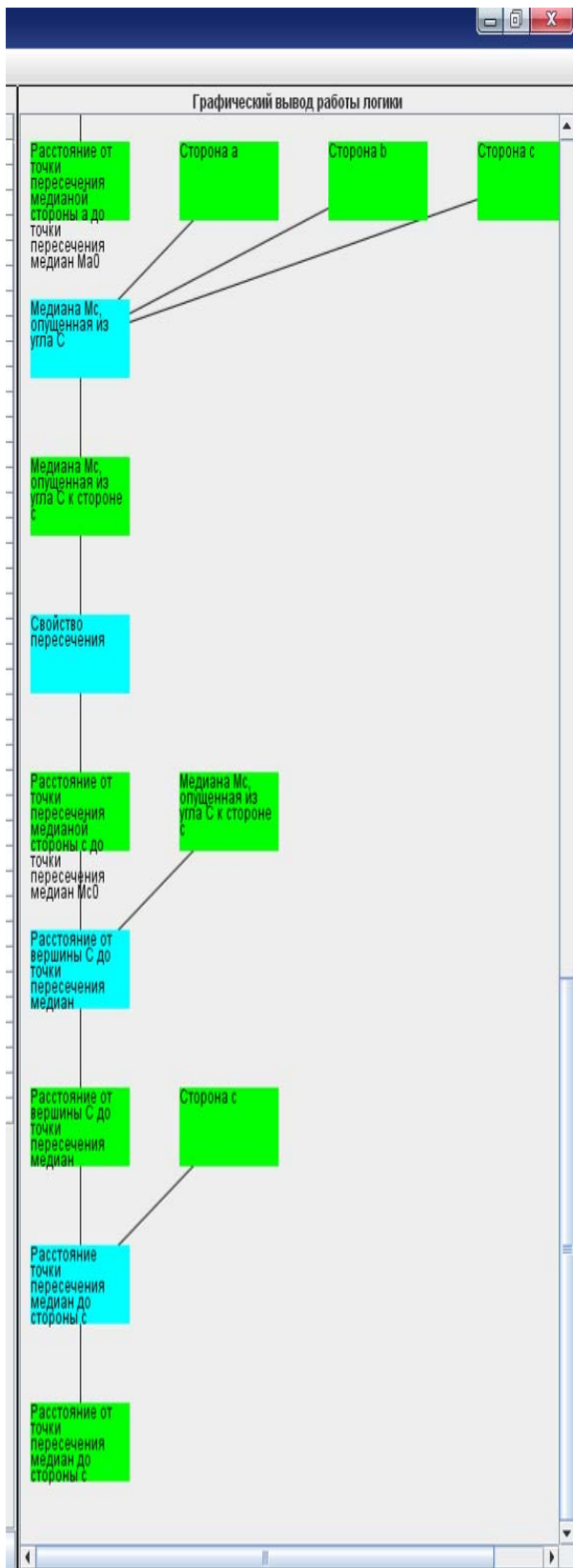


Рисунок 10 – Результаты работы (конец алгоритма)

В перспективе получится глобальная Миварная активная интернет-энциклопедия, в которой будут собраны объекты и правила для всех известных предметных областей и мы получим глобальную мультипредметную экспертную систему.

5. Моделирование контекста, понимание смысла и создание логического искусственного интеллекта на основе миварных технологий

Итак, обосновано, что миварные технологии позволяют эволюционно накапливать любые объемы данных и логически обрабатывать их с линейной вычислительной сложностью. На основе миваров создан программный комплекс УДАВ, который обрабатывает более 1,17 млн переменных и более 3,5 млн правил на обычных компьютерах и ноутбуках. Следовательно, миварный подход позволяет на новом уровне перейти к работе с большими контекстами и информационными моделями предметных областей.

В большинстве современных задач в различных областях, включая распознавание образов, поиск информации, перевод и аннотирование текстов, а также в других проблемах области искусственного интеллекта ключевую роль играет необходимость работы с огромными и структурно сложными контекстами.

По существу, контекст - это информационная модель предметной области, в которой есть данные и правила их обработки. В определенном смысле, адекватный контекст и есть "смысл информации". Одновременность накопления фактов и обработки миллионов правил в миварных системах позволяют перейти к созданию сложных и многоуровневых контекстов и адекватных информационных моделей предметных областей, что проиллюстрировано на рисунке 11.

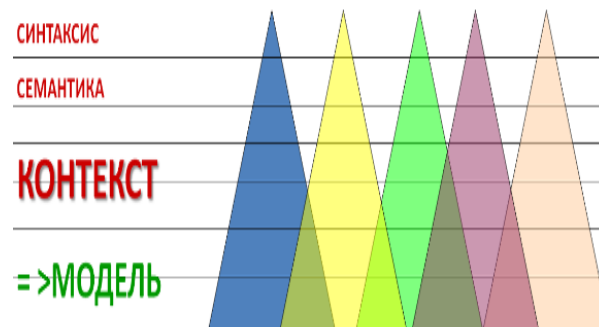


Рисунок 11 – Иллюстрация необходимости постепенного наращивания по уровням и переход от синтаксиса и семантики к обработке контекста

Такой переход можно осуществлять путем создания миварных информационных циклов и перехода от обработки синтаксиса и семантики к обработке контекста и информационных моделей предметных областей, что проиллюстрировано на рисунке 12. Сложные контексты, в свою очередь, позволят на новом уровне решать все интеллектуальные задачи на основе работы "компьютеров со смыслом информации". Как только компьютеры начнут работать со смыслом, можно говорить о создании полноценного логического искусственного интеллекта.

МИВАРНЫЕ ИНФОРМАЦИОННЫЕ ЦИКЛЫ => КОНТЕКСТ = ИНФОРМАЦИОННАЯ МОДЕЛЬ

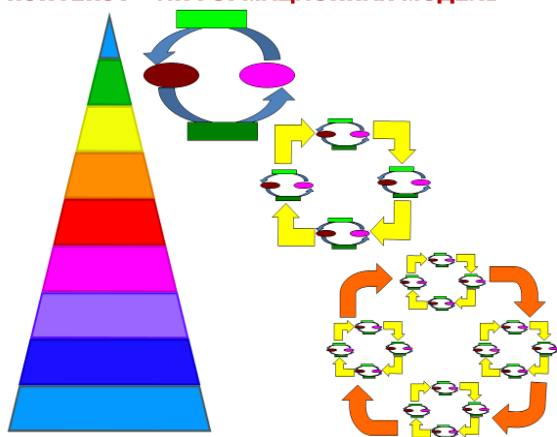


Рисунок 12 – Иллюстрация создания миварных информационных циклов и перехода от обработки синтаксиса и семантики к обработке контекста и информационных моделей предметных областей

В настоящий момент ограничения на возможности миваров по созданию сложных контекстов не обнаружены, т.к. возможности существующих компьютерных систем, объединенных сетями и ГРИД-технологиями, практически безграничны.

Таким образом, пролучаем следующую логическую цепочку: Мивары => БД + Логика => Контекст => Смысл => Искусственный интеллект!

Следовательно, мивары позволяют создавать новое поколение информационных систем, накапливающих огромные объемы данных и обрабатывающих неограниченное количество логических правил. Такие системы позволяют реализовать работу со сложным и неограниченным контекстом. А это, в свою очередь, позволит компьютерам постепенно, по мере их обучения, т.е. наращивания контекста и усложнения "миварными циклами" информационных моделей, работать со смыслом информации и породит полноценный логический искусственный интеллект!

ЗАКЛЮЧЕНИЕ

Целесообразно выделить 3 уровня научных исследований в области искусственного интеллекта. Миварный подход на логическом уровне позволяет создавать активные глобальные обучаемые логически рассуждающие автоматические познающе-диагностические системы реального времени. Для экспертных систем нового поколения вместо продукций и однодольных графов целесообразно использовать двудольные графы миварных сетей.

На основе миварных сетей создан программный комплекс УДАВ, который обрабатывает более 1,17 млн объектов и более 3,5 млн правил на обычных компьютерах и ноутбуках. Приведены результаты практических расчетов и решений различных

прикладных задач, которые на практике подтверждают линейную вычислительную сложность логического вывода и/или автоматического конструирования алгоритмов в формализме двудольных миварных сетей.

Впервые предоставлены возможности создания мультимедийных экспертных систем, способных работать с миллионами объектов и продукционных правил в формализме двудольных миварных сетей.

В работе обоснована следующая логическая цепочка: Мивары => БД + Логика => Контекст => Смысл => Искусственный интеллект!

Мивары позволяют создать новое поколение интеллектуальных систем, накапливающих огромные объемы данных и обрабатывающих очень большое (практически неограниченное) количество логических правил, сервисов и вычислительных процедур.

Миварные системы позволяют работать компьютерам с огромным контекстом в реальное время. Следовательно, компьютеры начинают работать со смыслом информации, что и порождает полноценный искусственный интеллект на логическом уровне научных исследований.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Варламов, 2002] Варламов О.О. Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство. - М.: Радио и связь, 2002. - 288 с.
- [Варламов, 2003] Варламов О.О. Основы многомерного информационного развивающегося (миварного) пространства представления данных и правил // Информационные технологии. 2003. № 5. С.42-47.
- [Варламов и др., 2010] Варламов О.О., Санду Р.А., Владимиров А.Н., Носов А.В., Оверчук М.Л. Миварный подход к созданию мультимедийных активных экспертных систем в целях обучения информационной безопасности и управления инновационными ресурсами в образовании // Известия Южного федерального университета. Технические науки. 2010. № 11. С. 226 - 232.
- [Варламов, 2011] Веб-сайт // <http://www.ovar.narod.ru>. 2011.
- [Владимиров и др., 2010] Владимиров А.Н., Варламов О.О., Носов А.В., Потапова 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.
- [Джарратано и др., 2007] Джарратано Дж., Райли Г. Экспертные системы: принципы разработки и программирование, 4е издание. - М.: Вильямс, 2007. 1152 с.
- [Коголовский, 2005] Коголовский М.Р. Энциклопедия технологий баз данных. - М.: Финансы и статистика, 2005. - 800 с.
- [Кузнецов, 2009] Кузнецов О.П. Дискретная математика для инженера. 6-е изд., стер. - СПб: Издательство "Лань", 2009. - 400 с.
- [Люгер, 2005] Люгер Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание.: Пер. с англ. - М.: Издательский дом "Вильямс", 2005. - 864 с.
- [Мивар, 2011] Веб-сайт компании МИВАР // <http://www.mivar.ru>. 2011.
- [Поспелов, 1989] Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов. - М.: Радио и связь, 1989. - 184 с.
- [Санду и др., 2010] Санду Р.А., Варламов О.О. Миварный подход к созданию интеллектуальных систем и искусственного интеллекта. Результаты 25 лет развития и ближайшие перспективы. - М.: Стандартинформ, 2010. - 339 с.

**THE MIVAR TECHNOLOGIES OF LOGICAL
ARTIFICIAL INTELLIGENCE CREATING:
LOGICAL INFERENCE WITH LINEAR
COMPUTATIONAL COMPLEXITY WITH
MORE THAN 3 MILLIONS PRODUCTION
RULES AND POSSIBILITY TO UNDERSTAND A
MEANING THROUGH MODELING OF A BIG
CONTEXT**

Varlamov O.O.

*Professor of Department of Applied Mathematics
at Moscow State Automobile and Road Technical
University (MADI), Professor of Department of
Radio and Information Technology FRTK Moscow
Institute of Physics and Technology (State
University), Moscow, Russia,*

Ovar@narod.ru, OVarlamov@gmail.ru

It is reasonable to define 3 levels of scientific researches in the sphere of Artificial Intelligence. The MIVAR method on the logical level allows to create automatic systems which have the following qualities: activeness, globality, learning capability, ability to make logical inferences and give conclusions in real time. It is reasonable for new generation of expert systems to use bipartite graphs of the MIVAR nets instead of production rules and monocots graphs.

Keywords: MIVAR, MIVAR net, logical inference, computational complexity, artificial intelligence, intelligent systems, expert systems, General Problem Solver.

The theoretical transition from the graphs of production systems to the bipartite graphs of the MIVAR nets is shown. Examples of the implementation of the MIVAR nets in the formalisms of matrixes and graphs are given. The linear computational complexity of algorithms for automated building of objects and rules of the MIVAR nets is theoretically proved. On the basis of the MIVAR nets the UDAV software complex is developed, handling more than 1.17 million objects and more than 3.5 million rules on ordinary computers. The results of experiments that confirm a linear computational complexity of the MIVAR method of logical inference. The MIVAR approach unifies and develops achievements from different scientific domains: databases, computational problems, logic processing, and includes two main technologies:

1. The MIVAR technology of information accumulation – is a method of creating of global evolutionary bases of data and rules (knowledge) with changeable structure based on the adaptive discrete MIVAR information space of unified representation of data and rules which bases on three main definitions: “Thing, Property, Relation”.
2. The MIVAR technology of information processing - is a method of creation of the system of logic inference or “automatic construction of algorithms from modules, services and procedures” based on the active MIVAR net of rules with linear computational complexity.

The MIVAR technology of information accumulation is designed for keeping any information with possible evolutionary change of its structure and without any restrictions of its volume and the form of representation. The MIVAR technology of information processing is designed for the processing of information, including logic inference, computational procedures and services. In fact, MIVAR nets allow to develop production approach and to create an automatic learning logically thinking system. The MIVAR approach unifies and develops production systems, ontology, semantic nets, service-oriented architectures, multi-agent systems and other modern information technologies.

There are three basic stages of MIVAR data processing:

- 1) Creation of the MIVAR matrix for the description of subject domain;
- 2) Working with the matrix and the construction (designer) of the algorithm for solving of the required problem;
- 3) Execution of all computations basing on the acquired algorithm.

The first stage can be seen as the formalization of subject domain in the form of productions with the following transition to the MIVAR rules:

"input objects – rules/procedures/services – output objects".

Currently, it is the most difficult stage that requires the participation of a human-specialist (expert) for creation of the MIVAR model of subject domain. On the second stage, automated construction of the algorithm and logic inference is being implemented. The input data is represented in the form of the MIVAR matrix of the description of subject domain and specified input (“GIVEN”) and required (“TO FIND”) objects-variables. On the third stage, the process of solution basing on the obtained algorithm is being executed. For the moment, in the software complex UDAV the work of the second and the third stages is combined. Currently, there are more than 7 different realizations of the MIVAR method. In some of them, three main stages are processed separately, but in this article the work of the UDAV software complex is described where all three stages can be combined.

Our paper is important for other researchers in the sphere of AI, because we present a new formalism of knowledge representation through which logical processing of data is increased significantly. Observing some restrictions, our formalism (MIVAR nets) provides linear computational complexity of algorithms or automated building (logical inference) on production systems base. The theoretical transition from AND/OR graphs of production systems to the bipartite graphs of the MIVAR (Multidimensional Informational Variable Adaptive Reality) nets is shown. Our results can be used for creation of intelligent systems, expert systems, General Problem Solver in different spheres.



УДК 510.63

ВЕРОЯТНОСТНАЯ ЛОГИКА И ОРТОГОНАЛЬНЫЙ БАЗИС СИЛЛОГИСТИКИ

Сметанин Ю.М

ГБОУ ВПО «Удмуртский государственный университет», г. Ижевск, Россия

gms1234gms@rambler.ru

В работе рассматривается алгебра случайных событий и её интерпретация в невырожденной булевой алгебре на основе множеств. Показано, что классическое использование понятий логика и вероятность вполне совместимо с инженерными логико-вероятностными исчислениями и вероятностной логикой, начало которой восходит к классическим работам в области искусственного интеллекта. Показано, что некоторые весьма трудные задачи могут быть решены с помощью предлагаемого логического описания и точной интерпретации.

Ключевые слова: алгебраическая система, исчисление конститuentных множеств, вероятность, ортогональный базис силлогистики.

1. Введение

Исторически сложилось три точки зрения на совместное использование понятий логика и вероятность:

- 1. классическая [Колмогоров 1986]
2. инженерная (логико - вероятностное исчисление - ЛВИ) [Рябинин 2003]
3. исследователей искусственного интеллекта (вероятностная логика) [Nilson 1986].

Под термином вероятностная логика в данной работе понимается по сути классическая алгебра случайных событий и с исключенным двузначным отношением X subseteq Y между случайными событиями X и Y. В работе показано, что различие во взглядах и «трудности» совмещения понятия логика и вероятность происходят от того, что Колмогоровский подход к исчислению вероятностей случайных событий использует невырожденную булеву алгебру на основе множеств [Владимиров 1969], а инженерный подход и подход искусственного интеллекта, использует вырожденную булеву алгебру на основе множеств индикаторов случайных событий, по сути своей - неадекватную модель объективной реальности.

2. Постановка задачи

В работах [Сметанин 2009, 2010] предложен ортогональный базис силлогистики, имеющий выгодное отличие от системы простых суждений Аристотеля его определение дано, в том числе работе [Сметанин 2011b], опубликованной в данном сборнике.

Семь расширенных Жергонновых отношений изображены на фоне универсума смотри рис. 1

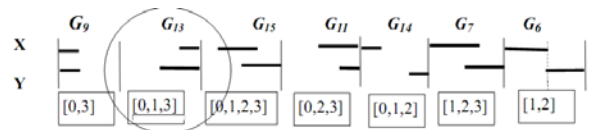


Рис. 1 Расширенные Жергонновы отношения

G11, G13, - правостороннее и левостороннее включение;

G9 - равенство (равносильность);

G15 независимое пересечение (независимость);

G6 - полное противоречие (событие X выполняется когда не происходит Y - полная несовместимость X' совпадает с Y);

G14 неполное противоречие - соподчиненность (события несовместны и их объединение не равно универсуму - неполная несовместимость X' совместно с Y);

G7 - зависимое пересечение.

Последний термин явно не используется в теории вероятностей и не является отрицанием термина независимость событий. Все пары событий, кроме тех которые находятся в отношении G15, являются зависимыми. Остальные восемь отношений являются вырожденными в том смысле, что допускают равенство универсуму сравниваемых множеств или их дополнений. Аристотель не допускал пустых терминов в рассуждениях [Брусенцов 1998].

На Рис. 2 показано, как соотносятся отношение между множествами X и Y и отношения между их индикаторами x и y.

Имя	Кон.	Соотн./мера	Образ	
G1		$X'Y = \emptyset$	$A(X', Y)(X' = \emptyset)(Y' = \emptyset)$	$(x' < y)(y' = 0)$
		$X'Y' = \emptyset$		
		$XY' = \emptyset$		
		$XY \neq \emptyset$	$P(XY) = 1$	
G2		$X'Y' = \emptyset$	$A(X', Y')(X' = \emptyset)(Y = \emptyset)$	$(x' < y')(x' = 0)(y = 0)$
		$X'Y = \emptyset$		
		$XY' \neq \emptyset$	$P(XY') = 1$	
		$XY = \emptyset$		
G3		$X'Y' = \emptyset$	$A(X'Y)(X' = \emptyset)(A(X', Y))$	$(x' \leq y)(x' = 0)(x' \leq y')$
		$X'Y \neq \emptyset$		
		$XY' \neq \emptyset$	$P(XY') = P(Y')$	
		$XY \neq \emptyset$	$P(XY) = P(Y)$	
G4		$X'Y' = \emptyset$	$A(X, Y)(Y' = \emptyset)(X = \emptyset)$	$(x < y)(x = 0)(y' = 0)$
		$X'Y \neq \emptyset$	$P(X'Y) = 1$	
		$XY' = \emptyset$		
		$XY = \emptyset$		$(x \leq y)(y' = 0)(x' \leq y)$
G5		$X'Y' = \emptyset$	$A(X, Y)(Y = \emptyset)A(X'Y)$	
		$X'Y \neq \emptyset$	$P(X'Y) = P(X')$	
		$XY' = \emptyset$		
		$XY \neq \emptyset$	$P(XY) = P(X)$	
G6		$X'Y' = \emptyset$	$Eq(X, Y')(X \neq \emptyset)(Y \neq \emptyset)$	$x = y$
		$X'Y \neq \emptyset$	$P(X'Y) = P(Y)$	
		$XY' \neq \emptyset$	$P(XY') = P(X)$	
		$XY \neq \emptyset$	$P(XY) = 0$	
G7		$X'Y' = \emptyset$	$A(X', Y)(X \neq \emptyset)(Y \neq \emptyset)$	$x' \leq y$
		$X'Y \neq \emptyset$	$P(X'Y) = P(X')$	
		$XY' \neq \emptyset$	$P(XY') = P(Y')$	
		$XY \neq \emptyset$	$P(XY) = 1 - P(X') - P(Y')$	
G8		$X'Y' \neq \emptyset$	$P(X'Y') = 1$	
		$X'Y = \emptyset$	$A(X, Y')(X = \emptyset)(Y = \emptyset)$	$(x = y)(x = 0)(y = 0)$
		$XY' = \emptyset$		
		$XY = \emptyset$		
G9		$X'Y' \neq \emptyset$	$P(X'Y') = P(X') = P(Y')$	
		$X'Y = \emptyset$	$Eq(X, Y)(X \neq \emptyset)(Y \neq \emptyset)$	$x = y$
		$XY' = \emptyset$		
		$XY \neq \emptyset$	$P(XY) = P(X) = P(Y)$	
G10		$X'Y' \neq \emptyset$	$P(X'Y') = P(X')$	
		$X'Y = \emptyset$	$A(X, Y')(X \neq \emptyset)(Y = \emptyset)$	$(x \leq y')(y = 0)(x' \leq y')$
		$XY' \neq \emptyset$	$P(XY') = P(X)$	
		$XY = \emptyset$		
G11		$X'Y' \neq \emptyset$	$P(X'Y') = P(X')$	
		$X'Y = \emptyset$	$A(X', Y')(X \neq \emptyset)(Y \neq \emptyset)$	$x' \leq y'$
		$XY' \neq \emptyset$	$P(XY') = 1 - P(X') - P(Y)$	
		$XY \neq \emptyset$	$P(XY) = P(Y)$	
G12		$X'Y' \neq \emptyset$	$P(X'Y') = P(Y')$	
		$X'Y \neq \emptyset$	$P(X'Y) = P(Y)$	
		$XY' = \emptyset$	$A(X, Y)(X = \emptyset)(Y \neq \emptyset)$	$(x \leq y)(x = 0)(x \leq y')$
		$XY = \emptyset$		
G13		$X'Y' \neq \emptyset$	$P(X'Y') = P(Y')$	
		$X'Y \neq \emptyset$	$P(X'Y) = 1 - P(X) - P(Y')$	
		$XY' = \emptyset$	$A(X, Y)(X \neq \emptyset)(Y \neq \emptyset)$	$x \leq y$
		$XY \neq \emptyset$	$P(XY) = P(X)$	
G14		$X'Y' \neq \emptyset$	$P(X'Y') = 1 - P(X) - P(Y)$	
		$X'Y \neq \emptyset$	$P(X'Y) = P(Y)$	
		$XY' \neq \emptyset$	$P(XY') = P(X)$	
		$XY = \emptyset$	$A(X, Y)(X \neq \emptyset)(Y \neq \emptyset)$	$x \leq y'$
G15		$X'Y' \neq \emptyset$	$X'Y' + X'Y + XY' +$	$y' + x'y + xy' +$
		$X'Y \neq \emptyset$	$+XY = U$	$+xy = I$
		$XY' \neq \emptyset$		
		$XY \neq \emptyset$		

Рис. 2 Проекция Жергонновых отношений

В столбце «Кон.» На рис. 2 показано, какие конститутенты в данном отношении являются пустыми либо непустыми множествами. В столбце «Соотн./мера» показано, как жергонново соотношение между множествами выражается через функторы ортогонального базиса силлогистики и дополнительные утверждения о равенстве (неравенстве) пустому множеству образующих соотношение множеств. Из соотношений, рассмотренных на рис.2, следует [Сметанин 2011a, 2011b] отсутствие изоморфизма между ними.

Алгебра множеств является булевой алгеброй относительно строгого и нестрогого включения - как естественного упорядочения. С каждой такой алгеброй автоматически связывается (в случае частичного упорядочения на основе нестрогого включения) изоморфная ей булева алгебра соответствующих характеристических функций (индикаторов) смотри [Сметанин 2001a, 2011b]. В случае частичного упорядочения на основе строгого включения между алгеброй множеств и алгеброй характеристических функций, устанавливается неизоморфное отображение. Случай нестрогого включения не обеспечивает односмысловость суждений, вследствие чего был введен ортогональный базис силлогистики [Сметанин 2009, 2010, 2011a, 2011b]

Таким образом, польку случайное событие имеет интерпретацию в форме множества, а высказывание интерпретируется как пропозициональная переменная с двумя возможными значениями “истина” и “ложь”, то изоморфизм между алгеброй событий и алгеброй их индикаторов возможен только в случае если множества - события в универсуме (достоверном событии), упорядочены отношением нестрогого порядка. Для случая строгого частичного порядка показано, что изоморфизма нет, а есть гомоморфизм. С точки зрения просто частичного порядка безразлично, что рассматривать – алгебраическую систему множеств или изоморфную ей алгебраическую систему характеристических функций этих множеств. Это отражено в теореме Стона [Горбатов 1976]. Для случая строгого частичного порядка справедлива

Теорема 1. Алгебраическая система $\sum = \{X_i\}$

задаваемая системой множеств с определенным на них отношением строгого частичного порядка (строгое включение) не изоморфна алгебраической системе S их индикаторов, на которой строгий частичный порядок из $\sum = \{X_i\}$ отражается в частичный порядок.

То есть, в случае рассмотрения строгого частичного порядка строгого изоморфизма между этими системами нет, а есть гомоморфное отображение первой во вторую. Проекционная модель [Вальков 1985] на основе индикаторов настолько грубая, что любое непустое подмножество универсума алгебраической системы множеств отражается в ней как ноль либо единица. Это, в свою очередь, является одной из причин парадоксов материальной импликации. Булевой алгеброй называется дистрибутивная структура с неравными друг другу нулем 0 и

единицей 1, в которой всякий элемент имеет дополнение. Таким образом, булева алгебра всегда содержит не менее двух элементов. Алгебра, содержащая только 0 и 1, называется вырожденной [Владимиров с.19]. **Классическая логика построена на основе вырожденной булевой алгебры, в которой 0 отождествлен с абстрактной ложью, а 1 с абстрактной истиной. То есть, она отражает объективную реальность как систему событий (минуя моделирование событий множествами) даже не в систему характеристических функции этих множеств, а в абстрактные, по Гильберту, пропозициональные переменные.**

В работах [Сметанин 2009, 2010, 2011b] в качестве модели высказывания предлагается рассматривать множество. Значение же высказывания определяется индикатором этого множества - высказывательной переменной. Эта идея восходит к самому Аристотелю, особенно наглядно ее представил Жергонн, однако алгебраический подход не получил развития в математической логике, и только в работах [Кулик1997, Кулик2010] получены существенные результаты. Причин здесь несколько и одна из наиболее веских - многосмысловость простых суждений Аристотеля (смотри работу [Сметанин 2011b] из настоящего сборника). Устранение многосмысловости за счет введения ортогонального базиса силлогистики (ОБ) сразу позволило значительно продвинуться вперед:

1. построить точную интерпретацию рассуждений в форме алгебраической системы;
2. построить эффективный алгоритм проверки логического следования заключений из посылок силлогизма;
3. указать на неправильные модусы Аристотеля и Б. Рассела [Сметанин 2011a];
4. доказать отсутствие парадоксов материальной импликации и вернуть ей смысл логического следования, а также установить причину, по которой анализируя словесную продукцию со связкой «если ..., то» логики, используя вырожденную булеву алгебру вынуждены признавать наличие парадоксов [Сметанин 2011a].
5. свести воедино три означенные в начале публикации точки зрения на совместное использование логики и вероятности.

Рассмотрение суждений как множеств позволяет сопоставить терминологию элементарной теории вероятностей и суждений ортогонального базиса. Здесь имеет место однооднозначное соответствие.

Испытание (опыт) – алгебраическая система с образующими $\sum = \{X_i\}$. X_i - случайное событие и одновременно суждение на основе ортогонального базиса. В случае составного (с использованием связей) события ему сопоставляется сложное суждение на основе простых суждений ОБ. Отношения между случайными событиями X_i (влечет, равносильно, независимость, несовместность, полное противоречие (событие противоположное данному) находятся во взаимно однозначном соответствии с семью расширенными жергонновыми отношениями и тремя

функторами ОБ, смотри рис. 1 и комментарии к нему. Рассмотрение набора суждений и логического следования одних суждений (событий) из комплекса других приводит к необходимости рассмотрения многоместных отношений (предикатов). Наглядная форма представления таких отношений – линейные диаграммы Лобанова [Лобанов2009], которыми можно иллюстрировать интерпретацию алгебраических систем выражающих постановку задач полисиллогистики и вероятностной логики. Ввиду недостатка места автор не будет далее углубляться в теорию. Вместо этого мы постараемся привести убедительные примеры, иллюстрирующие новые возможности вероятностной логики в связи с использованием в ней ОБ.

При исследовании вопроса о том, какие множества можно построить посредством операций (объединения «+», пересечения «*», дополнения до универсума «'») $X^2 = U \setminus X$ из порождающих n произвольных множеств X_1, X_2, \dots, X_n , вводится важное понятие конституенты. Обозначим

$$X_i^{\sigma_i} = \begin{cases} X_i, & \text{если } \sigma_i = 1 \\ \bar{X}_i, & \text{если } \sigma_i = 0 \end{cases}$$

Множество вида,

$$\prod_{i=1}^n X_i^{\sigma_i} = X_1^{\sigma_1} X_2^{\sigma_2} \dots X_n^{\sigma_n} \text{ где } \sigma_i = 0 \text{ или}$$

$\sigma_i = 1$ назовем конституентой.

Общее число не пустых конституент не превосходит 2^n . Каждой конституенте можно сопоставить двоичный набор $\prod_{i=1}^n x_i^{\sigma_i} = x_1^{\sigma_1} x_2^{\sigma_2} \dots x_n^{\sigma_n}$ длины n , где

$x_i^{\sigma_i}$ есть характеристическая функция множества

$$X_i^{\sigma_i}, x_i^{\sigma_i} = \begin{cases} x_i, & \sigma_i = 1 \\ \bar{x}_i, & \sigma_i = 0 \end{cases}$$

Определение 1. Булева переменная x является характеристической переменной (индикатором) множества X , если она определены следующим образом смотри (1), e – произвольный элемент универсума U .

$$x = \begin{cases} 1, & \text{если } e \in X \\ 0, & \text{если } e \notin X \end{cases}$$

Характеристическая функция множества (индикатор) X ставит в соответствие любому элементу универсума e двоичную переменную x , которая равна 1, если e принадлежит множеству X либо равна 0, если e не принадлежит X . Набор из индикаторов множеств составляющих конституенту будем называть характеристической функцией конституенты.

Определение 2. Базисом заданной алгебраической системы (системы с фиксированным порядком

номеров порождающих множеств) будем называть множество непустых конституент данной системы.

Определение 3. Базовым множеством номеров (БМН или BSN) заданной алгебраической системы назовем множество номеров непустых конституент, представленное в десятичном, либо в двоичном виде¹.

Алгебраическая система с зафиксированным линейным порядком множеств носителей далее будет называться заданной.

Приведем важную теорему, которая позволяет распознавать многомерные отношения, содержащие независимые в совокупности случайные события.

Теорема 2. Система $\Sigma = \{X_i\}$ из n множеств являющаяся образующей алгебраической системы с заданным отношением строгого порядка (строгое включение) может быть взаимно однозначно сопоставлена системе случайных событий независимых в совокупности тогда и только тогда, когда $БМН(\Sigma) = [0..2^n]$. Другими словами, тогда и только тогда, когда все 2^n конституент, построенных из $\{X_i\}$ являются непустыми множествами.

Теорема доказывается по индукции.

3. Примеры решения задач

Сначала рассмотрим задачу с которой начиналась вероятностная логика. [Nilson 1986]. Здесь данная задача приводится в нашей транскрипции.

Задача 1. (Вероятностная логика в ИИ) Дана совокупность случайных событий X и Y , заданы - вероятность $P(X)=p_1$ и вероятность $P(X \subset Y)=p_2$. Найти (оценить) вероятность $P(Y)=p$.

Сразу отметим, что опыт в котором воспроизводятся X и Y определяет одно из невырожденных отношений из рисунков 1 и 2, поэтому значения вероятностей p_1 и p_2 в обязательном порядке зависят от реализуемого отношения, что понимал Нельсон, вводя понятия возможных миров. Для расчета искомой вероятности достаточно рассмотреть все невырожденные отношения и рассчитать вероятность p по ним.

Для G_{15} p_2 должно быть равно $1-P(XY')=p_2$ в силу независимости X и Y имеем

$1-p_1(1-P(Y))=p_2$ или $1-p_1+p_1P(Y)=p_2$. Отсюда $P(Y)=(p_1+p_2-1)/p_1$. Последняя формула соответствует ответу для данной задачи, полученному в работе [Кулик 2010] другие случаи там не рассматриваются. Для G_{14} мы с необходимостью имеем $p_2=0$ и $P(Y)<1-P(X)$. Для G_{11} - $Y \subset X$ с необходимостью имеем $p_2=1$ и $p \geq p_1$. Для G_{13} - $X \subset Y$ с необходимостью имеем $p_2=p$. Для G_9 - $X=Y$ необходимо имеем $p_2=1$ и $p=p_1$. Для G_7 и G_6 с необходимостью имеем $p_2=0$ и $p \leq 1-p_1$.

Задача 2. (Пример взят из работ по ЛВИ).

Рассмотрим один из примеров оценки риска и эффективности при борьбе двух компаний за заказ при противодействии третьей компании. Дружественные компании A и B хотят получить выгодный заказ. Компания C может помешать им. Компания C (событие $X3$ с вероятностью p_3) вступит в борьбу за получение заказа и будет противодействовать компаниям A и B . Противодействие компании C могут заставить компанию A (событие $X5$ с условной вероятностью $p_5=P(X5/X3)$ при $P(X5/X3')=0$) и компанию B (событие $X4$ с условной вероятностью $p_4=P(X4/X3)$ при $P(X4/X3')=0$) отказаться от намерений. Если же компания B (событие $X1$ с вероятностью $p_1=P(X1/X5')$ при $P(X1/X5)=0$) и компания A (событие $X2$ с условной вероятностью $p_2=P(X2/X4')$ при $P(X2/X4)=0$) смогут получить заказ, то прибыль компании A составит $E=6$ млрд. и прибыль компании B составит $E=2$ млрд.

В примере вероятности p_1, p_2, p_3, p_4, p_5 назначены методом экспертной оценки с учетом внешних факторов и капитала фирм A, B и C .

Определить вероятность получения заказа компанией A или B и ожидаемый выигрыш двух компаний A и B .

Решим задачу традиционным способом.

Определим вероятности $P(X1), P(X2)$ в силу парной независимости ($X1$ и $X2$), ($X14$ и $X3$), ($X2$ и $X3$) и несовместимости ($X1$ и $X5$), ($X2$ и $X4$),

$X1 = (X1X5') + (X1X5) = X1X5'$ с учетом того, что $X1, X5$ несовместимы, $X1X5 = \emptyset$;

$X2 = (X2X4') + (X2X4) = X2X4'$ с учетом того, что $X2, X4$ несовместимы, $X2X4 = \emptyset$;

$X4 = X4X3 + X4X3' = X3X4$ с учетом того, что $X3'X4 = \emptyset$

$X5 = X5X3 + X5X3' = X3X5$ с учетом того, что $X3'X5 = \emptyset$

$P(X4) = P(X3) * P(X4/X3) + P(X3') * P(X4/X3') = p_3 * p_4 = r_4$;

$P(X5) = P(X3) * P(X5/X3) + P(X3') * P(X5/X3') = p_3 * p_5 = r_5$;

$P(X1) = P(X5') * P(X1/X5') = (1-r_5) * p_1 = (1-p_3 * p_5) * p_1 = r_1$;

$P(X2) = P(X4') * P(X2/X4') = (1-r_4) * p_2 = (1-p_3 * p_4) * p_2 = r_2$

Примем вероятности событий.

$p_1=0.85; p_2=0.95; p_3=0.7; p_4=0.4; p_5=0.5$

Целевое событие $Z=X1+X2$.

Вероятностный полином (функция) достижения цели $P(Z)=r_1 + r_2 - r_1 * r_2 = (1-p_3 * p_5) * p_1 + (1-p_3 * p_4) * p_2 - (1-p_3 * p_5) * p_1 * (1-p_3 * p_4) * p_2 = 0,85859$

Введем в условия примера показатели эффективности достижения трех разных целей:

$E1 = 6$, если свою цель достигнет только компания A $P1=P(X1X2') = 0,17459$;

$E2 = 2$, если свою цель достигнет только компания B $P2=P(X1'X2) = 0,30609$;

$E3 = 8$ если свои цели достигнуть обе компании, $P3=P(X1X2)=0,37791$;

Используя вычисленные вероятности определяем суммарную эффективность достижения трех (математическое ожидание) целей равна

$T = E1 * P1 + E2 * P2 + E3 * P3 = 6 * 0,17459 + 2 * 0,30609 + 8 * 0,37791 = 4,683$

¹ Сммотри пояснение к рисунку 3.

Решим задачу способом, ориентированным на применение компьютера, используя описание взаимосвязей между случайными событиями посредством функторов ортогонального базиса.

Построим систему случайных событий (алгебру) отражающую логическую структуру задачи, которая аналитически выглядит так:

Причинно следственные связи явлений (событий) процесса борьбы за заказ можно описать в виде четырех утверждений:

1. событие $X1$ влечет событие противоположное $X5$;
2. событие $X2$ влечет событие противоположное $X4$;
3. событие противоположное $X3$ влечет не наступление $X5$;
4. событие противоположное $X3$ влечет не наступление $X4$;

Тот же результат получается если, описать логику задачи в виде равенств равносильных отношению включения: $X1=X1X5'$; $X5=X5X3$; $X2=X2X4'$; $X4=X4X3$;

Для системы с заданными соотношениями включения и в виде равенств получены, с помощью программы, одинаковые базовые множества номеров Ur и построена линейная диаграмма рис 1.

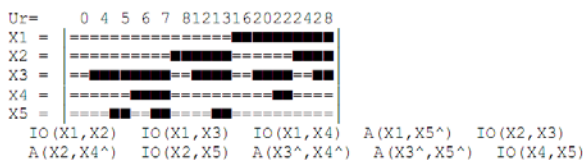


Рис. 3 Диаграмма логических связей задачи.

Номера входящие в множество Ur это номера непустых конституент упорядоченной (перенумерованной) системы множеств $X1, X2, X3, X4, X5$. Сопоставление конституент номерам и наоборот осуществляется очень просто. Например, номер 28 переводится в двоичную систему счисления как 11100 и ему ставится в соответствие конституента $X1X2X3X4'X5'$ где единица на i -ом месте сопоставляется множеству Xi , а ноль на j -ом месте дополнению множества Xj , то есть множеству

$$U \setminus X_j = X_j'$$

На диаграмме рис. 1 случайным событиям, которые могут произойти соответствуют подмножества номеров универсума Ur : Например, событию $X1=(X1X5')$ с вероятностью $r1$ соответствует множество номеров $X1=X1X5'=[16, 20, 22, 24, 28]$, при этом вероятность $P(X1X5')=P(X5')*P(X1/X5')=(1-r3)*p1=(1-p3*p5)*p1=r1$.

Последовательно, с учетом логических связей вычислим вероятность всех конституент, составляющих базовое множество номеров $Ur=[4,5,6,7,8,12,13,16,20,22,24,28]$. Используя логическое описание задачи, получим:

$$[28]=X1X2X3X4'X5' = X1X2X3 \text{ в силу 1 и 2. В силу независимости } X1, X2, X3 \text{ } P[28]=r1*r2*r3;$$

$$[24]=X1X2X3'X4'X5' = X1X2X3' \text{ в силу 1 и 2, } P[24]=r1*r2*(1-r3);$$

$$[2]=X1X2'X3X4X5' = X1X2'X4 \text{ в силу 1 и 3, } P[2]=r1*(1-r2)*r4;$$

$$[20]=X1X2'X3X4'X5' = X1X2'X3X4' \text{ в силу 1 и 3}$$

$$P[20]=r1*(1-r2)*(r3-r4);$$

$$[16]=X1X2'X3'X4'X5' = X1X2X3' \text{ в силу 3 и 4,}$$

$$P[16]=r1*(1-r2)*(1-r3);$$

$$[13]=X1'X2X3X4'X5 = X1'X2X5 \text{ в силу 2 и 4, } P[13]=(1-r1)*(r2)*(r5);$$

$$[12]=X1'X2X3X4'X5' = X1'X2X3X5' \text{ в силу 2 и 4}$$

$$P[12]=(1-r1)*(r2)*(r3-r5);$$

$$[8]=X1'X2X3'X4'X5' = X1'X2X3'X5' \text{ в силу 2 и 4}$$

$$P[8]=(1-r1)*(r2)*(1-r3);$$

$$[7]=X1'X2'X3X4X5 = X1'X2'X4X5 \text{ в силу 3 и 4, } P[7]=(1-r1)*(1-r2)*r4*r5;$$

$$[6]=X1'X2'X3X4X5' = X1'X2X4X5' \text{ в силу 3, } P[6]=(1-r1)*(1-r2)*r4*(1-r5);$$

$$[5]=X1'X2'X4'X5 \text{ в силу 1, } P[5]=(1-r1)*(1-r2)*(1-r4)*r5;$$

$$[4]=X1'X2'X3X4'X5' \text{ в силу 3 и 4 и независимости } X4 \text{ и } X5, P[4]=(1-r1)*(1-r2)*(r3-r4-r5+r4*r5);$$

$$[0]=X1'X2'X3'X4'X5' = X1'X2'X3' \text{ в силу 3 и 4 } P[0]=(1-r1)*(1-r2)*(1-r3);$$

Легко проверить, что

$$P[16..28]=P[16]+P[20]+P[22]+P[24]+P[28]=P(X1)=r1*(1-r2)*(1-r3)+r1*(1-r2)*(r3-r4)+r1*(1-r2)*r4+r1*r2*r3=r1;$$

$$P[8..13]+p[24,28]=P(X2)=(1-r1)*(r2)*(r5)+(1-r1)*(r2)*(r3-r5)+(1-r1)*(r2)*(1-r3)+r1*(1-r2)*r4+r1*r2*r3=r2;$$

$$P[4..7]+P[12,13]+P[20,22]+P[28]=P(X3)=(1-r1)*(1-r2)*(r3-r4-r5+r4*r5)+(1-r1)*(1-r2)*(1-r4)*r5+(1-r1)*(1-r2)*r4*r5+(1-r1)*(1-r2)*r4*(1-r5)+(1-r1)*(r2)*(r3-r5)+(1-r1)*(r2)*(r5)+r1*(1-r2)*(r3-r4)+r1*(1-r2)*r4+r1*r2*r3=r3;$$

$$P[6,7]+P[22]=P(X4)=(1-r1)*(1-r2)*r4*r5+(1-r1)*(1-r2)*r4*(1-r5)+r1*(1-r2)*r4=(1-r1)*(1-r2)*r4*(1-r2)+r4-r4*r5+r1*(1-r2)*r4=(1-r1)*(1-r2)*r4+r1*(1-r2)*r4=r4*(1-r2);$$

$$P[5,7,13]=(1-r1)*(1-r2)*(1-r4)*r5+(1-r1)*(1-r2)*r4*r5+(1-r1)*(r2)*(r5)=(1-r1)*r5;$$

$$P[8..28]=P(X1+X2)=r1+r2-r1*r2;$$

Этот и предыдущий результат указывают на детерминированную связь случайных событий $X1$ и $X5'$, $X2$ и $X4'$ одно не может произойти без другого. Кроме того, имеют место еще 2 равенства.

$$P[0..7]=P(X1'X2')=(1-r1)*(1-r2)=1-P(X1+X2)=(1-r1-r2+r1*r2);$$

$$P[0, 4, 5, 6, 7, 8, 12, 13, 16, 20, 22, 24, 28] = 1$$

Произведем декомпозицию исходной логики - алгебраической модели задачи (рис. 4).

В системе событий можно выделить два кластера на основе силы логических связей между ними это системы $\{X1, X3, X5\}$ и $\{X2, X3, X4\}$. Их линейные диаграммы показаны на рис. 4.

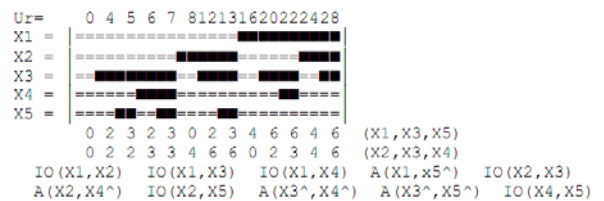


Рис. 4 Декомпозиция исходной задачи

То есть исходную задачу можно решать, разбив ее на две подзадачи. Решение приведено ниже. Разбиение можно осуществить на исходной диаграмме удалив из нее сначала множества $X2$ и $X4$, затем $X1$

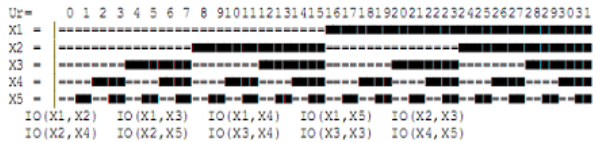


Рис. 8 Диаграмма логических связей задачи об электроснабжении.

Эта диаграмма удовлетворяет условиям теоремы 1, то есть события $\{X1, X2, X3, X4, X5\}$ независимы в совокупности. На рис. 9 показана часть этой диаграммы, конститuentы, которой образуют целевое событие Z.

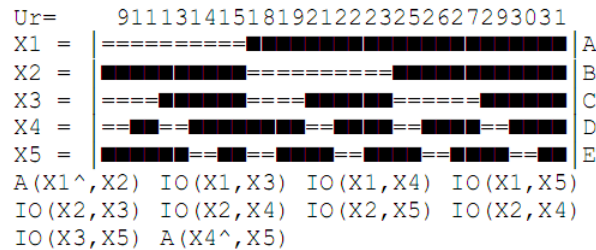


Рис. 9 Часть линейной диаграммы задачи, соответствующая событию целевому событию

$$Z = X1X4 + X1X3X5 + X2X4 + X2X3X4.$$

- $P[31] = r1 * r2 * r3 * r4 * r5;$
- $P[30] = r1 * r2 * r3 * r4 * (1 - r5);$
- $P[29] = r1 * r2 * r3 * (1 - r4) * r5;$
- $P[28] = r1 * r2 * r3 * (1 - r4) * (1 - r5);$
- $P[27] = r1 * r2 * (1 - r3) * r4 * r5;$
- $P[26] = r1 * r2 * (1 - r3) * r4 * (1 - r5);$
- $P[23] = r1 * (1 - r2) * r3 * r4 * (1 - r5);$
- $P[22] = r1 * (1 - r2) * r3 * r4 * r5;$
- $P[21] = r1 * (1 - r2) * r3 * (1 - r4) * r5;$
- $P[20] = r1 * (1 - r2) * r3 * (1 - r4) * (1 - r5);$
- $P[19] = r1 * (1 - r2) * (1 - r3) * r4 * r5;$
- $P[15] = (1 - r1) * r2 * r3 * r4 * r5;$
- $P[14] = (1 - r1) * r2 * r3 * r4 * (1 - r5);$
- $P[13] = (1 - r1) * r2 * r3 * (1 - r4) * r5;$
- $P[11] = (1 - r1) * r2 * (1 - r3) * r4 * r5;$
- $P[10] = (1 - r1) * r2 * (1 - r3) * r4 * (1 - r5);$

Для случая когда $r1=r2=r3=r4=r5=0,5$ имеем $P(Z)=0,5$, что соответствует результату Рябинина И.А. и Кулика Б.А.

Задача 4

Рассмотрим известный пример Бернштейна, иллюстрирующий, то обстоятельство, что попарно независимые случайные события числом не менее трех могут образовать систему событий не являющихся независимыми в совокупности.

У пирамиды четыре грани A, B, C, D. Грань A имеет цвет E, грань B имеет цвет F, грань C имеет цвет G, грань D трехцветная. Она имеет цвета E, F, G.

Случайные события при бросании тетраэдра:

A - выпала грань A; B - выпала грань B; C - выпала грань C; D - выпала грань D; E - выпал цвет E; F - выпал цвет F; G - выпал цвет G.

Попарно несовместимыми событиями являются события A, B, C, D. Являются ли независимыми в совокупности события E, F, G? Нет, не являются. Для того чтобы они были независимыми в совокупности необходимо, чтобы их БМН было равно $[0, 1, 2, 3, 4, 5, 6, 7]$. Это следует из теоремы 1

Её условие не выполняется, так как после удаления событий A, B, C, D БМН системы будет равно $[7, 1, 2, 4]$. Один из возможных вариантов задания логических связей задачи для компьютерной программы записан между служебными словами ZADNAME и END.

```
ZADNAME: BERN
NUMBERVAL:7
A B C D E F G
NUMBERPOS:13
EQ(A*B,U^)$
EQ(A*C,U^)$
EQ(A*D,U^)$
EQ(B*C,U^)$
EQ(C*D,U^)$
EQ(B*D,U^)$
EQ(A+D,E)$
EQ(B+D,F)$
EQ(C+D,G)$
A(D,E)$
A(D,F)$
A(D,G)$
EQ(A+B+C+D,U)$ END
```

Этим ограничениям сопоставляется линейная диаграмма причинно - следственных связей испытания, показанная на машинограмме рис. 10. Если из нее удалить множества, X1, X2, X3, X4, соответствующие событиям A, B, C, D, то БМН оставшейся системы множеств E=X5, F=X6, G=X7, то есть множество номеров непустых конститuent будет равно не $[0..7]$, а будет равно $[1, 2, 4, 7]$. Таким образом, мы доказали, что случайные события E, F, G не являются независимыми в совокупности.

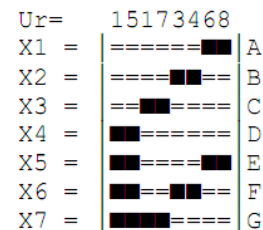


Рис. 10 Линейная диаграмма для примера Бернштейна

Рассмотренные примеры решения задач вероятностной логики, то есть задач теории вероятностей описанных в форме односмысловых логических суждений на основе функторов ортогонального базиса позволяют ставить и решать задачи компьютеризации интеллектуальной деятельности. По существу программная реализация точной интерпретации комплекса суждений является своеобразным усилителем естественного интеллекта.

ЗАКЛЮЧЕНИЕ

В статье анализируется актуальная проблема искусственного интеллекта совместное использование логики и вероятности при решении задач. Как отмечается в работе [Кулик 2010] это является нетривиальной задачей. Автором предлагается алгебраическая (на основе алгебры множеств) интерпретация аристотелевской силлогистики изоморфная алгебре случайных событий относительно отноше-

ния строгого включения одного множества в другое. Которая позволяет решать задачи проверки гипотез о логическом следовании суждения из множества посылочных суждений, то есть устанавливать правильность рассуждений в форме полисиллогизмов. По свидетельству профессора Непейводы Н.Н. количество работ в мировой науке, посвященных этому вопросу превышает 500, что указывает на сложность проблемы. Для ее решения предложен целый ряд расходящихся решений, предложенных самими авторитетными логиками. В работах [Сметанин 2010, 2011a, 2011b] предложена точная и непротиворечивая модель силлогистики, которая в данной публикации естественным образом «погружена» в вероятностную меру. В таком случае естественно возникает вопрос об адекватности этой модели содержательному смыслу. Наш анализ показывает, что эта модель не менее адекватна, чем лучшие из предложенных решений. Предлагаемый подход к компьютеризации решения задач является новым. Развитие предлагаемого подхода видится в создании формального исчисления конституентных множеств и построения системы искусственного интеллекта для решения вероятностных и логических задач, в частности возможно получить новые научно – обоснованные результаты в теории байесовских сетей [Тулупьев 2006], [Рассел 2006] и вероятностных выводов. Обратим внимание читателей на то, что предложенная модель не только является объединение всех трех подходов упомянутых во введении, но и естественным образом позволяет отразить не только причинно – следственные (детерминированные и стохастические) связи событий, но и естественное упорядочение их прообразов во времени в объективной реальности. Частичный порядок следования событий во времени легко укладывается в линейный порядок их следования в заданной алгебраической системе [Сметанин 2010]. В заключение автор обращается к научным работникам с предложением о партнерстве. Предлагается рассмотреть возможности совместных разработок для компьютеризации решения задач инжиниринга и реинжиниринга бизнес - процессов (BP), задач искусственного интеллекта, задач расчета надежности систем и безопасности бизнеса, задач анализа законодательных актов, задач модернизации и компьютеризации сложившейся в 20 веке дидактической системы обучения логике. Все это произвести на основе подхода развиваемого проф. Непейводой Н.Н. в области неклассических логик и хаотического программирования и подхода к решению задач полисиллогистики и вероятностной логики развиваемого доц. Сметаниным Ю.М. на основе ортогонального базиса силлогистики, являющегося альтернативой базису силлогистики Аристотеля.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Бочаров 2010] Бочаров В.А., Маркин В.И. Силлогистические теории. – М.: Прогресс – Традиция, 2010. 336 с.
 [Вальков 1985] Вальков К.И. Проекционное моделирование и автоматизация. Учебное пособие для факультета повышения квалификации. Л.: ЛИСИ, 1985, 86 с.
 [Владимиров 1969] Владимиров Д.А. Булевы алгебры. М.:

Наука 1969. С. 320, л.

- [Колмогоров] Колмогоров А. Н. Теория вероятностей и математическая статистика: //Сб. статей— М.: Наука, 1986.—535 с.
 [Рябинин 2003] Рябинин И.А. Логико-вероятностное исчисление как аппарат исследования надежности и безопасности структурно-сложных систем. // АИТ, «Наука», М, 2003, № 7, с. 178-186.
 [Лобанов 2009] Лобанов В.И. Русская вероятностная логика – М.: «Русская правда» 2009.- 320 стр.
 [Сметанин 2009] Сметанин Ю. М. Ортогональный базис силлогистики // Вестник Удмуртского университета. Серия математика, механика. Компьютерные науки. Вып. 4 . 2009 г. С .155-166
 [Сметанин 2010] Сметанин Ю. М. Алгоритм решения полисиллогизмов в ортогональном базисе посредством исчисления конституентных множеств // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2010. Вып. 4.-С. 172-185.
 [Сметанин 2011a] Анализ парадоксов материальной импликации в ортогональном базисе силлогистики // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2011. Вып. 4.- С.144-162
 [Сметанин 2011b] Медицинская диагностика и ортогональный базис силлогистики.// В настоящее сборнике.
 [Горбатов 1976] Горбатов В.А. Теория частично упорядоченных систем. – М.: «Советское радио», 1976. – 336 с.
 [Кулик 1997] Кулик Б.А. Логические основы здравого смысла. Под редакцией Поспелова Д.А. – СПб.: Политехника, 1997. – 131 с.
 [Кулик 2010] Алгебраический подход к интеллектуальной обработке данных и знаний . Б.А. Кулик, А.А. Зуенко, А.Я. Фридман. – СПб.: изд-во Политех. Ун-та, 2010. - 235 с.
 [Порецкий 1884] Порецкий П.С. О способах решения логических равенств и об одном обратном способе математической логики.// Сборание протоколов заседаний секции физико - математических наук общества естествоиспытателей при Казанском университете, т. 2, Каз., 1884.
 [Брусенцов 1998] Брусенцов Н.П. Искусство достоверного рассуждения. Неформальная реконструкция аристотелевой силлогистики и булевой математики мысли. - М.: Фонд «Новое тысячелетие», 1998.
 [Nilson 1986] Nilson N. J. Probabilistic Logic /N/ J/ Nilson // Artificial Intelligense. 1986 N 28/ - pp 71-87.
 [Тулупьев 2006] Тулупьев А. Л., Николенко С. И., Сироткин А. В. Байесовские сети: Логико-вероятностный подход. — СПб.: Наука, 2006. — 607 с.
 [Рассел 2006] Рассел, Стюарт, норвиг, Питер. Искусственный интеллект: современный подход, 2 – е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1408 с.: ил.]

Probabilistic logic and ortogonal basis of syllogistics

Smetanin Yu.M.

Udmurt State University, Izhevsk

gms1234gms@rambler.ru

RESUME

In this paper, we consider the algebra of random events and its interpretation in a non-degenerate Boolean algebra on the bases of sets/ It is shown that the classical use of the concepts of logic and probability is quite compatible with the engineering logical –and probabilistic estimate and probabilistic logic, the beginning of which goes back to classical works in the field of artificial intelligence/ It is shown that some of most labour-wide tasks can be solved with the help of the proposed logical description and accurate interpretation.

Key words: algebraic system, the calculation of constituent sets, probability, orthogonal line basis of syllogistic.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ РЕШАТЕЛЕЙ ЗАДАЧ

Заливако С.С., Д.В. Шункевич

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

zalivako@mail.ru

shu.dv@tut.by

В работе приводится описание открытой семантической технологии проектирования интеллектуальных решателей задач. Отдельное внимание уделяется методике проектирования решателей и операций, являющихся составными частями таких решателей. Также в работе рассмотрено несколько примеров использования технологии при проектировании конкретных интеллектуальных систем по различным предметным областям.

Ключевые слова: интеллектуальная система, интеллектуальный решатель задач, логический вывод

ВВЕДЕНИЕ

В настоящее время большую актуальность имеет переход от ориентирования проектировщиков интеллектуальных систем с навязываемой (предлагаемой) машины обработки знаний на проектирование решателей задач из готовых компонентов. В основе этого подхода лежат общие принципы организации машин обработки знаний, позволяющих осуществить интегрирование различных моделей решения задач, как существующих, так и новых. Это предоставляет возможность реализации данной технологии на основе любых моделей решения задач.

В работе рассматривается методика проектирования интеллектуальных решателей задач, которая входит в состав комплексной открытой технологии проектирования интеллектуальных систем OSTIS [OSTIS, 2011].

1. Цель и задачи предлагаемой технологии

В основе данной работы лежит тезис о том, что различных интеллектуальных системах могут применяться различные модели решения задач. Гипотетически возможно существование

универсального решателя задач, объединяющего в себе все известные способы и методы решения задач. Однако использование такого решателя в прикладных целях не является реальным и тем более целесообразным. Таким образом, наиболее приемлемым вариантом становится создание библиотеки совместимых между собой компонентов, из которых впоследствии может быть скомпилирован решатель, удовлетворяющий необходимым требованиям.

Основным требованием, предъявляемым при таком подходе к реализации моделей решения задач, является представление их в соответствующем формальном виде – как многоагентной системы, работающей над общей памятью. Приведение существующих моделей к такому формализму является нетривиальной задачей, хоть и не приносит радикальных изменений в текущее состояние теории решения задач в целом.

На основании сказанного выше, можно выдвинуть следующую гипотезу: любые модели решения задач могут быть представлены в описанном ранее формализме с целью максимального упрощения интеграции данных моделей между собой.

Целью данной работы является создание открытой семантической технологии компонентного проектирования интеллектуальных решателей задач,

позволяющей осуществить компиляцию готового решателя из имеющихся компонентов.

К рассматриваемой технологии выдвинуты следующие требования:

- открытость (технология является частью открытого проекта OSTIS);
- расширяемость (возможность свободного добавления новых компонентов);
- гибкость (пользователь может использовать лишь часть компонентов из предложенной библиотеки для обеспечения требуемого функционала разрабатываемой интеллектуальной системы);
- доступность (сравнительно неподготовленный пользователь может использовать технологию для создания интеллектуального решателя задач по интересующей его предметной области);
- универсальность (технология может быть использована в любых предметных областях, обеспечивая при этом полный функционал в случае необходимости);
- модульность (технология включает структурированную библиотеку компонентов обработки знаний);
- полнота (технология должна обеспечивать решение как можно большего набора задач из заданной предметной области за конечное время).
- асинхронность и параллельность (технология должна обеспечивать возможность параллельной асинхронной работы операций и, как следствие, возможность параллельного решения ряда задач)

Таким образом, при достижении поставленной цели возникают следующие проблемы:

- обеспечение предметной независимости и универсальности компонентов технологии и всей технологии в целом;
- обеспечение синхронизируемости отдельных компонентов технологии между собой;
- обеспечение синхронизируемости всей технологии в целом с другими направлениями проекта OSTIS, в частности с технологией проектирования баз знаний;
- обеспечение самодостаточности компонентов (или групп компонентов) технологии, т.е. способности их функционировать отдельно от других компонентов без утраты целесообразности их использования;
- обеспечение антропоморфности, и как следствие, доступности широкому кругу пользователей принципов и методов, рассматриваемых в рамках технологии при решении задач;
- возрастание времени решения задачи при расширении функционала интеллектуального решателя;
- возрастание времени решения задачи при расширении базы знаний системы.

Рассмотрим задачи, детализирующие подход к преодолению описанных проблем:

- разработка четкой системы правил проектирования операций интеллектуального решателя;

- уточнение языка вопросов, используемого при взаимодействии операций с другими компонентами технологии, в том числе и с другими операциями;
- разработка систем операций, обеспечивающих поиск решения задачи по одной из известных стратегий решения или их комбинации;
- разработка систем операций, обеспечивающих поиск решения задачи по одному из известных правил логического вывода или их комбинации, в рамках некоторой стратегии решения;
- реализация разработанных операций и их отладка;
- разработка классификации операций решателя;

Более подробно приведенные задачи описаны ниже.

2. Аналоги и преимущества предлагаемой технологии

В качестве аналогов предлагаемой технологии компонентного проектирования интеллектуальных решателей задач можно назвать следующих представителей:

- 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)

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 сказалась нерешенность проблемы совмещения эффективной стратегии поиска с эффективным представлением знаний. [Ефимов, 1982]

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 QAZ от эвристического поиска. Таким образом, попытка построить дедуктивный решатель, используя в полной степени формализм принципа резолюции, оказалась, как показала система QAZ, также неуспешной. [Ефимов, 1982]

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]

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. [Кахро и др., 1988]

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, характеризующие 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]

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, 2010]

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 (semantic code programming) [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.

3. Состав технологии проектирования интеллектуальных решателей задач

В состав технологии входят следующие компоненты:

- модель интеллектуального решателя задач;
- библиотека ip-компонентов (intelligent property components – компонентов интеллектуальной собственности) решателя;
- система автоматизации проектирования;
- методика проектирования интеллектуальных решателей задач;
- help-системы поддержки проектирования интеллектуальных решателей задач;

- система управления коллективным проектированием интеллектуальных решателей задач.

Одним из достоинств семантической технологии компонентного проектирования интеллектуальных решателей задач является предметная независимость системы операций, используемых решателем, что позволяет не привязываться к конкретной предметной области.

Далеко не все задачи возможно решить путем применения классической дедуктивной логики. В связи с этим технология предполагает возможность применения различных логических подходов к решению задач в различных предметных областях.

Некоторые логические подходы, которые используются в технологии:

- классическая дедуктивная логика;
- методы индуктивного вывода;
- абдуктивный вывод;
- нечеткие логики;
- правдоподобные рассуждения;
- логика умолчаний;
- темпоральная логика.

Различные логические подходы позволяют проектировать решатели задач для интеллектуальных систем в разных предметных областях, учитывая их специфику.

4. Модель интеллектуального решателя задач

В предлагаемом подходе к преодолению приведенных проблем решатель задач рассматривается в неклассическом варианте. В данном случае решатель задач представляет собой графодинамическую sc-машину (память в качестве модели представления знаний использует семантическую сеть), состоящую из двух частей:

- графодинамической sc-памяти;
- систему sc-операций.

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 решателя путем добавления или удаления из его состава некоторого набора операций. Более подробно процесс проектирования операций и предъявляемые к ним требования рассмотрены в соответствующем разделе.

Отличительной особенностью решателя задач как многоагентной системы в рамках данного подхода является принцип взаимодействия операций-агентов. Агенты обмениваются сообщениями

исключительно через общую память путем использования соответствующего языка взаимодействия (языка вопросов-ответов), в отличие от большинства классических многоагентных систем, 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.

5. Библиотека универсальных совместимых ip-компонентов решателя

Центральным элементом семантической технологии проектирования интеллектуальных решателей задач является библиотека совместимых ip-компонентов.

Опишем структуру этой библиотеки.

- Библиотека готовых решателей задач:
 - решатель задач для экспертных систем, построенных на основе производственной модели представления знаний;
 - решатель задач по геометрии;
 - решатель задач по теории графов;
 - и другие.

- Библиотека стратегий решения задач:
 - разбиение задачи на подзадачи:
 - поиск критерия разбиения задачи;
 - разбиение задачи по заданному критерию;
 - соединение решений, полученных в результате разбиения;
 - удаление дублирующихся знаний;
 - и другие;
 - решение задачи с конца (стратегия обратного вывода)
 - унификация;
 - проверка противоречивости базы знаний;
 - соединение И-подцелей;
 - соединение ИЛИ-подцелей;
 - получение решения;
 - удаление промежуточных утверждений;
 - и другие;
 - упрощение задачи (переход от формулировки в терминах предметной области к формулировке на логическом языке):
 - операция обобщения;
 - вывод обобщенного логического высказывания;
 - фаззификация;
 - дефаззификация;
 - и другие;
 - случайный поиск и метод проб и ошибок (применимо в тех случаях, когда известно, что задача имеет небольшое число путей решения):
 - определения числа возможных путей решения;
 - применения пути решения;
 - оценка эффективности пути решения;
 - получения решения из всех путей;
 - и другие;
 - использование правил для решения типовых задач:
 - поиск подходящего класса задач для данной задачи;
 - решения задачи методами найденного класса задач;
 - поиск наиболее близкого класса задач для данной задачи;
 - и другие;
 - метод деления пополам:
 - разделение множества предполагаемых решений пополам;
 - принятие решения об отказе от одной из половин;
 - восстановление решения;
 - и другие;
 - применение аналогии:
 - генерация логического утверждения по аналогии;
 - восстановление решения после применения аналогии;
 - генерация фактов по аналогии;
 - и другие;
 - метод поиска в глубину:
 - операция поиска в глубину;
 - генерация знаний на заданной глубине поиска;
 - попытка решения задачи после генерации знаний;
 - восстановление шагов решения после применения поиска;
 - и другие;
- метод поиска в ширину:
 - операция поиска в ширину;
 - построение дерева решений поиска в ширину;
 - нахождения наиболее короткого решения (за наименьшее число шагов);
 - поддержка очереди фактов;
 - и другие;
- перебор вариантов решения:
 - генерация варианта решения;
 - проверка варианта решения;
 - выполнение «отката»;
 - применение решения;
 - восстановление решения;
 - и другие;
- генерация всех возможных следствий (прямой логический вывод):
 - поиск всех возможных правил для применения;
 - применение найденных правил;
 - проверка новых фактов на то, что они отсутствуют в базе знаний;
 - дополнение базы знаний сгенерированными фактами;
 - восстановление решения;
 - и другие;
- Библиотека операций:
 - логического вывода:
 - генерация знаний на основании определения (эквиваленции);
 - генерация знаний на основании продукции (импликации);
 - генерация определения на основании двух импликаций;
 - генерация более частного импликативного высказывания на основании более общего;
 - интерпретация арифметического выражения;
 - вывод обобщенного высказывания;
 - и другие;
 - поисковые операции:
 - операция поиска значения;
 - операция поиска формулы для нахождения значения искомой характеристики;
 - операция поиска доказательства;
 - операции других машин интеллектуального поиска;
 - и другие;
 - интерпретации хранимых способов решения задач:
 - алгоритмов;
 - процедурных программ;
 - логических программ;
 - нейронных сетей;
 - генетических алгоритмов;
 - методических указаний к решению задач;
 - и других;

- «сборки мусора»:
 - удаления всех шаблонов, по которым осуществлялся поиск;
 - удаления всех сгенерированных промежуточных логических утверждений;
 - и другие;
- мониторинга качества базы знаний:
 - устранение избыточности;
 - устранение противоречивости;
 - проверка полноты;
 - и другие.
- Библиотека базовых преобразований:
 - поиск изоморфной конструкции по образцу;
 - генерация изоморфной конструкции по образцу;
 - поиск всех выходящих из узла дуг;
 - генерация тройки (узел-дуга-узел);
 - генерация пятерки (узел-дуга с атрибутом-узел);
 - поиск пятерки (узел-дуга с атрибутом-узел);
 - и другие.
- Библиотека программ, реализованных на различных языках программирования и на различных платформах

В языке SCP часто используются такие платформенно независимые процедуры, как:

- **find_link_end** – процедура поиска второго компонента связки бинарного отношения, если первый компонент известен;
- **gen_meta_atom** – процедура, генерирующая конструкцию, изоморфную заданной, причем переменным элементам первой конструкции соответствуют метапеременные элементы второй. Константные элементы не изменяются;
- **merge** – процедура, выполняющая слияние двух узлов в памяти системы;
- и другие.
- Платформенно зависимые процедуры, написанные на языке C++:
 - **createElement** – процедура, выполняющая генерацию узла или дуги в sc-памяти;
 - **getContent** – процедура получения содержимого узла;
 - **createIterator** – создание итератора для просмотра элементов множества;
 - и другие.

6. Средства автоматизации проектирования интеллектуальных решателей задач

В состав семантической технологии так же входят и инструментальные средства, которые позволяют автоматизировать процесс создания интеллектуальных решателей задач. Эти средства позволяют работать на более высоком уровне абстракции, чем текст программы на языке SCP.

Перечислим основные возможности, которые имеет данное средство автоматизации:

- возможность включить (исключить) операцию из проектируемой системы операций интеллектуального решателя;
- верификация спроектированного набора операций;
- проверка операций на предметную независимость, универсальность;
- возможность отладки системы операций на конкретной базе знаний;
- профайлер производительности (отслеживание процессорного времени работы операций);
- профайлер памяти (отслеживание текущего состояния sc-памяти);
- просмотр стека вызовов (последовательность сгенерированных в памяти вопросов, операций, обрабатывающих данные вопросы и сгенерированные ответы на данные вопросы);

Таким образом, средства автоматизации позволяют разработчику интеллектуального решателя задач создавать наборы операций, которые реализовывают различные подходы к решению задач в рамках различных логических подходов.

7. Методика применения технологии при проектировании конкретных решателей задач

Технология проектирования интеллектуальных решателей задач основана на задачно-ориентированной методологии. В связи с этим проектирование системы операций состоит из четырех основных этапов:

- создание тестового сборника задач, которые решаются в рамках исследуемой предметной области;
- определение набора предметно независимых операций, которые будут использоваться при решении задач из тестового сборника;
- уточнение семантической спецификации каждой из указанных операций;
- реализация и отладка операций.

В общем случае можно выделить следующие предметно независимые классы задач

- Задачи синтеза доказательства
- Задачи верификации
- Задачи синтеза способа (алгоритма) решения
- Задачи анализа
- Задачи классификации

В качестве примера предметной области рассмотрим геометрию Евклида. Тогда классификация задач тестового сборника будет выглядеть следующим образом:

- по способу решения:
 - вычислительные задачи;
 - задачи на доказательство;
 - задачи на построение;
 - задачи на уточнение;
 - комбинированные задачи;
- по объекту решения:

- геометрические точки;
- прямые и отрезки;
- треугольники;
- многоугольники;
- окружности;
- и другие;
- по размерности пространства:
 - планиметрические;
 - стереометрические.

Семантическая спецификация операции представляет собой sc-конструкцию, которая описывает интерфейс взаимодействия проектируемой операции с другими операциями.

Данный интерфейс включает в себя:

- условие инициирования;
- узел, обозначающий sc-операцию;
- возможные результаты работы операции.

Формально спецификация операции описывается следующим образом:

• Сама sc-операция описывается предметным узлом, при этом указывается принадлежность операции множеству sc-операций.

• Связкой отношения «*условие инициирования**» операция соединяется с конструкцией, которая является условием инициирования данной операции

• Связкой отношения «*результат работы**» операция соединяется с логическим высказыванием, представляющим собой строгую дизъюнкцию атомарных высказываний (конструкций), каждое из которых может являться результатом работы данной операции, т.е. отражает изменения, произошедшие в памяти в результате работы данной операции.

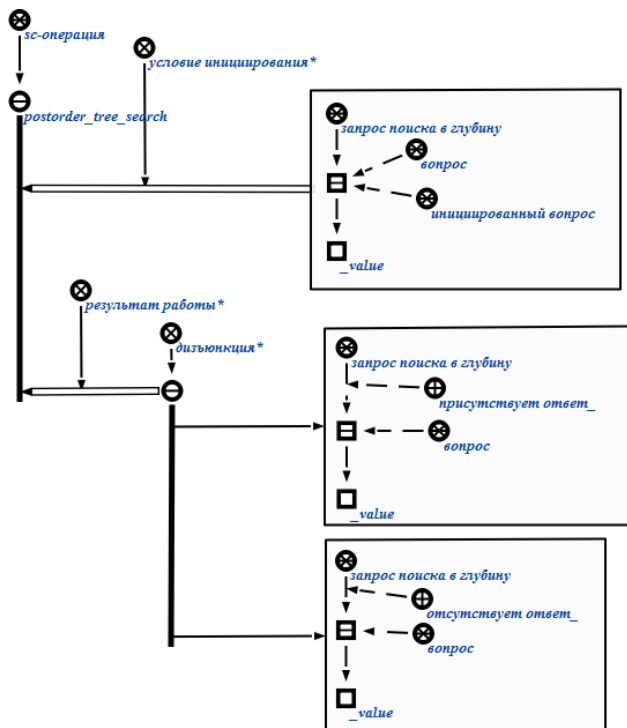


Рисунок 1 – Пример семантической спецификации операции поиска в глубину.

Реализацию операции условно можно разбить на 2 этапа:

- разработка алгоритма операции;
- реализация программы на специальном процедурном языке программирования SCP, предназначенном для обработки семантических сетей. При этом технология не ограничивает разработчика выбором именно данного языка для реализации, однако данный подход является наиболее оптимальным.

Тестирование sc-операций подразделяется на 3 этапа:

- подготовительный:
 - наполнение базы знаний (вносятся те знания, которые необходимы для работы операций);
- проверка совместимости:
 - проверка совместимости операции с прочими ip-компонентами системы. При этом ip-компонентами могут являться как определенные фрагменты базы знаний, так и другие операции, а также компоненты пользовательского интерфейса;
 - проверка правильности синхронизации с другими операциями;
- этап выявления и исправления ошибок:
 - локализация ошибки;
 - исправление ошибки.

8. Help-система для разработчиков интеллектуальных решателей

Семантическая технология предусматривает help-систему, которая позволяет обучать неподготовленного разработчика методике проектирования интеллектуальных решателей задач. Данная система представляет собой интеллектуальную справочную систему, построенную на базе открытой семантической технологии проекта OSTIS.

Help-система содержит знания о различных логических подходах и стратегиях, которые позволяют достигнуть решения той или иной задачи. Система помощи отслеживает действия пользователя и тем самым позволяет наилучшим образом подбирать подсказки и советы к каждому конкретному пользователю.

Help-систему по семантической технологии проектирования интеллектуальных решателей задач вполне можно считать полноценной справочной системой по логике. Что позволяет не только научить разработчика методике проектирования решателей, но и объяснить с формальной точки зрения процесс выбора операций, стратегий решения задач, логических подходов.

Help-система также содержит рекомендации по проектированию интеллектуальных решателей задач на основе имеющейся библиотеки, а также включает в себя подробную спецификацию указанной библиотеки и ее компонентов. Это позволяет считать рассматриваемую Help-систему одновременно руководством разработчика и пользователя.

9. Методика проектирования операций

Рассмотрим ряд принципов, соблюдение которых необходимо для корректности работы разрабатываемого решателя задач.

Каждая операция должна быть предметно независимой, т.е. в секции констант данной операции не должны быть описаны константы, имеющие отношение непосредственно к рассматриваемой предметной области. Исключение составляют понятия, которые могут использоваться в различных предметных областях (например, отношения «*включение**» и «*часть-целое**»). Данное правило может также быть нарушено в случае, если операция является вспомогательной и ориентирована на обработку какого-либо конкретного класса объектов (например, арифметические операции могут напрямую работать с конкретными отношениями «*сложение**» и «*умножение**» и т.п.). Всю необходимую для решения задачи информацию операция должна извлекать из семантической окрестности запроса.

Операция должна по возможности меньше ориентироваться на фиксированную форму представления фрагментов базы знаний, на работу с которыми она ориентирована. Степень глубины формализации и другие аспекты базы знаний определяются инженером по знаниям и не должны влиять на корректность работы операции. При обнаружении некорректности в представлении рассматриваемого фрагмента базы знаний операция, как и вся система не должна терять управления и, по возможности, сообщить пользователю о некорректности приведенных знаний.

Одна операция может состоять из ряда подпрограмм на выбранном языке программирования. Не стоит путать понятия «*операция*» и «*программа*» («*подпрограмма*»). Подпрограмма не является агентом и должна вызываться другой подпрограммой. Операция представляет собой как минимум одну подпрограмму, которая имеет особый формат входных и выходных данных, автоматически реагирует на состояние sc-памяти и при необходимости запускает другие подпрограммы.

При проектировании подпрограмм следует учитывать возможность использования различными операциями одних и тех подпрограмм. Таким образом, появляется необходимость говорить не только о библиотеке sc-операций, но и библиотеках подпрограмм на различных языках программирования, например библиотеке sc-подпрограмм.

Каждая операция должна самостоятельно проверять полноту соответствия условия иницирования конструкции, имеющейся в памяти системы на данный момент. В процессе решения задачи может возникнуть ситуация, когда на появление одной и той же конструкции среагировали несколько операций. В таком случае выполнение продолжает только та операция, условие иницирования которой полностью соответствует сложившейся

ситуации. Остальные операции обязаны в данном случае прекратить выполнение.

Выполнение предыдущего пункта достигается за счет тщательного уточнения спецификаций разрабатываемых операций. В общем случае условия иницирования у нескольких операций может совпадать, однако такая ситуация является очень нежелательной и может быть реализована в том случае, если операции не вносят критических изменений в ту область памяти, с которой работают остальные операции.

При проектировании систем операций интеллектуального решателя рекомендуется по возможности использовать операции, уже имеющиеся в библиотеке операций [1]. При необходимости реализации новой операции следует проектировать ее по возможности более общей, однако необходимо выделять в отдельные операции фрагменты рассуждений, которые могут быть использованы отдельно при решении другого класса задач.

Если в процессе работы операция генерирует в памяти какие-либо временные конструкции, то при завершении работы она обязана удалять всю информацию, использование которой в системе более нецелесообразно. Исключение составляют ситуации, когда подобная информация необходима нескольким операциям для решения одной задачи, однако после решения задачи информация становится бесполезной или избыточной и требует удаления. В данном случае ни одна из операций может оказаться не в состоянии удалить информационный мусор. В таком случае возникает необходимость говорить о специализированных вспомогательных операциях, задачей которых является уничтожение информационного мусора.

Операции необходимо объединять в группы для решения многоходовых задач, т.е. таких задач, для решения которых недостаточно всего одной операции. Очевидно, что под данное определение попадает большинство задач из практически любой предметной области. Группа операций является в некотором смысле самостоятельной подсистемой в рамках целостной системы операций.

При объединении операций в группы рекомендуется проектировать операции таким образом, чтобы они могли быть использованы не только в рассматриваемой группе. В случае, если это не представляется возможным и некоторые операции, будучи отделенными от группы, теряют смысл, необходимо указать данный факт при документировании рассматриваемых операций.

Инициатором запуска операции может быть как непосредственно пользователь системы, так и другая операция. При этом это никак не должно отражаться в работе самой операции. Необходимость вывода (трансляции) какого-либо фрагмента памяти пользователя отслеживается компонентами пользовательского интерфейса.

Язык взаимодействия операций через sc-память представляет собой подмножество языка вопросов. При расширении языка вопросов для введения какой-либо новой операции необходимо

максимально сокращать sc-конструкцию, представляющую собой вопрос. В вопросе должна указываться только критически важная информация, все остальные знания операция должна находить самостоятельно в семантической окрестности вопроса. Это позволяет уменьшить сложность восприятия интерфейса операции потенциальным пользователем и унифицировать форматы вопросов у большого числа различных операций.

При выделении в алгоритме решения задачи отдельных операций необходимо учитывать два фактора:

- операции должны быть как можно более универсальными, т.е. использоваться при решении как можно большего числа задач, что позволит избежать повторной реализации одних и тех же фрагментов рассуждений и уменьшит избыточность знаний в системе;
- операции должны быть по возможности антропоморфными, т.е. одна операция должна моделировать некий единый законченный акт мыслительной деятельности человека. Не следует искусственно увязывать ряд действий в одну операцию и наоборот, расчленять одно самостоятельное действие на поддействия. Это вызовет сложности восприятия принципов работы операции разработчиками и не позволит использовать операцию в ряде систем (например, в обучающих системах, которые должны объяснять ход решения пользователю);

Таким образом, в процессе разработке системы операций можно выделить следующие этапы:

- определение необходимого набора операций (с учетом уже имеющихся в библиотеке);
- определение ключевых узлов языка вопросов для связи операций через графодинамическую память;
- составление спецификаций каждой из операций;
- реализация и тестирование операций;

10. Пример описания алгоритма выполнения операции

В качестве примера, описывающего особенности выполнения алгоритма, была выбрана операция получения значения продукции. Данная операция является одной из ключевых в системе операций прямого логического вывода.

Описание алгоритма работы операции:

1. Ищем узел связки конкретной теории.

1.1. Если узел связки конкретной теории не найден, то переходим к шагу 2.

1.2. Если узел связки конкретной теории найден, то получаем объект из запроса продукции.

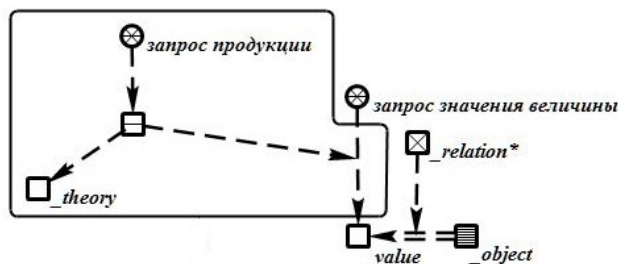


Рисунок 2 – Пояснение шага 1.2.

1.2.1. Если объект из запроса продукции не найден, то переходим к шагу 2.

1.2.2. Если объект из запроса продукции найден, то получаем условие (в конкретной теории находится под атрибутом 1_) и следствие (в конкретной теории находится под атрибутом 2_) из конкретной теории.

1.2.2.1. Если условие и следствие не были получены, то переходим к шагу 2.

1.2.2.2. Если условие и следствие получены, то получаем связанные узлы конкретной теории.

1.2.2.2.1. Если связанные узлы конкретной теории не получены, то переходим к шагу 2.

1.2.2.2.2. Если связанные узлы конкретной теории не получены, то происходит поиск фрагмента база знаний по шаблону (условие теории) с сохранением множества пар результатов поиска.

1.2.2.2.3. Среди множества пар соответствия из шага 1.2.2.1.2. определяем ту, где под атрибутом 2_ имеем объект из шага 1.3.

1.2.2.2.4. Из полученной пары из шага 1.2.2.2.3. получаем узел под атрибутом 1_.

1.2.2.2.5. Ищем конкретный фрагмент базы знаний для объекта из шага 1.2. с сохранением множества пар результатов поиска.

1.2.2.2.6. Формируем множество связок, где под атрибутом 1_ находится связанная переменная из условия теории, которая была найдена в шаге 1.2.2., а под атрибутом 2_ находится соответствующий узел из базы знаний.

1.2.2.2.7. Для найденных пар из шага 1.2.2.2.5. ищем, где под атрибутом 1_ находятся связанные переменные из шага 1.2.2. и добавляем во множество связок из шага 1.2.2.2.6. связку, где под атрибутом 1_ находится узел из выше найденной пары под атрибутом 1_, под атрибутом 2_ находится узел из найденной пары под атрибутом 2_.

1.2.2.2.8. Генерируем по следствию из теории константный контур, где заменяем связанные переменные из шага 1.2.2. на конкретные с помощью сформированного множества пар из шагов 1.2.2.2.6. и 1.2.2.2.7.

1.2.2.2.9. Если множество связок к шагу 1.2.2.2.7. не сформировано, то переходим к шагу 2.

1.2.2.2.10. Генерируем условие успешного завершения операции.

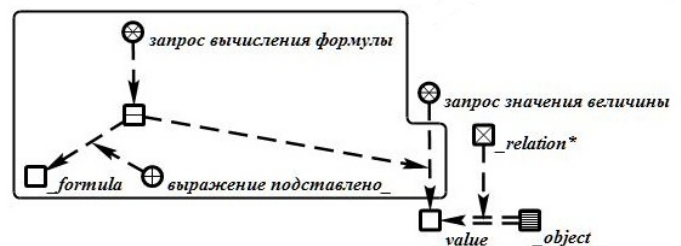


Рисунок 3 – Результат успешного завершения работы операции.

1.2.2.2.11. Переходим к шагу 3.

2. Генерируем условие неудачного завершения операции.

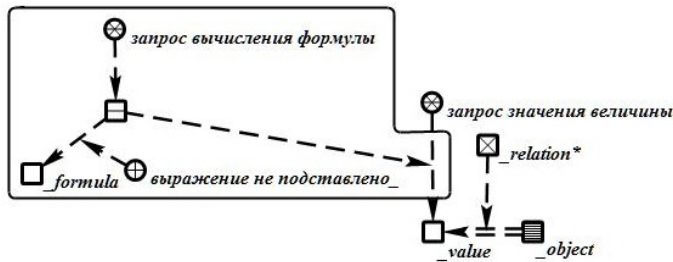


Рисунок 4 – Результат неудачного завершения работы операции.

3. Завершение работы операции.

11. Описание существующей реализации технологии

Рассмотрим более подробно текущую версию реализации предлагаемой технологии на примере разработанного группой авторов интеллектуального решателя задач. Как следует из самой концепции технологии, разработанный решатель является независимым от предметной области и представляет собой взаимосвязанную систему операций, работающих над общей памятью.

На настоящий момент интеллектуальный решатель реализует смешанную стратегию решения, представляющую собой комбинацию традиционной продукционной модели и метода поиска в глубину.

Уровень логического вывода на данный момент представлен возможностью вывода на основе продукций вида «если...» - «то...» по правилу *modus ponens*. При этом возможен вариант преобразования найденной продукции в более частую форму, если это возможно с точки зрения исходных данных и не противоречит логической структуре утверждения. Также интеллектуальный решатель способен оперировать утверждениями, представленными в виде эквиваленции, такими как определения и утверждения о необходимости и достаточности. В указанном случае утверждение может быть применено как продукция, где в качестве посылки и заключения используются атомарные высказывания, входящие в указанное утверждение об эквиваленции. Роли данных высказываний определяются в зависимости от контекста в каждом конкретном случае.

Процесс решения задачи в текущей реализации можно разделить на следующие этапы:

- Этап работы поисковых операций.

Вне зависимости от типа задачи 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, решатель

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. На данный момент, как уже было сказано, интеллектуальный решатель реализует комбинированную стратегию.

Вначале рассматривается некоторый объект, для которого осуществляется поиск всех классов объектов, которым он принадлежит. Далее для каждого класса осуществляется поиск утверждений, справедливых для данного класса объектов (с целью оптимизации на настоящий момент данный факт должен быть явно указан проектировщиком базы знаний). При рассмотрении каждого утверждения осуществляется попытка применить его в рамках некоторой семантической окрестности рассматриваемого объекта, 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). На данный момент реализован логический вывод на основе правила *modus ponens*. В будущем предполагается расширить набор подобных правил с целью увеличения количества различных типов утверждений, интерпретируемых решателем. 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 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

- формула вычисления радиуса вписанной в треугольник окружности как отношения площади треугольника к его полупериметру;

Далее представлено формальное описание условия задачи, а также формальное описание некоторых фрагментов базы знаний. Полное описание всех фрагментов базы знаний можно найти на сайте OSTIS.

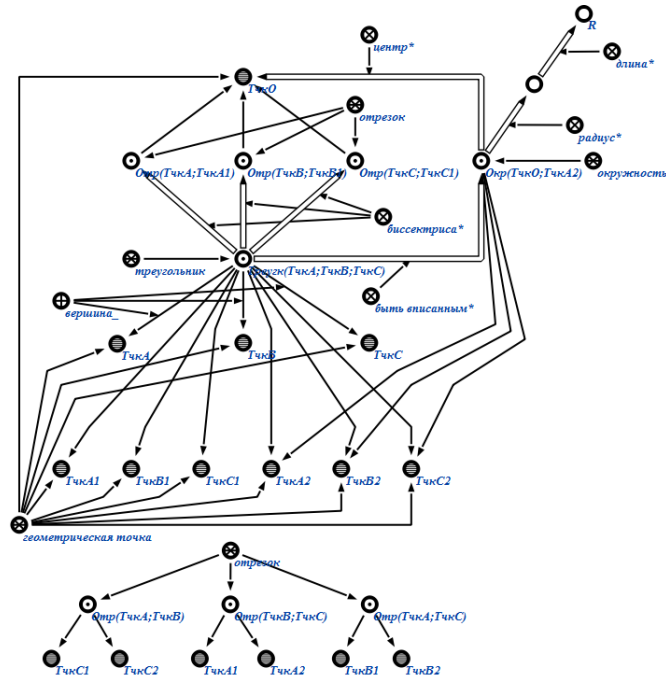


Рисунок 6 – Формальное описание условия задачи (фрагмент 1)

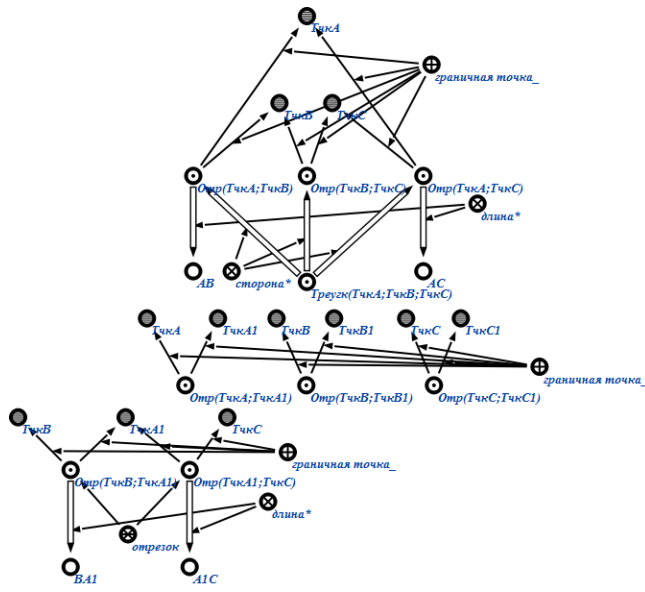


Рисунок 7 – Формальное описание условия задачи (фрагмент 2)

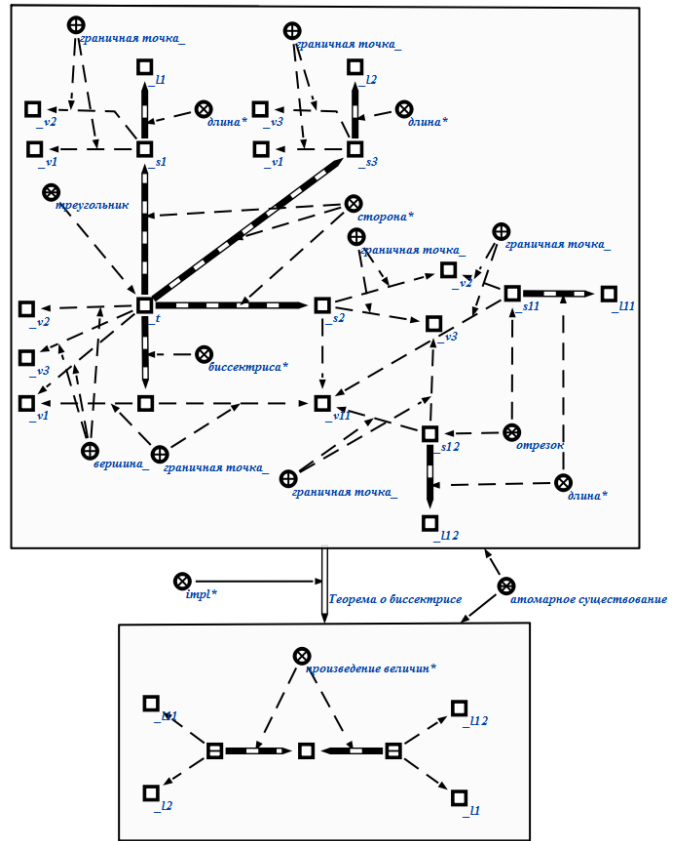


Рисунок 8 – Теорема о биссектрисе

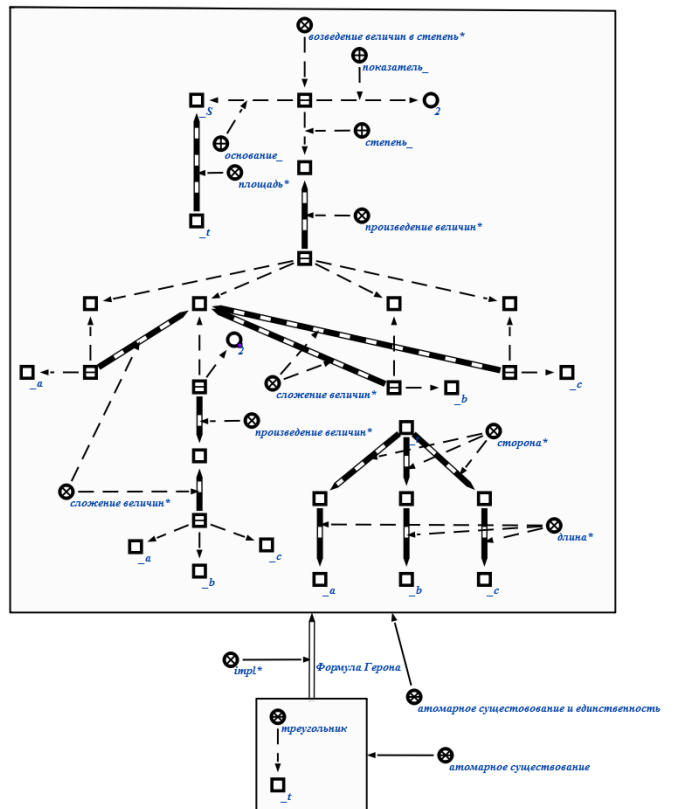


Рисунок 9 – Формула Герона

Для инициирования требуемого набора операций необходимо создать в памяти вопросную ситуацию:

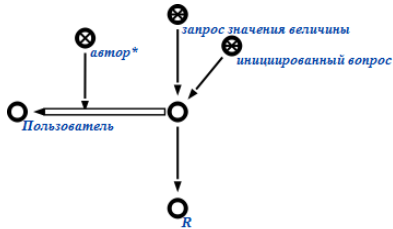


Рисунок 10 – Формат вопроса

Опишем краткий протокол решения задачи по шагам. Полную версию протокола решения данной задачи можно найти на сайте OSTIS.

• Шаг 1

Используемая операция

Операция поиска значения (find_value)

Пояснение

Операция пытается найти уже имеющееся значение требуемой величины

Требуемое значение отсутствует

Условие инициализации

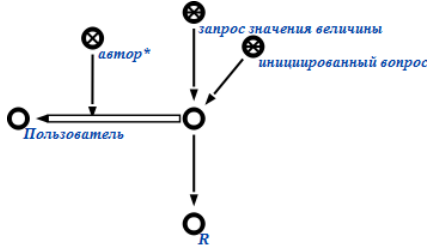


Рисунок 11 – Условие инициализации операции на шаге 1

Результат работы

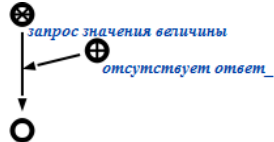


Рисунок 12 – Результат работы операции на шаге 1

• Шаг 2

Используемая операция

Операция, организующая поиск в глубину (postorder_tree_search_manager)

Пояснение

Операция организует запуск рекурсивной операции поиска в глубину.

Для этого на рассматриваемый объект устанавливается запрос поиска в глубину

Условие инициализации

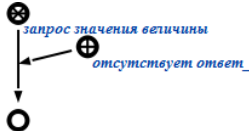


Рисунок 13 – Условие инициализации операции на шаге 2

Результат работы

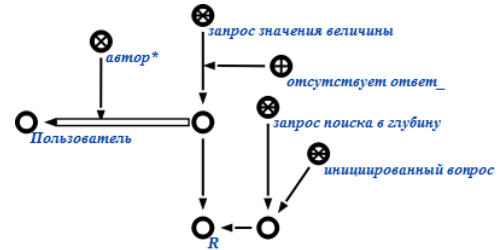


Рисунок 14 – Результат работы операции на шаге 2

• Шаг 3

Используемая операция

Операция поиска в глубину (postorder_tree_search)

Пояснение

Операция просматривает все объекты, связанные с исходным объектом и пытается сгенерировать новые знания. Если знания сгенерировать не удалось, запрос поиска в глубину устанавливается на объекты, связанные с данным. Просмотренные узлы добавляются в множество просмотренных узлов. В данном случае новые знания генерируются для объекта Треугол(ТчкА;ТчкВ;ТчкС). При этом используется утверждение «Теорема о биссектрисе».

Условие инициализации

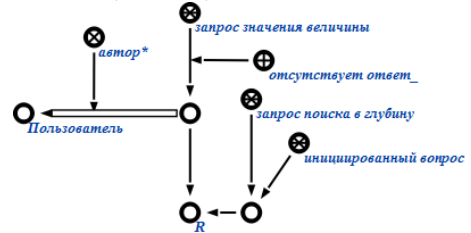


Рисунок 15 – Условие инициализации операции на шаге 3

Результат работы

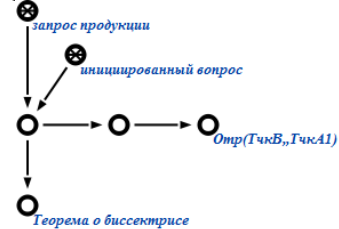


Рисунок 16 – Результат работы операции на шаге 3

• Шаг 4

Используемая операция

Операция получения значения продукции (find_value_production)

Операция интерпретации арифметического выражения (calculation)

Пояснение

На основании теоремы о биссектрисе вычисляется длина отрезка Отр(ТчкВ;ТчкА1) – бсм.

Условие инициализации

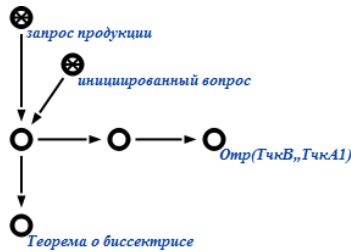


Рисунок 17 – Условие инициализации операции на шаге 4

Результат работы

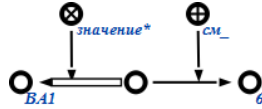


Рисунок 18 – Результат работы операции на шаге 4

- Шаг 5 Аналогичен шагу 1
Используемая операция
Операция поиска значения (find_value)
- Шаг 6 Аналогичен шагу 2
Используемая операция
Операция, организующая поиск в глубину (postorder_tree_search_manager)
- Шаг 7 Аналогичен шагу 3
Используемая операция
Операция поиска в глубину (postorder_tree_search)
Пояснение
В данном случае новые знания генерируются для объекта $Отр(ТчкВ;ТчкС)$. При этом используется утверждение «Формула вычисления длины отрезка»
- Шаг 8
Используемая операция
Операция получения значения продукции (find_value_production)
Операция интерпретации арифметического выражения (calculation)
Пояснение
Вычисляется длина отрезка $Отр(ТчкВ;ТчкС)$ как сумма длин отрезков $Отр(ТчкВ;ТчкА1)$ и $Отр(ТчкА1;ТчкС)$ – 11 см.

Результат работы

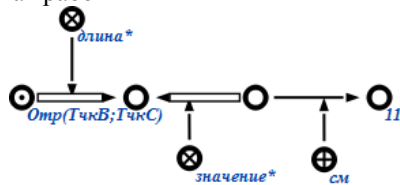


Рисунок 19 – Результат работы операции на шаге 8

- Шаг 9 Аналогичен шагу 1
Используемая операция
Операция поиска значения (find_value)
- Шаг 10 Аналогичен шагу 2
Используемая операция
Операция, организующая поиск в глубину (postorder_tree_search_manager)
- Шаг 11 Аналогичен шагу 3
Используемая операция
Операция поиска в глубину (postorder_tree_search)

Пояснение

В данном случае новые знания генерируются для объекта $Треугк(ТчкА;ТчкВ;ТчкС)$. При этом используется утверждение «Формула вычисления периметра треугольника»

- Шаг 12

Используемая операция

Операция получения значения продукции (find_value_production)

Операция интерпретации арифметического выражения (calculation)

Пояснение

Вычисляется периметр треугольника $Треугк(ТчкА;ТчкВ;ТчкС)$ – 33 см.

Результат работы

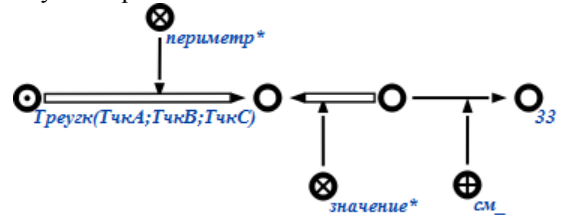


Рисунок 20 – Результат работы операции на шаге 12

- Шаг 13 Аналогичен шагу 1
Используемая операция
Операция поиска значения (find_value)
- Шаг 14 Аналогичен шагу 2
Используемая операция
Операция, организующая поиск в глубину (postorder_tree_search_manager)
- Шаг 15 Аналогичен шагу 3
Используемая операция
Операция поиска в глубину (postorder_tree_search)
Пояснение
В данном случае новые знания генерируются для объекта $Треугк(ТчкА;ТчкВ;ТчкС)$. При этом используется утверждение «Формула Герона»
- Шаг 16
Используемая операция
Операция получения значения продукции (find_value_production)
Операция интерпретации арифметического выражения (calculation)
Пояснение
Вычисляется площадь треугольника $Треугк(ТчкА;ТчкВ;ТчкС)$ – 51,52 см².

Результат работы

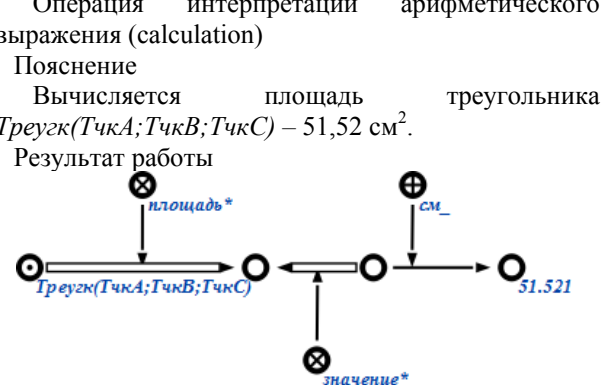


Рисунок 21 – Результат работы операции на шаге 16

- Шаг 17 Аналогичен шагу 1
Используемая операция
Операция поиска значения (find_value)
- Шаг 18 Аналогичен шагу 2

Используемая операция

Операция, организующая поиск в глубину (postorder_tree_search_manager)

- Шаг 19 Аналогичен шагу 3

Используемая операция

Операция поиска в глубину (postorder_tree_search)

Пояснение

В данном случае новые знания генерируются для объекта $Окр(ТчкО; ТчкА2)$. При этом используется утверждение «Соотношение площади треугольника и радиуса вписанной окружности».

- Шаг 20

Используемая операция

Операция получения значения продукции (find_value_production)

Операция интерпретации арифметического выражения (calculation)

Пояснение

Вычисляется радиус окружности $Окр(ТчкО; ТчкА2) - 3,1225$ см.

Результат работы

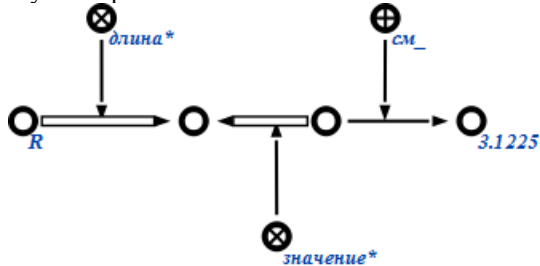


Рисунок 22 – Результат работы операции на шаге 20

- Шаг 21

Используемая операция

Операция поиска значения (find_value)

Пояснение

Операция пытается найти уже имеющееся значение требуемой величины

Требуемое значение присутствует

Результат работы

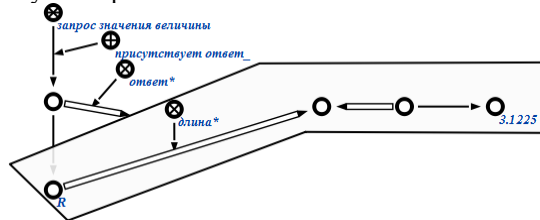


Рисунок 23 – Результат работы операции на шаге 21

Выбор операций, необходимых для решения задач в каждой конкретной прикладной интеллектуальной системе, определяется разработчиком интеллектуального решателя. В связи с этим некоторые операции, необходимые в одной предметной области будут избыточными в другой.

Например, операции нечеткого и правдоподобного вывода будут очень полезны в системах, где имеется много критериев для принятия решения, анализируется множество характеристик, которые просто невозможно описать с точки зрения однозначной истинности или ложности.

Эти же операции в геометрии Евклида, напротив, будут избыточно, т.к. решение задач осуществляется только по правилам классического вывода (дедуктивного, обратного и т.д.).

ЗАКЛЮЧЕНИЕ

Таким образом, семантическая технология проектирования интеллектуальных решателей задач позволяет проектировать такие системы предметно независимых операций, которые способны решать унифицированным образом множество различных задач одного класса. Каждая из операций представляет собой ip-компонент, который может быть использован в других прикладных системах. Многоагентная модель позволяет легко осуществлять интеграцию компонентов машин обработки знаний при условии корректной интеграции баз знаний. При этом никаких дополнительных действий по интеграции машин обработки знаний не требуется, т.к. грамотно разработанная операция многоагентной модели самостоятельно контролирует условие инициирования и текущее состояние памяти.

Основным практическим результатом изложенной работы является успешная попытка привести одну из существующих моделей решения задач к описанному выше формализму. Как показали результаты, данная задача вполне решаема. В процессе работы над проблемой был создан прототип универсального решателя задач, обладающий достаточно широким функционалом. Полученные результаты позволяют сделать предположение об истинности выдвинутой в разделе 1 гипотезы.

Основная цель подобной работы - позволить даже относительно неподготовленному (в области проектирования интеллектуальных систем) человеку создать интеллектуальную справочную систему по интересующей его предметной области, обладающую гибким функционалом, который определяется разработчиком на стадии проектирования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Бенерджи, 1972] Бенерджи, Р. Теория решения задач / Р. Бенерджи - М., Изд-во «Мир» 1972.

[Вагин, 1988] Дедукция и обобщение в системах принятия решений / Вагин В.Н.; - М. :Изд-во «НАУКА», 1988.

[Вагин и др., 2008] Достоверный и правдоподобный вывод в интеллектуальных системах / Вагин В.Н. [и др.]; - М. : ФИЗМАТЛИТ, 2008.

[Вагин и др., 2010] Алгоритм абдуктивного вывода с использованием систем поддержки истинности на основе предположений / Вагин В.Н., Хотимчук К.Ю.; // Двенадцатая национальная конференция по искусственному интеллекту с международным участием. Труды конференции; - М. : ФИЗМАТЛИТ, 2010.

[Виноградов, 2010] Логика умолчаний как альтернатива модификационных исчислений / Виноградов Д.В.; // Двенадцатая национальная конференция по искусственному интеллекту с международным участием. Труды конференции; - М. : ФИЗМАТЛИТ, 2010.

[Владимиров и др., 2010] Владимиров А.Н., Варламов О.О., Носов А.В., Потапова 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.

[Гладун, 1987] Гладун В.П. Планирование решений / В.П. Гладун; – Киев. : Изд-во «Наукова думка», 1987.

[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.

[Градштейн, 1965] Градштейн, И.С. Прямая и обратная теоремы / И.С. Градштейн; - М. : Наука, 1965.

[Грунский и др., 2010] Алгоритмы резолюции в логике высказываний при 0-1-ном представлении дизъюнктов / Грунский И.С., Волченко М.В.; // Двенадцатая национальная конференция по искусственному интеллекту с международным участием. Труды конференции; - М. : ФИЗМАТЛИТ, 2010.

[Гуц, 2003] Гуц, А.К. Математическая логика и теория алгоритмов/ А.К. Гуц ; - Омск : «Наследие. Диалог-Сибирь», 2003.

[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. Тыгу;– М., Изд-во «Финансы и статистика», 1988.

[Кулик, 2001] Кулик, Б. А. Логика естественных рассуждений / Б. А. Кулик; - СПб.: Изд-во «Невский диалект», 2001.

[Лакатос, 1967] Лакатос, И. Доказательство и опровержение / И. Лакатос; - М. : Наука, 1967. Пер. с английского.

[Маслов, 1986] Маслов, С.Ю. Теория дедуктивных систем и ее применения / С.Ю. Маслов; - М. : «Радио и связь», 1986.

[Найденова, 2010] Найденова К.А. Принципы организации правдоподобных рассуждений в интеллектуальных системах / Найденова К.А. // Двенадцатая национальная конференция по искусственному интеллекту с международным участием. Труды конференции; - М. : ФИЗМАТЛИТ, 2010.

[Непейвода, 2000] Непейвода Н.Н. Прикладная логика. Учебное пособие/ Непейвода Н.Н.; – Новосибирск. :НГУ, 2000.

[Нильсон, 1973] Нильсон Н. Искусственный интеллект. Методы поиска решений. Нильсон Н.; – М. :Мир, 1973.

[Нильсон, 1985] Нильсон Н. Принципы искусственного интеллекта. Н. Нильсон; – М. : «Радио и связь», 1985.

[Поспелов, 1981] Поспелов Д.А. Логико-лингвистические модели в системах управления / Д.А.Поспелов; – М. :Изд-во «Энергоиздат», 1981.

[Поспелов, 1989] Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов / Д.А.Поспелов; – М. :Изд-во «Радио и связь», 1989.

[Пойа, 1975] Пойа Д. Математика и правдоподобные рассуждения / Пойа Д.; – М. :Изд-во «НАУКА», 1975.

[Пойа, 1976] Пойа Д. Математическое открытие / Пойа Д.; – М. :Изд-во «НАУКА», 1976.

[Плесневич, 2006] Плесневич Г.С. Силлогистика для семантических сетей / Плесневич Г.С.; // Десятая национальная конференция по искусственному интеллекту с международным участием. Труды конференции; - М. : ФИЗМАТЛИТ, 2006, т.1. – с. 321-330.

[Плесневич, 2010] Плесневич Г.С. Нечеткая аристотелева логика / Плесневич Г.С.; // Двенадцатая национальная конференция по искусственному интеллекту с международным участием. Труды конференции; - М. : ФИЗМАТЛИТ, 2010.

[Рассел, Норвиг 2006] Рассел С., Норвиг П. Искусственный интеллект. Современный подход / Рассел С., Норвиг П. ; - М. : Вильямс, 2006.

[Розен, 1982] Розен В.В. Цель – оптимальность – решение / В.В. Розен; – М. :Изд-во «Радио и связь», 1982.

[Смирнов, 1972] Смирнов, В.А. Формальный вывод и логические исчисления / В.А. Смирнов; – М. : Изд-во «Наука», 1972

[Стефанюк, 2004] Стефанюк, В.Л. Локальная организация интеллектуальных систем / В.Л.Стефанюк; – М.: ФИЗМАТЛИТ, 2004.

[Столяр, 1965] Столяр А.А. Логические проблемы преподавания математики / Столяр А.А.; – Минск. : Изд-во «Вышэйшая школа», 1965.

[Столяр, 1971] Столяр А.А. Логическое введение в математику / Столяр А.А.; – Минск. :Изд-во «Вышэйшая школа», 1971.

SEMANTIC TECHNOLOGY OF COMPONENTIAL DESIGNING OF INTELLECTUAL PROBLEM SOLVERS

Shunkevich D.V., Zalivako S.S.

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

shu.dv@tut.by

zalivako@mail.ru

[Столяр, 1982] Столяр А.А. Как математика ум в порядок приводит / Столяр А.А.; – Минск. :Изд-во «Вышэйшая школа», 1982.

[Столяр, 1987] Столяр А.А. Зачем и как мы доказываем в математике / Столяр А.А.; – Минск. :Изд-во «Народная асвета», 1987.

[Столяр, 1991] Математическая логика: Учеб. пособие / Л.А.Латонин, Ю.А.Макаренков, В.В.Николаева, А.А.Столяр. Под общ.ред. А.А.Столяра.; – Минск. :Изд-во «Вышэйшая школа», 1991.

[Тарасов, 2002] Тарасов, В.Б. От многоагентных систем к интеллектуальным организациям / В.Б. Тарасов; – М. :Изд-во УРСС, 2002.

[Теи и др., 1990] Логический подход к искусственному интеллекту / А. Теи, П. Грибомон, Ж. Луи, Д. Снийерс и др.; – М. :Изд-во «Мир», 1990.

[Туманов, 1969] Туманов, С.И. Поиск решения задач / С. И. Туманов; – М. :Изд-во «Просвещение», 1969.

[Тыгу, 1984] Тыгу, Э.Х. Концептуальное программирование / Э.Х. Тыгу;– М. :Изд-во «Наука», 1984.

[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.

[Финн, 2008] Финн, В.К. Многозначные логики и их применение. Логические исчисления, алгебры и функциональные свойства / В.К. Финн. Том 1. М.: УРСС, 2008.

[Финн, 2008] Финн, В.К. Многозначные логики и их применения: Логика в системах искусственного интеллекта. / В.К. Финн. Том 2. М.: УРСС, 2008.

[Финн, 2010] Финн, В.К. Индуктивные методы милевского типа в системах искусственного интеллекта / Финн В.К.; // Двенадцатая национальная конференция по искусственному интеллекту с международным участием. Труды конференции; - М. : ФИЗМАТЛИТ, 2010.

[Финн, 2011] Финн, В.К. Искусственный интеллект. Методология, применение, философия / В.К. Финн. М.: Изд-во «Красанд», 2011.

[Эрдниев, 1998] Эрдниев, О.П. От задачи к задаче – по аналогии / Эрдниев О.П.; - Элиста : Калмыцкий государственный университет, изд. «Столетие», 2010.

[Kowalsky, 1975] Kowalsky, R. A. Proof procedure using connection graphs / R. A. Kowalsky // Journal of the ACM. – 1975 – № 22(4).

[OSTIS, 2011] Проект OSTIS [Электронный ресурс]. Минск, 2011. – Режим доступа: <http://ostis.net/>. – Дата доступа: 20.11.2011.

In this work there is a describing of open semantic technology of componential designing of intellectual problem solver. Single attention it is given to the designing technique of solvers and the operations which are components of such solvers. Also in this work there are several examples of technology using during the designing of concrete intellectual systems in different object domains.

INTRODUCTION

Now transition from orientation of designers of intellectual systems from the imposed (offered) knowledge processing machine on designing problem solvers from ready components has the big actuality. This approach is based on common principles of organization of knowledge processing machines, which allows performing integration of different problem solution models, both existing and new. This gives an opportunity of realization of given technology on the base of any problem solution model.

In this work it is represented the methodology of intellectual problem solvers designing, which is the part of complex open technology of intelligent systems designing OSTIS.

MAIN PART

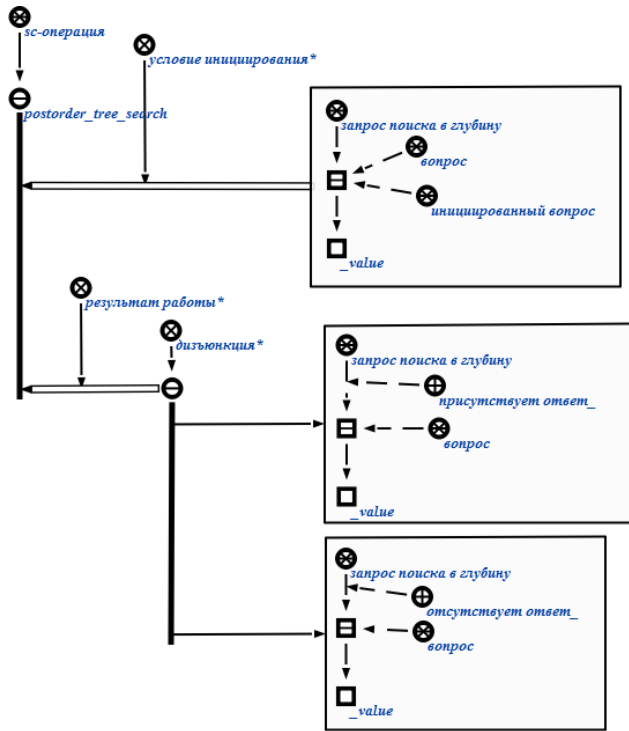
Here we will show the example of the semantic specification of the operation.

Semantic specification represents sc-construction, which describes an interface of interaction between the operation and other operations inside the system.

Interface includes:

- Initialization condition
- Node, designating the sc-operation
- Potential results of operation working

Formally, an operation is described in the next way:



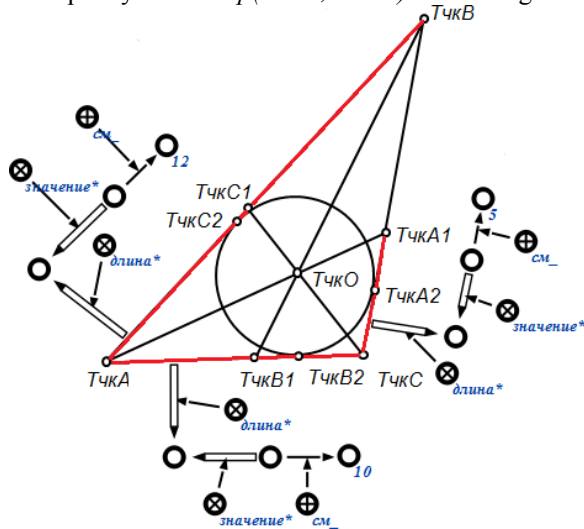
Drawing 24 – Semantic specification example.
Postorder tree search operation.

Next, we will show an example of problem solution.
Problem situation:

- In triangle $Треугол(ТчкА;ТчкВ;ТчкС)$ there are three bisectrices $Отр(ТчкА;ТчкА1)$, $Отр(ТчкВ;ТчкВ1)$ and $Отр(ТчкС;ТчкС1)$.
- In triangle $Треугол(ТчкА;ТчкВ;ТчкС)$ circle $Окр(ТчкО;ТчкА2)$ is inscribed.
- Circle $Окр(ТчкО;ТчкА2)$ and triangle $Треугол(ТчкА;ТчкВ;ТчкС)$ have common points $ТчкА2$, $ТчкВ2$, $ТчкС2$.
- Segment $Отр(ТчкА;ТчкВ)$ length is 12 sm.
- Segment $Отр(ТчкА;ТчкС)$ length is 10 sm.
- Segment $Отр(ТчкА1;ТчкС)$ length is 5 sm.

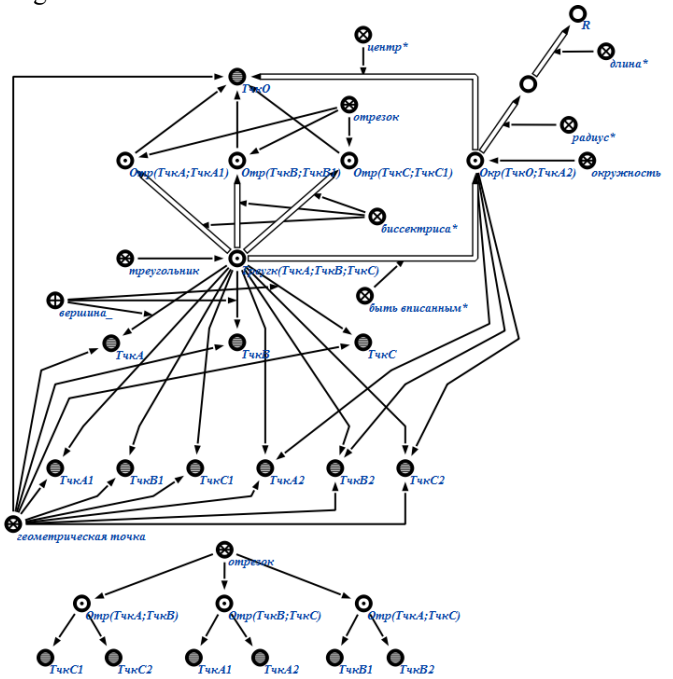
Problem:

Specify circle $Окр(ТчкО;ТчкА2)$ radius length.

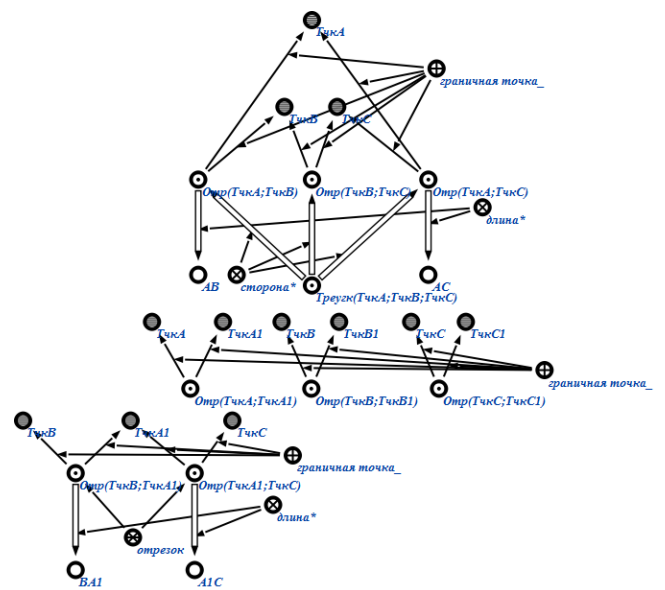


Drawing 25 – Problem illustration

Next there is represented formal problem situation description. Full description of all knowledge base fragments can be found on the OSTIS site.



Drawing 26 – Formal problem situation description
(fragment 1)



Drawing 27 – Formal problem situation description
(fragment 1)

CONCLUSION

Thereby, semantic technology of intellectual problem solvers designing allows to design such a systems of domain-independent operations, which are able to solve a number of problems from one class in unified way.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.81:159.942.52

ДИНАМИЧЕСКИЕ МОДЕЛИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ АДАПТИВНОГО РЕЗОНАНСА

Комарцова Л.Г., Лавренков Ю.Н., Антипова О.В., Кадников Д.С.

*Московский государственный технический университет им. Н.Э.Баумана
(Калужский филиал), г. Калуга, Россия*

lkomartsova@yandex .ru

polikarp1@rambler.ru

В статье рассматриваются проблемы создания динамических систем искусственного интеллекта. Исследуются вопросы отображения изменения параметров динамических процессов во времени, выделены особенности и признаки их характеризующие. Решается задача создания динамических моделей, обеспечивающих возможность функционирование подобных систем на основе модифицированной нейронной сети адаптивного резонанса.

Ключевые слова: эволюционные процессы, динамические системы искусственного интеллекта, нейросетевые модели, нейросетевая база знаний, нейронная сеть адаптивного резонанса.

ВВЕДЕНИЕ

Развитие систем искусственного интеллекта (СИИ) в настоящее время связано с созданием новых методов и технологий, направленных на решение сложных проблем, таких, как предсказание, планирование, распознавание образов и т.д. в различных прикладных областях [Ларичев О.И. и др., 1998]. Многие из публикаций по этим проблемам, несмотря на появление работ по созданию систем реального времени [Еремеев А.П., 2004], основываются на предположении неизменности во времени предъявляемых СИИ данных, статичности внешней среды. Однако статические модели не могут быть применены к моделированию процессов, которые изменяются во времени, развиваются и эволюционируют.

Для эволюционирующих процессов необходимым является отслеживание в динамике их изменения во времени (т.е. функционирование в режиме on-line) и адаптации к внешней среде (например, движение робота в незнакомой среде). В работе исследуются некоторые вопросы, которые необходимо решать для создания эволюционных систем, моделирующих процедуру принятия решений человеком в сложных трудно формализуемых задачах в реальном времени.

В соответствии с определением, данным в [Grossberg S., 1987], эволюция (evolutio) означает необратимый процесс исторического изменения живого, формирование таких комбинаций признаков и свойств организма, которые ведут к возникновению у них адаптации к условиям внешней среды. Поскольку эволюционный процесс

развивается во времени его трудно моделировать, так как характеризующие его параметры либо являются неизвестными априорно, либо изменяются случайно. При построении моделей эволюции и создании интеллектуальных эволюционных коннекционистских систем (ИЭКОС) необходимо уметь отображать изменения параметров динамических процессов во времени и выделять особенности и признаки их характеризующие [Chin-Teng Lin, 1996]. Главной проблемой при этом является хорошо известная дилемма «стабильности-пластичности», наиболее известный принцип жизни, реализуемый в коннекционистских - нейросетевых вычислительных моделях [Люгер, 2005], примером таких моделей являются нейронные сети типа ART (Adaptive Resonance Theory). В этой связи объединение эволюции и коннекционистских систем в рамках одной вычислительной модели представляется наиболее перспективным направлением развития СИИ.

1. Архитектура ИЭКОС

Целенаправленное, устойчивое к флуктуациям поведение ИЭКОС определяется двумя основными факторами [Комарцова Л.Г., 2004]:

- цельный интеллект имеет распределенную структуру и возникает из взаимодействия множества простых, отдельных, агентских (модульных) интеллектов;

- развитие эволюционных моделей опирается на дарвиновский принцип естественного отбора как на основной механизм, формирующий поведение

отдельных агентов и обеспечивающий их адаптацию к внешней среде.

Перечислим основные особенности, которые должны быть реализованы в подобных ИЭКОС: 1) модульность; 2) открытость; 3) взаимодействие отдельных модулей при функционировании системы; 4) обучаемость и дообучаемость в процессе всего жизненного цикла существования системы; 5) адаптивность к решаемой задаче; 6) необходимость работы в режиме on-line.

На основе анализа особенностей, которыми должна обладать эволюционная система, предложена архитектура ИЭКОС, представляющая собой многомодульную и многоуровневую открытую конструкцию, содержащую следующие основные части:

1. Входной слой, обеспечивающий предварительную обработку информации (фильтрацию, нормирование, центрирование и т.д.). Число входов может меняться в зависимости от числа выделенных признаков, характеризующих поступающую в систему информацию.
2. Запоминающая часть (нейросетевые модули), представляющая собой нейросетевую базу знаний (НСБЗ) и формирующаяся на основе информации, поступающей из различных источников в процессе функционирования системы. Это многомодульная структура с эволюционными связями между модулями, объединенными в распределенные группы.
3. Слой принятия решений, состоящий из нескольких модулей, каждый из которых принимает решение по определенной проблеме. Модули имеют обратную связь с внешней средой и системой, осуществляющей адаптацию НСБЗ к решаемой задаче.
4. Выходной слой обеспечивает выдачу информации в заданном формате из слоя принятия решений во внешнюю среду.
5. Система адаптации, осуществляющая подстройку одного или нескольких модулей НС к решаемой задаче за счет их дообучения или включения новых нейронов в выходной слой (например, подобно тому, как это делается в сетях ART).
6. Знаниеориентированная подсистема, извлекающая информацию из НС и представляющая ее в виде правил IF-THEN. Наличие нечеткой базы правил обеспечивает объяснение результатов работы нейросетевых модулей.

Такую архитектуру можно рассматривать как теоретическую модель для построения реальных динамических систем. Несмотря на значительные достижения в области развития теории искусственного интеллекта, все еще не достигнут тот уровень «интеллектуальности», который свойственен человеку. Поэтому большое значение для практики имеет исследование методов построения эволюционных систем, которые могли бы эволюционировать и адаптироваться к

решаемым задачам, повышая тем самым уровень своего «интеллекта».

2. Функционирование ИЭКОС

Рассмотрим основные моменты функционирования эволюционной системы. Вначале эта система содержит небольшое число модулей, определяемых априорными знаниями специалистов в некоторой проблемно-предметной области, при этом некоторое множество правил может быть заранее встроено в систему. На основе процедуры самоорганизации и адаптации к внешней среде происходит постепенное накопление знаний системы подобно тому, как это делается в мозге человека, т.е. система становится все более интеллектуальной.

В ответ на входное воздействие активизируется определенный нейросетевой модуль на основе введенной меры сходства входного вектора с вектором весовых коэффициентов входных связей этого модуля. Если ни один из модулей не активизируется или эта активность находится ниже заданного порога, создается новый модуль (в простейшем случае нейронный узел или даже нейрон). Связи между модулями и нейронами определяются «возрастом» и средней активностью модулей на протяжении всего жизненного цикла существования эволюционной системы. Одни и те же нейроны могут быть объединены в классы, на основе их реакций на входные образы. Если, допустим, создается новый нейронный узел в ответ на некоторый входной вектор X , то этот узел на основе принципов самоорганизации начинает «сближаться» с нейронами, имеющими высокую активность на вектор X даже при недостаточном уровне сходства этого вектора с векторами входных связей нейронов. Таким образом, система эволюционирует во времени.

Реализация такого поведения системы осуществляется с использованием двух основных форм обучения: с учителем и без учителя (на основе самоорганизации). Основные требования, которые предъявляются к нейросетевым модулям в составе эволюционной системы, связаны с необходимостью обеспечения пластичности – способности воспринимать новые образы, и в то же время сохранению стабильности, гарантирующей не разрушение старых образов без потери ранее накопленной информации. К нейронным сетям, обладающим таким свойством, относятся сети адаптивного резонанса (Adaptive Resonance Theory - ART-сети). Для повышения эффективности функционирования нейросетевых модулей на основе обучения с учителем необходимо модифицировать существующие алгоритмы обучения с тем, чтобы адекватно реагировать на изменения во внешней среде. Такие алгоритмы реализованы в [Комарцова Л.Г., 2002].

Наиболее важные достоинства нейронных сетей с адаптивным резонансом (ART-сетей) сводятся к следующему:

- при подаче на вход НС некоторого образа принимается «пластичное» решение о появлении нового образа и «стабильное» решение о совпадении со старым образом; это позволяет решить проблему стабильности-пластичности, которую очень трудно реализовать в сетях перцептронного типа.

- НС типа ART (ART-1, ART-2, ART-3) обладают свойством самоорганизации, т.е. обучаются по алгоритму обучения без учителя, который основывается на использовании методов on-line кластеризации, квантизации и прототипного обучения [Metropolis N. and al.];

- в ART-сетях реализована возможность создания оптимальной топологии НС (с минимальным числом нейронов в выходном слое).

Однако эти достоинства влекут за собой необходимость управлять ростом сети, что в оригинальном алгоритме обучения ART осуществляется путем выбора пользователем желаемой величины параметра сходства $\rho \in [0,1]$. Чем выше значение этого параметра, тем большее число классов прототипов, и, соответственно, нейронов выходного слоя будет создано сетью. В предельном случае ($\rho=1$) происходит образование отдельного класса для каждого экземпляра обучающей выборки. Очевидно, что в этом случае существенно снижается способность нейронной сети к обобщению.

Чрезмерный рост числа распознаваемых классов приводит к неэффективному использованию компьютерных ресурсов и замедлению кластеризации. Слишком малое количество образованных классов прототипов, в свою очередь, может обусловить низкую точность распознавания.

Задача выбора числа необходимых классов, которые может распознать сеть, перекладывается на пользователя, который должен выбрать значение параметра ρ . Оптимальный выбор этого параметра трудоемок и требует проведения большого числа экспериментов. Поэтому первостепенное значение приобретает автоматизация процесса роста нейронной сети без задания пользователем конкретного значения параметра сходства. Один из подходов к решению этой проблемы для сети ART-1, работающей с двоичными входными образами и обучающейся без учителя, рассмотрен ниже.

3. Алгоритм обучения сети ART-1

Образы, соответствующие некоторому узлу j НС ART-1 (рис.1), представляются весовым вектором этого узла – T_j . Реакция узла j на особенности входного образа определяется другим весовым вектором V_j . Реализация ключевого элемента ART «стабильность-пластичность» осуществляется на основе использования параметра сходства ρ , который контролирует степень различия между новыми образами и запомненными старыми: $\rho \in [0,1]$.

Будем рассматривать алгоритм обучения ART-1 на основе on-line кластеризации с использованием

стратегии конкурентного обучения. On-line кластеризация – процесс группирования входных данных $X(t)$, $t=1, 2, \dots$, поступающих из непрерывного потока данных. При этом предполагается, что каждый входной вектор появляется однажды в системе и не может быть доступен снова.

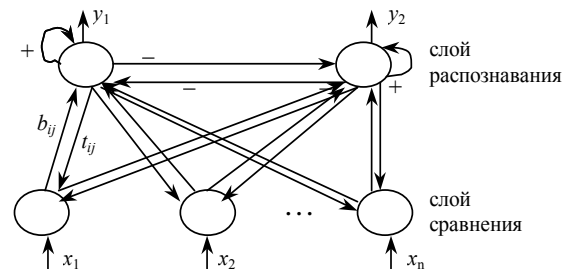


Рис. 1 Схема обучения сети ART-1

Многие реально существующие информационные системы оперируют такими данными, как: данные из сети Интернет, индекс акций (товаров), биологическая информация, информация о климате и т. д. Если данные являются динамичными (изменяемыми во времени), то это требует использования адекватных вычислительных моделей, которые должны приспосабливаться к ситуации. Поскольку для таких данных трудно определить их принадлежность классу, используются алгоритмы обучения без учителя. Таким образом, использование on-line кластеризации при обучении ART-1 позволяет использовать этот тип сети для построения интеллектуальных эволюционных систем, могущих работать в реальном масштабе времени, что создает для предпосылки для реализации ИЭКОС. Сеть ART является единственным типом сетей, в которой реализована возможность создания оптимальной топологии сети (минимальное число нейронов в выходном слое). Важное достоинство сетей этого типа проявляется в способности решить проблему стабильности-пластичности

В ART-1 (рис. 1) имеются следующие типы связей между нейронами: b_{ij} – от каждого i -го входного нейрона к каждому j -му выходному ($i=1, \dots, n$, $j=1, \dots, m$); t_{ij} – от каждого выходного нейрона к входным; все выходные нейроны имеют рефлексивную положительную связь и тормозящую связь с другими выходными нейронами (на рис. 1 в выходном слое для простоты изложения алгоритма обучения представлены только 2 нейрона).

Алгоритм обучения ART-1 состоит из двух главных фаз. Первая фаза связана с определением нейрона - победителя путем вычисления степени активности выходных нейронов сети в ответ на подачу некоторого входного вектора: нейрон - победитель имеет наибольшую степень активности. Во второй фазе вычисляется рассогласование между входным образом и прототипом, связанным с нейроном - победителем. Если это рассогласование ниже необходимого, этот прототип обновляется в соответствии с новым вектором, если превышает, то

процедура продолжается для других нейронов или создается новый нейрон.

Запишем алгоритм обучения ART-1.

Шаг 1. Инициализация весовых коэффициентов:

$$t_{ij} := 1; \quad b_{ij} := 1/(1+n) \quad \forall i=1,2,\dots,n; j=1,2,\dots,m$$

Шаг 2. Установка параметра сходства ρ из условия: $0 \leq \rho \leq 1$.

Шаг 3. WHILE (имеются входные вектора) DO

(a) подача на вход сети нового входного вектора:

$$\mathbf{X} = (x_1, x_2, \dots, x_n);$$

(b) вычисление значений выходов сети:

$$y_j = \sum_{i=1}^n b_{ij} x_i, \quad \forall j=1,2,\dots,m$$

(c) определение номера нейрона - победителя j^* с наибольшей величиной выхода;

(d) определение сходства входного вектора \mathbf{X} (образа) с прототипом \mathbf{T}_{j^*} , который соответствует нейрону- победителю j^* : IF (число единиц в двоичном представлении пересечения векторов \mathbf{X} и \mathbf{T}_{j^*} , деленное на число единиц в \mathbf{X} больше, чем ρ , THEN GOTO (f)

ELSE

(e) выход нейрона j^* сбрасывается, затем продолжается процедура поиска нового нейрона-победителя; GOTO (b);

(f) входной вектор \mathbf{X} ассоциируется с прототипом \mathbf{T}_{j^*} , поэтому производится изменение весовых коэффициентов прототипа:

$$t_{ij} := t_{ij} * x_{ij}^*, \quad \forall i=1,2,\dots,n;$$

(g) весовые коэффициенты b_{ij} также изменяются:

$$b_{ij} := b_{ij}^* + t_{ij}^* x_{ij} / (0.5 + \sum_{i=1}^n t_{ij}^* x_i).$$

Шаг 4. Конец.

В сети ART-1 число формируемых кластеров зависит от величины ρ . Чем выше значение ρ , тем большее число кластеров требуется для представления входных данных. Для автоматического формирования кластеров будем использовать возможности нечеткого управления.

4. Управление ростом сети ART-1 на основе методов нечеткой логики

В настоящее время известны два, наиболее часто применяемые на практике типа алгоритмов обучения нечеткой ART: 1) быстрого обучения и 2) использующие нормализацию входных векторов с помощью комплементарного (дополнительного) кодирования [Kussul E.M. and al.]. Нечеткость рассматриваемой модели ART-1 определяется видом используемого логического оператора AND (вместо оператора пересечения \cap в ART-1 оператора \wedge (min) в fuzzy ART-1). Оператор min сводится к оператору пересечения в случае двоичных аргументов.

В алгоритмах быстрого обучения в формуле обновления векторов:

$$\mathbf{T}_j(t+1) = \beta(\mathbf{X} \wedge \mathbf{T}_j(t) + (1-\beta)\mathbf{T}_j(t),$$

где β - коэффициент обучения, устанавливается $\beta=1$, посредством этого входной вектор \mathbf{X} быстро сходится к некоторому кластеру или создается новый нейрон, после чего β устанавливается <1 , когда определен кластер для входного вектора.

Для более сложного алгоритма в случае нормирования входных векторов в качестве меры нечеткости вводится функция выбора нейрона - победителя:

$$A_j(\mathbf{X}) = \frac{|\mathbf{X} \wedge \mathbf{T}_j|}{\varepsilon + |\mathbf{T}_j|},$$

где $\varepsilon > 0$, является константой, а нечеткий AND оператор \wedge определяется как оператор min, $|\bullet|$ - норма вектора. Функция выбора показывает степень, с которой весовой вектор \mathbf{T}_j является нечетким подмножеством входного вектора \mathbf{X} .

В предлагаемом алгоритме нечеткость вводится на уровне задания приращений параметра сходства ρ , в зависимости от ситуации, возникающей в ART-1 при кластеризации.

Общая схема подстройки различных параметров обучения нейронной сети (НС) с помощью нечеткой логики представлена на рис. 2. Нечеткая база правил, содержащая экспертные знания, используется для адаптивного изменения значений параметров обучения, в зависимости от состояния сети и реального значения параметров обучения.

Определим в качестве параметра управления величину ρ . Предположим, что априорно известно число кластеров N_a . Проведем первоначальную кластеризацию по приведенному выше алгоритму: пусть количество реальных кластеров равно N_r . Если $N_r < N_a$, то требуется подстройка параметра ρ с последующим новым предъявлением входных данных. Если $N_r > N_a$, ρ можно увеличивать, так как имеется число дополнительных кластеров $N_a - N_r$ (что приведет к более точной классификации); в противном случае, при $N_r < N_a$ - ρ должно уменьшаться, чтобы все входные векторы могли быть отнесены к одному из существующих кластеров.

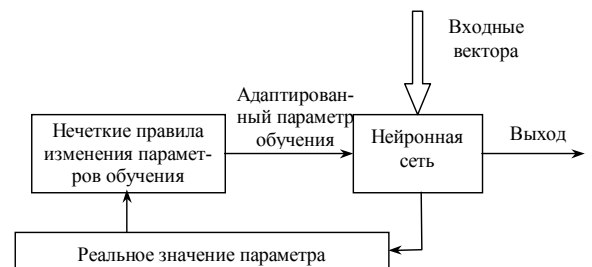


Рис. 2. Схема подстройки параметров обучения НС на основе методов нечеткой логики

Таким образом, в алгоритме обучения нечеткой ART-1 после проведения очередной кластеризации по алгоритму ART-1 необходимо определять

величину изменения ρ , т.е. $\Delta\rho$ которое позволит N_r приблизиться к N_a . Схема вычисления искомого значения приращения параметра обучения $\Delta\rho$ (рис. 3) соответствует базовой схеме нечеткого контроллера [Kussul E.M. and al.].

Нечеткие правила, построенные на основе проведенного эмпирического анализа результатов работы нечеткой ART, позволили определить процедуру изменения ρ в соответствии с табл.2. Здесь $E = N_a - N_r$ и IE - изменение E. Определены следующие нечеткие множества: NB – отрицательное большое; NS – отрицательное малое; ZE – нуль; PS – положительное малое; PB – положительное большое.

Из табл.1, например, можно извлечь следующее правило: IF E есть положительное малое (PS) AND IE есть положительное большое (PB), THEN $\Delta\rho$ есть положительное малое (PS).

Далее по правилам нечеткой логики, зная вид функции принадлежности нечеткой переменной $\Delta\rho$ и используя дефазификацию, извлекается числовое значение приращения.

Табл. 1 Фрагменты нечетких правил

$IE \backslash E$	NB	NS	ZE	PS	PB
NB	–	NS	ZE	PS	–
NS	NB	NS	ZE	PS	PB
ZE	NB	NS	ZE	PS	PB
PS	NB	NS	ZE	PS	PB
PB	–	NS	ZE	PS	–

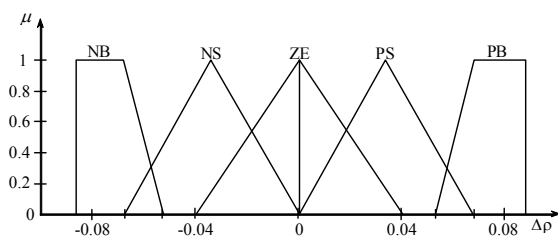


Рис. 3 Функции принадлежности для $\Delta\rho$

Определив значение $\Delta\rho$, сеть ART-1 выполняет повторную классификацию при новом значении $\Delta\rho$. Алгоритм заканчивает свою работу (устанавливает соответствующее значение ρ), когда N_r сравнивается с N_a .

Следует отметить, что на этапе распознавания в нечеткой ART-1 используется стандартная процедура поиска нейрона – победителя с $\rho = 0$, поскольку в общем случае образования новых классов не происходит.

Для оценки эффективности предложенного алгоритма были использованы стандартный тестовый набор по распознаванию ирисов и генератор тестов. Характеристики используемых в

экспериментах тестовых наборов представлены в табл. 2.

При проведении экспериментов параметр сходства ρ в одном случае был выбран равным нулю для обеспечения минимального размера сети, а в другом - равным $\rho = 0.9$ для достижения более высокой точности. В табл. 3 представлены результаты классификации на выбранных наборах данных. В качестве результатов использовались ошибки классификации (даваемые в процентах от общего числа имеющихся примеров) и число образованных в результате функционирования ART-1 кластеров.

Табл. 2. Характеристики тестовых наборов

Тесты	Кол-во классов	Размерность векторов	Кол-во векторов
Распознавание сортов ирисов	3	4	150 (75, 5)
Генератор тестов	5	4	100-500

Результаты классификации для генератора тестов усреднены по 10 независимым компьютерным экспериментам, для распознавания сортов ирисов - по 100 экспериментам и представлены в табл.3. (ошибки распознавания образов даются в % от общего числа предъявляемых образов).

Табл. 3. Результаты классификации

Тесты	ART-1 $\rho=0$		ART-1 $\rho=0.9$		ART-1 с неч. управлением ρ	
	Ошибки (%)	Кластер	Ошибки (%)	Кластер	Ошибки (%)	Кластер
Распознавание сортов ирисов	6,2	5	5,4	18	4,2	11
Генерат. тестов (200)	17,2	8	14,2	26	13,1	17
Генерат. тестов (500)	16,2	7	13,7	19	10,8	15

Анализ полученных результатов показывает, что ошибка распознавания при использовании сети ART-1 с нечетким управлением ρ значительно меньше по сравнению с другими алгоритмами.

ВЫВОДЫ

Проведенное экспериментальное исследование показало, что использование методов нечеткой логики для управления параметрами обучения нейронной сети адаптивного резонанса ART-1 позволяют автоматизировать процедуру настройки параметров сети в процессе ее

функционирования, что позволяет сделать вывод о целесообразности использования этого типа сетей для построения ИЭКОС.

Точность распознавания (табл.3.) рассмотренного алгоритма практически не уступает точности стандартного алгоритма ART –1 при больших значениях параметра сходства, а число образуемых кластеров оказывается меньше. Кроме того, на основе предложенного алгоритма по желанию пользователя можно регулировать соотношение точности – размер нейронной сети.

Рассмотренные гибридные нечеткие нейросетевые модели на основе ART-1, в отличие от других подобных моделей, позволяют повысить качество работы динамических интеллектуальных систем за счет использования лучших свойств каждой из объединяемых технологий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Ларичев О.И. и др.], Ларичев О.И., Финн В.К. Теория и методы создания интеллектуальных компьютерных систем // Информационные технологии и вычислительные системы. - 1998.-№1.
- [Еремеев А.П.] Еремеев А.П. Концепции времени и их применение в интеллектуальных системах // Сб. научн. тр. «Интеллектуальные системы и технологии». –М.:МИФИ.-2004.
- [Grossberg S.] Grossberg S. Stadies of Mind and Brain. –Reidel. – Boston.-1982.
- [Люгер Дж.] ЛюгерДж. Искусственный интеллект. Стратегии и методы решения сложных проблем.- Пер. с англ. –М.: Изд дом «Вильямс».-2003.
- [Комарцова Л.Г.] Комарцова Л.Г. Исследование алгоритмов обучения многослойного персептрона //Нейрокомпьютеры. Разработка и применение. – М.:Радиотехника. 2002.-№12.
- [Kirkpatrick S. and al.] Kirkpatrick S, Gellat C.D. Vecchi M. Optimization by Simulated Annealing. Sciece. -1983.-vol. 220.
- [Metropolis N. and al.] Equation of calculation by fast computing machines. -J. of Chem. Phys. -21(6). – 1998.
- [Комарцова Л.Г. и др.] Комарцова Л.Г. Максимов А.В. Нейрокомпьютеры.-М.: МГТУ им. Н.Э. Баумана.-2004.
- [Kussul E.M. and al.], Kussul E.M., BaidykT.N., LukowichV.V., Rachkovskij D.A. Adapt. NN-classifier with multfloat input coding //Proc.of 6-th Int. Conf. "NeuroNimes-2008". -Nimes, France. -2008.
- [Chin-Teng Lin] Chin-Teng Lin. Neural Fuzzy Systems to Intelligent Systems//A.Simon and Schuster Company. Upper Saddle River, NJ 07458.-1996 by Prentice Hall.

DYNAMIC MODELS OF INTELLIGENT SYSTEMS ON THE BASE OF ADAPTIVE RESONANCE THEORY NEURAL NETWORK

Komartsova L.G., Lavrenkov Ju.N.,
Antipova O.V., Kadnikov D.S.

*Moscow Bauman State Technical University,
Kaluga filial, Russia*

ikomartsova@yandex.ru, polikarp1@rambler.ru

The article is devoted to the problems for solving difficult problems, such as prediction, planning, pattern recognition and knowledge discovery in a number of application areas: bioinformatics, speech and language, image and video analysis, other engineering disciplines.

Most of these publications deal with static process, assuming that the process is represented adequately by the data available at present and that it does not change over time.

When the process is evolving, the modelling system needs to be trace the dynamics of the process and to be adapt to changes in the process. Many real-world problems from engineering, economics, social sciences require continuously adapting models. The modelling system needs to be evolving.

INTRODUCTION

For the evolutionary process is necessary to monitor the dynamics of their changes over time (ie, the operation mode of on-line) and adaptation to the environment (for example, the robot in an unfamiliar environment). In this paper we investigate some issues that need to be addressed for the creation of evolutionary systems that simulate human decision-making process in the complex is difficult to formalize problems in real time.

MAIN PART

Based on the analysis of the features that should have an evolutionary system architecture proposed by intellectual evolving system, which is a multi-module and multi-tier open architecture that can adapt to a particular domain based on the additional training of neural network modules in the operation of this system.

Solved the problem of automating the process of growth of the neural network ART-1, without specifying the user-specific values for similarity based on the use of fuzzy logic methods, which significantly reduce training time.

The fuzzy rule base that contains expertise has been used for adaptive changes in the values of learning parameters, depending on network conditions and real values of the parameters of learning.

At the stage of recognition in fuzzy ART-1 uses a standard search procedure of the neuron - the winner with $\rho = 0$, since in general the formation of new classes is not happening. To evaluate the effectiveness of the proposed algorithm was used a standard test set for recognizing irises and test generator.

CONCLUSION

Experimental tests showed that the use of fuzzy logic methods to control the parameters of neural network learning adaptive resonance ART-1 allow us to automate network configuration during its operation, which suggests the usefulness of this type of networks for dynamic intelligent systems.

The above hybrid fuzzy neural network model based on the ART-1, in contrast to other similar models can improve the dynamic performance of intelligent systems by using the best features of each of the merged technologies.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.942

ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЙ ПРИ ПОСТРОЕНИИ СИСТЕМ РАСПОЗНАВАНИЯ ОБРАЗОВ

Жукевич А.И., Олизарович Е.В., Родченко В.Г.

*Гродненский государственный университет имени Янки Купалы,
г. Гродно, Республика Беларусь*

san@grsu.by

e.olizarovich@grsu.by

rovar@mail.ru

При построении систем распознавания образов предусматривается возможность выполнения процедуры обучения в автоматическом режиме на основе анализа исходных данных, предварительно формируемых в виде классифицированной обучающей выборки (КОВ). Процесс формирования КОВ начинается с определения алфавита классов и априорного словаря признаков. Создание необходимых онтологий позволит автоматизировать как процесс построения алфавита классов и априорного словаря признаков, так и представление результатов распознавания.

Ключевые слова: алфавит классов, онтология, система распознавание образов, словарь признаков.

ВВЕДЕНИЕ

При решении целого ряда фундаментальных и прикладных задач в области естественнонаучных, социально-экономических и гуманитарных дисциплин приходится работать с многомерными объектами, которые обычно характеризуются большим числом признаков. В этом случае применение традиционного математического аппарата оказывается весьма затруднительным, и в качестве альтернативы более эффективным оказывается использование подходов и методов теории распознавания образов [Васильев, 1989].

Построение системы распознавания образов на основе наблюдаемых данных предполагает выполнение двух основных этапов, связанных с реализацией процедуры обучения и процедуры принятия решения. Процесс начинается с определения исходного алфавита класса и априорного словаря признаков.

Соответствующий алфавит классов формируется на основе предварительного анализа требований, предъявляемых к системе распознавания. В априорный словарь должны включаться такие признаки, которые отражают наиболее характерные особенности распознаваемых системой классов. Очевидно, что задача выбора наиболее четко разделяющих классы признаков, является нетривиальной, поскольку, с одной стороны, состояния сложной системы обычно

характеризуются большим числом разнообразных по своей природе признаков, а, с другой стороны, часто оказывается невозможным детерминировать закономерности функционирования сложных систем.

При реализации систем распознавания важно ориентироваться на использование комплексного подхода, который предусматривает учет и анализ признаков, характеризующих разнообразные аспекты исследуемых объектов.

Для построения систем распознавания разработан ряд алгоритмов, которые позволяют в автоматическом режиме выполнить как процедуру обучения, так и процедуру принятия решения. Наиболее “узким” местом, с точки зрения автоматизации, остаются шаги алгоритма, связанные с формированием алфавита классов, априорного словаря признаков и представлением результатов распознавания [Родченко, 2008].

Использование онтологий является одним из перспективных направлений автоматизации как выполнения подготовительной работы по формированию алфавита классов и априорного словаря признаков, так и реализации заключительного этапа, связанного с представлением результатов. Таким образом, при построении систем распознавания образов могут быть использованы START-онтологии (S-онтологии) и RESULT-онтологии (R-онтологии).

Описание алгоритма построения системы распознавания

Построение системы распознавания можно осуществить на основе использования универсального алгоритма, предусматривающего выполнение семи шагов.

Первый шаг построения системы распознавания связан с формированием алфавита классов $A = \{A_1, A_2, \dots, A_k\}$ и априорного словаря признаков $P = \{P_1, P_2, \dots, P_n\}$. На основе тесного взаимодействия специалистов в соответствующей прикладной области и в области компьютерного анализа данных, путем проведения экспертных оценок и заключений формируются, во-первых, перечень соответствующих классов, который в данном случае будет представлять собой алфавит классов, и, во-вторых, априорный словарь признаков.

На втором шаге построения системы распознавания формируется классифицированная обучающая выборка. Результаты измерений значений всех признаков из априорного словаря для каждого экземпляра класса представляют собой вектор-столбец $x^T = (x_1, x_2, \dots, x_n)$. Если для каждого j -го класса состояний m_j значений всех соответствующих векторов-столбцов записать в виде таблицы, то результат описания объектов этого класса будет представлять прямоугольную матрицу, состоящую из n строк и m_j столбцов. В результате для каждого класса $A_j \subset A$ формируется соответствующая матрица X_j размерности $n \times m_j$, где m_j – число измерений экземпляра j -го класса. Матрица X_j будет представлять собой формальное описание j -го класса в многомерном априорном признаковом пространстве. Объединенная прямоугольная матрица, построенная на множестве $X = \{X_1, X_2, \dots, X_k\}$ будет представлять собой классифицированную обучающую выборку.

Третий шаг предназначен для сепарирования признаков из априорного словаря с целью исключения из дальнейших исследований малоинформативных признаков. Природа этих признаков такова, что они “размывают” образы эталонов классов и в итоге создают помехи при выполнении заключительной процедуры принятия решения. В результате выполнения формируется уточненный рабочий словарь, содержащий только признаки, которые наиболее четко отражают особенности каждого класса из априорного словаря.

Отметим, что задача формирования наилучшей системы признаков относится к разряду наиболее сложных с технической и методологической точек зрения. В реальных системах далеко не все признаки, которые первоначально включаются в априорный словарь, пригодны для выполнения непосредственно процедуры принятия решений [Вакульчик и др., 2005]. Ошибка в выборе признаков может приводить к содержательно ложной классификации, даже если при этом она будет формально обоснованной.

Для решения задачи построения наилучшей

системы признаков традиционно предлагается воспользоваться эвристическими алгоритмами, которые базируются:

- на полном переборе вариантов и максимизации некоторого критерия информативности признака или подсистемы признаков [Барабаш, 1983];
- на основе случайного поиска с адаптацией, когда наиболее информативная подсистема признаков обнаруживается с помощью случайного перебора подсистем с “поощрением” и “наказанием” отдельных признаков [Загоруйко и др., 1985];
- на применении метода экстремальной группировки признаков или метода корреляционных плеяд [Айвазян и др., 1989].

Для сепарирования признаков по уровню информативности, предлагается воспользоваться алгоритмом, который основывается не на полном или частичном переборе подсистем признаков, а на анализе и учете статистических характеристик выборок значений признаков. Такой подход позволяет на основе априорного словаря автоматически сформировать уточненный рабочий словарь.

Признаки из исходного априорного словаря сепарируются на три вида. К первому виду будут относиться такие признаки, значения которых фактически подчиняются одному и тому же закону распределения во всех классах $A = \{A_1, A_2, \dots, A_k\}$. Эти признаки не несут разделяющей разницы между классами, а потому будут “размывать” образы классов, как на этапе обучения системы, так и при выполнении процедуры принятия решения.

Ко второму виду будут относиться те признаки, для которых в результате сопоставления всех пар выборок значений этого признака из разных классов оказалось, что ни разу не выполнен соответствующий критерий однородности. Признаки такого вида будут обеспечивать разделение формальных образов классов в многомерном пространстве решений. Именно они и включаются в рабочий словарь информативных признаков, на основе которого в дальнейшем строятся компактные и разделенные в многомерном пространстве решений эталоны классов.

Признаками же третьего вида являются те, которые в процессе выполнения процедуры сепарирования не были отнесены ни к первому, ни ко второму виду. Природа этих признаков такова, что они не отражают какие-либо четко выраженные межклассовые различия.

На основании вышеизложенного, сепарирование признаков из априорного словаря выполняется по следующему сценарию: анализируется содержимое матрицы X , которая получается путем объединения матриц X_1, X_2, \dots, X_k . Матрица X будет содержать n строк и m столбцов, причем значение m будет представлять собой сумму количества столбцов во всех матрицах соответственно X_1, X_2, \dots, X_k , т.е. $m = m_1 + m_2 + \dots + m_k$. На основе соответствующего исследуемому признаку P_i статистического

критерия однородности последовательно исследуются все признаки из априорного словаря $P = \{P_1, P_2, \dots, P_n\}$. В результате они разбиваются на три вида $P^{(1)} = \{P_1^{(1)}, P_2^{(1)}, \dots, P_{n_1}^{(1)}\}$, $P^{(2)} = \{P_1^{(2)}, P_2^{(2)}, \dots, P_{n_2}^{(2)}\}$, $P^{(3)} = \{P_1^{(3)}, P_2^{(3)}, \dots, P_{n_3}^{(3)}\}$, где $P = P^{(1)} \cup P^{(2)} \cup P^{(3)}$ и $n_1 + n_2 + n_3 = n$.

Отнесение очередного признака P_i к одному из трех видов производится по следующему правилу:

- если для всех возможных пар классов подтвердились гипотезы о статистической однородности выборок значений этого признака для двух сравниваемых классов, то признак P_i относится к первому виду;
- если для всех возможных пар классов оказалось, что выборки значений признака P_i для двух сравниваемых классов подтвердили гипотезу об их неоднородности, то этот признак P_i относится ко второму виду;
- если для признака P_i не выполнилось ни одно из двух предыдущих условий, то он относится к признакам третьего вида.

В рабочий словарь включаются только признаки второго вида $P^{(2)} = \{P_1^{(2)}, P_2^{(2)}, \dots, P_{n_2}^{(2)}\}$. Отметим, переход к следующему шагу алгоритма происходит только в случае непустого рабочего словаря (т.е. когда $n_2 \neq 0$), а иначе необходимо возвращаться к началу и формировать новый вариант априорного словаря.

Четвертый шаг алгоритма предусматривает проведение аттестации признаков из словаря $P^{(2)}$ и проверки достоверности распознавания на основе использования этих признаков. Из матриц X_1, X_2, \dots, X_k исключаются строки, содержащие значения признаков первого $P^{(1)}$ и третьего $P^{(3)}$ видов, а все значения признаков второго вида нормируются к единичному интервалу по формуле $y_i = (x_i - x_{\min}) / (x_{\max} - x_{\min})$. В итоге получаются матрицы Y_1, Y_2, \dots, Y_k размерности $n_2 \times m_i$. Для аттестации признаков матрицы, множество $Y = \{Y_1, Y_2, \dots, Y_k\}$ распределяются на два подмножества таким образом, что $Y_i = Y_i^{(1)} + Y_i^{(2)}$. Матрица $Y_i^{(1)}$ имеет размерность $n_2 \times m_i^{(1)}$ (где $m_i^{(1)} = \lfloor m_i / 2 \rfloor$ - количество объектов, включенных в матрицу $Y_i^{(1)}$), а матрица $Y_i^{(2)}$ будет размерности $n_2 \times m_i^{(2)}$ (где $m_i^{(2)} = m_i - m_i^{(1)}$ - количество объектов, включенных в матрицу $Y_i^{(2)}$). На основе столбцов матрицы $Y^{(1)} = \{Y_1^{(1)}, Y_2^{(1)}, \dots, Y_k^{(1)}\}$ строятся эталоны $E^{(1)} = \{E_1^{(1)}, E_2^{(1)}, \dots, E_k^{(1)}\}$ для каждого класса и задается пороговое значение допустимости ошибочных классификаций Q .

Далее проводится классификация объектов из множества $Y^{(2)} = \{Y_1^{(2)}, Y_2^{(2)}, \dots, Y_k^{(2)}\}$ и вычисляется число ошибочных классификаций G . Затем подмножества $Y^{(1)} = \{Y_1^{(1)}, Y_2^{(1)}, \dots, Y_k^{(1)}\}$ и $Y^{(2)} = \{Y_1^{(2)}, Y_2^{(2)}, \dots, Y_k^{(2)}\}$ меняются ролями, и на основе $Y^{(2)}$ строятся эталоны классов, а объекты из $Y^{(1)}$ подвергаются процедуре контрольного распознавания. Если значение G не превышает пороговое значение Q , то контрольная аттестация прошла успешно, а иначе необходимо вернуться к

началу алгоритма и сформировать новый вариант априорного словаря.

На пятом шаге алгоритма выполняется построение эталонов классов. Один из вариантов реализации этого шага предусматривает, что на основе матриц $Y = \{Y_1, Y_2, \dots, Y_k\}$ формируются эталоны классов $E = \{e_1, e_2, \dots, e_k\}$ для “компактных” образов, где эталон i -го класса вычисляется как “центр тяжести” для Y_i , или в более общем виде $E = \{E_1, E_2, \dots, E_k\}$ для эталонов произвольной сложности.

Шестой шаг связан с выполнением процедуры принятия решения. Распознаваемый объект на основе использования признаков из рабочего словаря формально представляется в виде матрицы Y_{k+1} . Измеряются характеристики взаимного размещения образов эталонов $E = \{E_1, E_2, \dots, E_k\}$, и образа распознаваемого объекта.

Завершающий седьмой шаг алгоритма связан с окончательной интерпретацией полученных результатов и выработкой заключения. Если в полученном наборе доминируют индексы одного и того же j -го класса, и к Y_{k+1} ближе всего расположен эталон E_j , то это говорит о том, что исследуемый объект относится к j -ому классу.

О применении онтологий

Онтология позволяет определить единый словарь для специалистов, которые совместно используют информацию в соответствующей предметной области. В нашем случае S-онтология должна включать машинно-интерпретируемые формулировки необходимых базовых понятий предметной области и отношения между ними, тогда как R-онтология ориентирована на машинно-интерпретируемое представление результатов выполнения процедуры распознавания. Онтологии могут быть представлены в виде иерархии и в свою очередь состоят из взаимозависимых онтологий, которые могут быть декомпозированы на составляющие.

Опыт реализации систем распознавания подсказывает, что разработка онтологий потребует затрат разнообразных ресурсов, а потому существует вопрос о реальной потребности в построении онтологий [Олизарович и др., 2010].

Наличие S-онтологии при построении системы распознавания образов предоставляет возможность автоматизировать процедуру формирования алфавита классов, содержащих определение диагностируемых состояний, и процедуру определения априорного словаря признаков, который в общем случае представляют собой выборку из генерального словаря. Построение S-онтологии осуществляется на основе использования семантических сетей.

По результатам проведенного анализа содержимого классифицированной обучающей выборки может быть построена R-онтология,

которая будет использоваться для оперативного выполнения процедуры принятия решений. Специфика R-онтологии такова, что она базируется на использовании матричного формата представления образов типа “объект-свойство” и на формализованном представлении знаний в виде кластерных структур.

Опыт построения онтологии для практического их использования свидетельствуют о том, что любая онтология представляет собой сложную систему, которая в полной мере отвечает пяти признакам, сформулированным Г.Бучем [Буч, 2000].

Выбор того, какие компоненты в онтологии считаются элементарными, относительно произволен и в большей степени остается на усмотрение специалистов. Внутриконтентная связь сильнее, чем между компонентами, а сами онтологии состоят из немногих типов структурных компонентов.

Любая работающая онтология является развитием более простой, а разработка онтологии представляет собой итеративный процесс, который обычно продолжается в течение всего жизненного цикла онтологии.

ЗАКЛЮЧЕНИЕ

Анализ существующих подходов к построению онтологий свидетельствует об отсутствии некой универсальной методологии их разработки. Таким образом, на сегодняшний день не существует некоторого единственно правильного способа моделирования предметной области. Обеспечение доступа всех заинтересованных специалистов к разработке онтологии потенциально должно обеспечивать формирование наиболее качественного варианта и здесь в полной мере и проявляются преимущества проекта OSTIS.

При использовании онтологий в процессе построения аналитических систем предоставляется возможность отделения системных знаний предметной области от оперативных, случайных данных. Таким образом, в процессе построения систем распознавания можно независимо разработать онтологию обобщенного алфавита классов и онтологию обобщенного словаря признаков, а затем на их основе реализовать их оперативное наполнение с целью формирования классифицированной обучающей выборки.

Построение специализированных S-онтологий, в значительной мере ориентировано на использование результатов развития смежных онтологий. Если одна группа специалистов улучшает смежную онтологию, то и за счет этого может происходить развитие S-онтологии и соответственно повышается качество и универсальность соответствующей системы распознавания.

Построение и расширение R-онтологий способствует получению новых знаний об исследуемых сложных системах, поскольку

процедура обучения направлена на выявление новых признаков классификации.

Разработка онтологий для анализа систем на основе методов распознавания образов, связана с реализацией как совместного использования специалистами, или программными агентами, так и совместной разработки, что в конечном итоге обеспечивает гораздо более качественное понимание структуры информации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Айвазян и др., 1989] Айвазян, С.А. Прикладная статистика. Классификация и снижение размерности: Справ. изд. / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин. – М.: Финансы и статистика, 1989.
- [Барабаш, 1983] Барабаш, Ю.Л. Коллективные статистические решения при распознавании / Ю.Л.Барабаш – М.: Радио и связь, 1983.
- [Буч, 2000] Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++ / Г. Буч – М.: СПб.: БИНОМ – Невский диалект, 2000.
- [Вакульчик и др., 2005] Вакульчик В.Г. Об одном методе построения математической модели исследования патологических процессов: диагностика острого аппендицита у детей / В.Г. Вакульчик, Ю.В. Макаревич, В.Г. Родченко // Известия Гомельского государственного университета имени Франциска Скорины. – 2005. –№ 5(35). – С.16-19.
- [Васильев, 1989] Васильев, В.И. Проблема обучения распознаванию образов / В.И. Васильев – К: Выща шк. Головное изд-во, 1989.
- [Загоруйко и др., 1985] Загоруйко, Н.Г. Алгоритмы обнаружения эмпирических закономерностей /Н.Г. Загоруйко, В.Н. Елкина, Г.С. Лбов – Новосибирск: Наука. – 1985.
- [Олизарович и др., 2010] Олизарович, Е.В. Метод построения систем диагностики компьютерных сетей на основе применения аппарата прикладной статистики / Е.В. Олизарович, В.Г. Родченко // Известия Гомельского государственного университета имени Франциска Скорины -2010.-№5(62).-С.84-88.
- [Родченко, 2008] Родченко, В.Г. Об одном методе формирования пространства решений при построении систем распознавания образов / В.Г. Родченко // Известия Гомельского государственного университета имени Франциска Скорины - 2008. - № 5(50). - С.95-99.

USE OF ONTOLOGY FOR THE CONSTRUCTION OF PATTERN RECOGNITION SYSTEMS

Zhukevich A.I., Olizarovich E.V., Rodchenko V.G.

*Yanka Kupala State University of Grodno,
Grodno, Republic of Belarus*

san@grsu.by

e.olizarovich@grsu.by

rovar@mail.ru

Construction of systems for pattern recognition provides for the possibility of teaching procedures implementation in automatic mode based on the analysis of initial data pre-generated in the form of classified training set (CTS). The process of CTS forming begins with identifying of the alphabet of classes and a priori features dictionary. Formation of the necessary ontology will allow to automate the process of building of the alphabet of classes and a priori features dictionary, as well as presentation of recognition results.

СЕКЦИЯ № 4

СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ МУЛЬТИМОДАЛЬНЫХ И ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ИНТЕРФЕЙСОВ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

SECTION № 4

SEMANTIC TECHNOLOGIES OF DESIGNING MULTIMODAL AND NATURAL LANGUAGE INTERFACES OF INTELLECTUAL SYSTEMS

Данная секция посвящена семантическим моделям пользовательских интерфейсов, поддерживающих различные формы взаимодействия пользователей с интеллектуальной системой.

Основное внимание уделено графическим и естественно-языковым интерфейсам. Кроме того рассмотрены различные методы, обеспечивающие в ходе диалога с пользователем приобретение различной информации о самом пользователе (о его особенностях и текущем состоянии). В частности рассмотрены методы определения эмоционального состояния пользователя.

Стремительный прогресс информационных технологий ставит новые задачи и требования к разработке пользовательских интерфейсов. Существует широкий круг задач, для которых требуется разработка альтернативных или так называемых мультимодальных пользовательских интерфейсов, т.е. интерфейсов, в которых используется звуковой, визуальный, тактильный способ передачи информации. Проектирование и реализация таких интерфейсов требует принципиально новых моделей и методов их создания. Представленные в рамках данной секции доклады описывают модели и методы, необходимые для создания интерфейсов новых типов.

Статьи коллектива авторов из Волгоградского государственного технического университета посвящены актуальному направлению, связанному с определением эмоциональных реакций человека по движениям и позам. Ими дана классификация характерных поз и их соответствие эмоциональным реакциям, включая интерпретацию, вербальное описание и изображение, которые описывают поведение человека во время его недовольства или нетерпения. Построены модели идентификации позы человека и эмоциональной реакции, характерных жестов и телодвижений в виде нечеткого последовательного темпорального высказывания. Построены правила соответствия полученных описаний интерпретациям психологов.

В работе коллектива авторов из Научно-Исследовательского Центра Электронной Вычислительной Техники (г. Москва) рассмотрена задача анализа изображений. Ими проанализированы существующие подходы к анализу изображений и предложен новый гибридный метод анализа, основанный на интеграции числового и символьного методов распознавания. В качестве языка представления знаний для символьного метода используется семантический гиперграф. Результатом анализа является описание ситуации на изображении, которое содержит не только экземпляры классов распознанных классов объектов, но также их характеристики и связи между ними.

Авторы Национального технического университета Украины «Киевский политехнический институт» рассмотрели вопрос обработки зашумленных физических сигналов, поступающих от различных датчиков. Авторами предложены методы фильтрации и сглаживания входных сигналов, реализован набор программных средств для обеспечения достоверности поступающих исходных данных. Приведенные результаты тестирования алгоритмов показали эффективность применения предложенных методов.

Большое число докладов данной секции посвящено семантическим моделям естественно-языковых интерфейсов интеллектуальных систем, актуальность разработки которых в последнее время неуклонно возрастает.

ВОПРОСЫ, ВЫНОСИМЫЕ НА ОБСУЖДЕНИЕ КРУГЛОГО СТОЛА СЕКЦИИ № 4 :

- *Что дает построение пользовательского интерфейса как интеллектуальной системы, семантически совместимой и интегрируемой с предметной интеллектуальной системой.*
- *Каковы направления интеллектуализации пользовательского интерфейса.*
- *Каковы направления и перспективы визуализации различных знаний.*
- *В чем суть когнитивной визуализации знаний.*
- *Нужно ли для диалога с пользователем проводить полный синтаксический семантический анализ ЕЯ-текстов.*
- *Какие перспективы для понимания ЕЯ-текстов имеет трактовка ЕЯ-интерфейса как интеллектуальной системы, семантически совместимой и интегрированной с предметной интеллектуальной системой.*
- *Как в процессе понимания ЕЯ-текстов могут взаимодействовать ЕЯ-интерфейс и предметная интеллектуальная система.*



УДК 004.822:514

ГИБРИДНАЯ СХЕМА АНАЛИЗА ИЗОБРАЖЕНИЙ

С.С. Курбатов *, А.П. Лобзин *, К.А. Найденова **, Г.К. Хахалин *

** Научно Исследовательский Центр Электронной Вычислительной Техники, г. Москва, Россия*

curbatow.serg@yandex.ru

lobzin@rambler.ru

gkhakhalin@yandex.ru

*** Военно-медицинская академия, г. Санкт-Петербург, Россия*

naidenovakseniya@mail.ru

В работе рассматривается схема анализа изображений, соединяющая преимущества числового и символического методов распознавания. Гибридизация двух подходов при использовании прикладной онтологии и при наличии в интегральной системе лингвистического транслятора и синтезатора изображений позволяет выйти на качественно более высокий уровень понимания изображений. В качестве языка представления знаний для символического метода используется семантический гиперграф.

Ключевые слова: гибридная система, прикладная онтология, семантический гиперграф, числовые и символические методы распознавания.

свойств подобной системы.

ВВЕДЕНИЕ

Разработчики систем распознавания вообще и, в частности, систем распознавания изображений чаще всего придерживаются тенденции разрабатывать системы, основанные на каком то одном из подходов, и пытаются доказать преимущества выбранного подхода над другими. Такой взгляд не всегда приводит к успеху, поскольку для решения проблемы анализа изображений часто требуются различные виды представления и обработки данных. В этой ситуации наиболее адекватным представляется попытка для решения задач восприятия и анализа изображений интегрировать разные подходы, разработанные в этой области, в «гибридную систему», чтобы полностью использовать преимущества каждой составляющей, а не пытаться решить их с помощью единственного подхода.

В данной работе кратко анализируется спектр существующих методов распознавания, обосновывается выбор составляющих гибридной системы и рассматриваются функции системы анализа изображений.

Изложение иллюстрируется примерами из проблемной области «Планиметрия», включающей как стандартные плоские фигуры, так и «детские» рисунки, составленные из этих фигур. В заключении дается характеристика синергетических

1. Типология методов распознавания образов и анализа изображений

Наши представления о компьютере как о «числовой мельнице» породили классические числовые методы распознавания (первое направление в этой области), основанные на цифровом представлении изображений. Успехи в исследованиях по искусственному интеллекту рассматривают компьютер как «интеллектуальный символический процессор», способный запоминать и обрабатывать информацию, не выраженную числами. Такой взгляд породил символический подход (второе направление) к обработке изображений, который позволяет решать не только задачи распознавания, но и задачи понимания изображений, включающие вывод по символической информации. Третьим направлением в обработке изображений является нейросетевой подход, который, как полагают, подражает человеческой системе восприятия и представляется более эффективным, чем современные компьютеризированные системы.

Коротко опишем «наполнение» каждого из этих направлений, отмечая преимущества и недостатки. Распределение методов распознавания по направлениям представлено на рисунке 1.

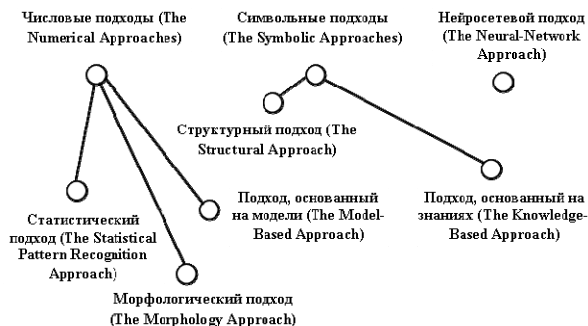


Рисунок 1 – Типология методов распознавания

Статистическое распознавание образов (The Statistical Pattern Recognition) определяет проблему распознавания изображений как задачу классификации на признаках, выделенных на изображении. Объекты представляют собой n -мерные векторы в пространстве признаков, а процесс принятия решения основывается на вычислении меры близости наблюдаемого вектора к области определения искомого объекта. В качестве возможных мер близости используются: расстояние, дискриминантная функция, вероятностные функции и т.п. Классификация проводится в соответствии со статистическими методами по многомерным функциям плотности условной вероятности [Горелик и др., 2004].

Фактически при таком подходе не требуется каких-либо конкретных знаний о прикладной области. Эти преимущества оплачиваются типичной трудностью в использовании априорных знаний о прикладной области, включая контекстуальную и структурную информацию.

Преимуществами подобных систем распознавания являются относительная эффективность, «прозрачная» постановка задачи, набитый потенциал и имеющие признание методы обучения.

Подход, основанный на эталонах (моделях) (The Model-Based Approach). Обычно цели этого подхода – распознать объекты и оценить их положение в зашумленных средах (например, содержащих частично перекрывающиеся объекты при анализе сцен). Распознавание выполняется сопоставлением данных изображения с моделями (эталонами) объектов. Модели описывают формы объектов, которые обычно устойчивы. Эталоны чаще всего основываются на пространственно локализованных признаках, например, ограничивающие грани, типы углов и т.п. Процесс поиска выполняется в пространстве всех возможных соответствий между данными и моделями. Процесс принятия решения представляет собой сравнение со всеми эталонами и вычисление критерия согласованности (например, корреляционной функции). Этот подход хотя и прост по логике (в основном над исходными изображениями производится лишь предварительная обработка с целью очистки изображения от шумов), но требует больших

вычислительных затрат, поскольку сравнение приходится проводить для всех возможных положений объекта в поле изображения [Катыс, 1990].

Морфологический подход (The Morphology-Based Approach). Под морфологическим анализом понимаются методы решения задач узнавания, классификации объектов, выделения отличий в сценах по их изображениям, оценивания параметров объекта по его изображению, основанные на исследовании математических моделей изображений [Животников и др., 2005]. Так как на практике некоторые детали модели, связывающей объект с его изображением, как правило, неизвестны, (например, условия освещения, оптические свойства поверхностей объекта, характеристики системы формирования изображения), то можно указать лишь множество изображений, порожденное данной сценой при всех возможных параметрах модели формирования изображений. Это множество изображений и называется его формой. Если это множество выпукло и замкнуто в нормированном пространстве всех изображений, а расстояние между изображениями – сильно выпуклая функция, то с формой изображения сцены можно однозначно связать проектор на это множество. В терминах этих проекторов и решаются названные задачи анализа изображений. Подобные методы достаточно ресурсоемки, но их развитие, основанное на нелинейных проекторах и теоретико-возможностных моделях, оказываются достаточно эффективны при неформализованном характере данных и ограниченном размере обучающей выборки [Пытьев, 2007].

Структурное распознавание образов (The Structural Pattern Recognition) имеет целью распознавать объекты на изображении на основе синтаксического и контекстуального описания [Горелик и др., 2004]. Эти методы предусматривают включение в процесс распознавания информации о структуре объекта и требуют установить, какие элементы-подобразы можно выделять в объекте, насколько они могут или должны быть элементарны и в каких отношениях находятся. Идея структурных методов состоит в изучении и использовании структур взаимоотношений характерных элементов объекта; она реализуется построением описания сложного объекта в виде иерархической структуры более простых подобразов.

При реализации структурного метода используется аналогия между построением объекта с помощью соединения различными способами подобразов и построением фраз и предложений языка с помощью соединения слов, состоящих, в свою очередь, из букв.

В рамках этого подхода для решения задач распознавания используются различные лингвистические конструкции, образованные словарем признаков и грамматикой — правилами конструирования фраз, в совокупности

описывающих классы и подлежащий распознаванию объект. В результате каждый объект получает свое представление с помощью некоторого набора непроемких элементов и ряда фиксированных синтаксических операций.

Система должна обладать способностью обнаруживать синтаксические связи, существующие в объекте. Решение о синтаксической правильности представления объекта, т. е. о принадлежности его к определенному классу, задаваемому определенной синтаксической системой или грамматикой, вырабатывается синтаксическим анализатором (блоком грамматического разбора). При выполнении грамматического разбора анализатор обычно воспроизводит полное синтаксическое описание объекта в виде дерева грамматического разбора, если соответствующий объект является синтаксически правильным. Структурные методы распознавания базируются на порождающей грамматике [Новикова, 2008].

Подход, основанный на знаниях (The Knowledge-Based Approach), интенсифицирует использование различных видов явных знаний, выраженных в символьной форме, и управление знаниями с помощью методов символьной обработки [Rohlfetal., 1996]. Такие знания касаются описаний объектов и их возможных конфигураций, стратегий распознавания изображений («управляющие знания») и т.д. В области искусственного интеллекта существуют методы для эффективного представления вышеперечисленных знаний (семантические сети, фреймы, продукционные системы и онтологии).

Модели и данные об изображениях обычно представляются на нескольких иерархических уровнях абстракции. Простыми примерами уровней являются уровень частей объекта и уровень целостных объектов. В этом случае модели частей объекта сопоставляются с признаками, тогда как группы смежных признаков («сложные признаки») сопоставляются с моделями целостного объекта. Обычно применяется парадигма «гипотеза-проверка»: гипотезы распознавания генерируются на основе доступных знаний и проверяются с помощью процесса сопоставления. Генерирование гипотез обычно выполняется по схемам вывода, которые реализуются по правилам типа «ЕСЛИ-ТО».

Подход, основанный на знаниях, использует более обширные и конкретные знания о приложении. Такие знания позволяют решать сложные задачи распознавания изображений. С другой стороны, использование знаний, специфических для конкретного приложения, часто делает эти системы распознавания подходящими только для ограниченных областей. Тем не менее, для задач распознавания, включающих много атрибутов и классов, этот подход позволяет получать оптимальные решения за разумную цену при использовании соответствующих экспертных знаний о проблемной области, декомпозиции задачи

распознавания на подзадачи и представление данных и моделей на различных уровнях абстракции.

Этот подход вбирает многое из структурного распознавания, только языки, используемые в данном подходе, являются более мощными и гибкими.

Нейросетевой подход (The Neural-Network Approach) предназначен для построения параллельных компьютерных архитектур, называемых искусственными нейронными сетями, моделирующими биологические нейронные сети. Обычно каждый «нейрон» сети, реализует нелинейную вход/выходную характеристику и связывается с другими элементами посредством связей с весами. Алгоритмы обучения позволяют значения весов настраивать на характеристики входных данных. Были разработаны различные нейросетевые модели, каждая отличающаяся топологией сети, вход/выходными характеристиками нейронов и используемым алгоритмом обучения [Николаев и др., 2003].

При данном подходе, также как и при статистическом, не закладывается каких-либо конкретных знаний о прикладной области. В этом случае знания включаются в значения весов и в топологию сети. А, следовательно, нет возможности в явном виде использовать априорные знания о прикладной области.

2. Задача анализа изображений

В системах анализа изображений используются широкий спектр описаний: от набора признаков до структурных (символьных) представлений. В нашем случае под задачей полного анализа изображения подразумевается описание распознаваемых объектов и их расположение на изображении в терминах и в структурах языка прикладной онтологии. Результатом анализа должно быть описание ситуации на изображении, состоящей из экземпляров распознанных классов объектов с их означенными характеристиками и отношениями между ними, в разной степени «полноты» в зависимости от внешних критериев.

Построение описаний в онтологии (модели) предполагает наличие достаточно выразительных средств описания (модель, язык и форма представления описания изображений). Модель организуется в виде прикладной онтологии со структурами, каждая из которых соответствует обобщенному описанию некоторого класса объектов на изображении с его структурной организацией и взаимосвязями с другими классами. В качестве языка и формы представления описаний изображений в онтологии выбран семантический гиперграф ([Хахалин, 2009], [Баранович, 2011]), вершины которого соответствуют элементам описываемого изображения, а дуги – отношениям между этими элементами.

На входном изображении в общем случае задается не один образ, а целая ситуация и ее надо описать в терминах взаимосвязей означенных структур классов объектов.

В качестве предметной среды рассматривается мир плоских двумерных изображений объектов: планиметрические фигуры и «детские» рисунки(прямоугольники, трапеции, окружности и т.п., и кораблики, домики, паровозики и т.п.).

3. Гибридная схема анализа изображений

Цель разработки системы для решения задачи полного анализа изображений – совместить достаточно эффективный процесс распознавания с помощью программ, работающих непосредственно с изображениями, но дающих результат на «примитивном» концептуальном уровне (имя класса + параметры экземпляра) при использовании дискриминантных, статистических методов, и достаточно эффективный с языковой точки зрения процесс концептуального анализа изображений, используя методы, основанные на знаниях. Для этого мы выбрали двухуровневую систему: нижний уровень, который непосредственно работает с изображением, и верхний уровень, который работает с моделью изображения, но с мощным языком символического описания среды. Они работают последовательно с возможностью управления процессом распознавания на нижнем уровне с концептуального уровня (см. ниже).

3.1. Система статистического распознавания (PR-система)

Примеров систем статистического распознавания много (CSoft [RasterArts, 2008], IcadGT [IcadGT, 2007], MyScriptNotes [MyScriptNotes, 2007]). Для достаточно разнообразных и сложных сред (таблицы, контурные карты, схемы, архитектурные планы, печатные платы, чертежи по механике, аэрофотоснимки и т.д.) можно использовать первые две системы. Но специфические средства представления информации в этих системах, рассчитанные на конкретные области приложения, не позволяют достаточно простое использование в комплексе с концептуальной системой распознавания.

В качестве опоры для иллюстраций в выбранной предметной области мы используем реальную систему MyScriptNotes. Эта система решает стандартные задачи предварительной обработки изображений, включая кодирование, сглаживание, фильтрацию, сегментацию и т.д., избавляя от этих процедур верхний уровень гибридной системы. На вход программы MyScriptNotes подаются нарисованные геометрические рисунки. На выходе получаем для каждого класса распознаваемых объектов – имя класса, вероятность распознавания и значения геометрических параметров (заранее определенных), привязанных к полю изображения

(координатной системе). Кроме того дается альтернативная информация распознавания, когда приводятся возможные другие кандидаты на распознавание данного объекта с соответствующей вероятностью.

Множество распознаваемых классов объектов ограничивается перечнем планиметрических фигур: *отрезок прямой, треугольник, прямоугольник, ромб, дуга, окружность, эллипс* и др. + класс неизвестных объектов (рисунки произвольной, нераспознаваемой формы - *drawing*).

Для входного изображения прямоугольника и окружности (см. на рисунке 2, слева) в выходной информации представлены имена классов *rectangle* и *circle*, вероятность распознавания, значения соответствующих координат и возможные дополнительные кандидаты на распознавание (*parallelogram, drawing, circle* и др.). Справа на рисунке 2 дано синтезированное программой MyScriptNotes изображение по результатам распознавания.

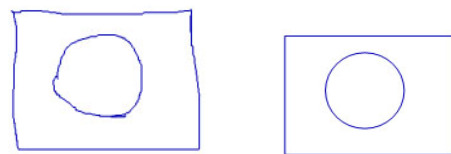


Рисунок 2 – Пример входного и синтезированного изображения

Пример более сложного изображения для гибридной системы распознавания приведен на рисунке 3. На нем представлены как объекты, распознаваемые программой MyScriptNotes (*прямоугольник, ромб* и др.), так и не распознаваемые на этом уровне классы объектов (*домик, кораблик, машинка*).

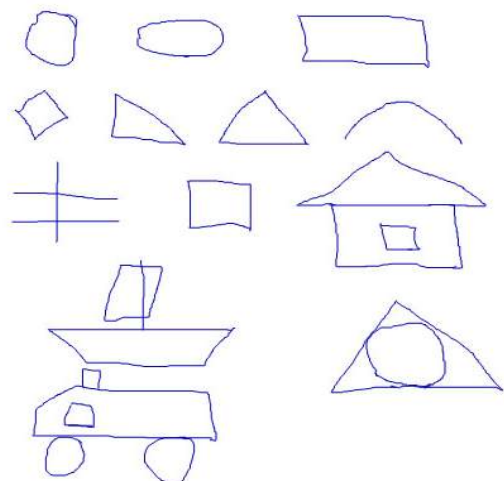


Рисунок 3– Представители объектов для распознавания

Для нераспознанных и нераспознаваемых PR-системой объектов как раз и служит верхний уровень гибридной системы.

3.2. Система символьного анализа изображений (КВ-система)

В качестве основы для системы анализа верхнего уровня выбран метод контекстуального целенаправленного анализа изображений, который подробно описан в [Курбатов и др., 2010] и [Khakhalin et al., 2012]. Подчеркнем только главные дополнения для этого уровня.

1. Анализ осуществляется не в поле реального изображения, а по модели изображения, полученной по результатам числового (статистического) распознавания. Первоначально анализ осуществляется по распознанным PR-системой классам объектов. Если в модели изображения присутствуют только «простые» объекты (из множества распознаваемых PR-системой), то функция КВ-системы заключается в означивании структур этих объектов из онтологии и в «наведении» геометрических отношений между ними, которые в неявном виде присутствуют в выходных данных PR-системы. Например, для рисунка 2 выходом КВ-системы будет описание «простой» ситуации в виде означенных структур «Прямоугольник» и «Окружность» с отношением «находится внутри».

Если же в модели изображения присутствуют не только те объекты, которые PR-система может распознавать, но и те, которые распознать не может, то КВ-система осуществляет поиск в онтологии структур «сложных» объектов и сопоставление с фрагментами модели изображения. Например, для объекта «кораблик», результат распознавания которого дан на рисунке 4, КВ-система, используя информацию о дополнительных кандидатах распознавания вместо нераспознаваемых (кандидаты подчеркнуты), выделит образ «трапеция». И далее, поскольку параллелограмм-«парус» и перпендикулярный отрезок-«мачта» распознаны, после процедуры «наведения» отношений сопоставит этому изображению структуру в онтологии с именем *кораблик*.

```
parallelogram 0.841063 453.52 495.446 672.868 491.344 765.673
137.967 546.324 142.07
drawing 0.5
triangle 0.31167 422.79 725.928 831.228 140.91 533.597 139.68
ellipse 0.290373 610.072 309.389 211.211 122.372 -1.14976
arc 0.290167 610.072 309.389 211.211 122.372 -1.14976 0 6.28319
line 0.731522 278.853 890.619 881.88 876.762
drawing 0.5
arc 0.100776 561.153 888.015 253.049 6.93823 -0.0208331
2.56482 6.28319
drawing 0.5
line 0.436261 192.309 655.127 287.242 894.925
arc 0.201896 232.866 775.533 118.11 9.32409 1.19305 3.09251
6.28319
line 0.834599 186.09 667.422 1004.87 650.702
drawing 0.5
arc 0.125702 585.368 655.953 392.703 6.51846 -0.00942964 -
1.11674 3.12932
drawing 0.5
line 0.423253 870.282 895.919 1011.53 654.31
arc 0.162266 936.106 773.071 125.247 7.98859 -1.02929 -0.589049
3.15386
line 0.957515 652.818 71.9868 653.825 669
drawing 0.5
```

Рисунок 4— Результат распознавания «кораблика»

2. В случае частичного сопоставления фрагмента модели изображения со структурой онтологии выдвигается гипотеза о дополнительном анализе на уровне PR-системы и на этот уровень КВ-система передает управляющую информацию для числового распознавания. Проиллюстрируем это на примере распознавания *домика*, представленного на рисунке 3. Если по фрагменту результатов числового распознавания синтезировать с помощью программы MyScriptNotes изображение, то получим фигуру, представленную на рисунке 5.

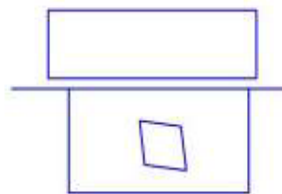


Рисунок 5— Синтезированное изображение по результатам распознавания PR-системы

Совершенно очевидно, что PR-системой была допущена ошибка: она распознала *крышу домика* не как *треугольник*, а как *прямоугольник*. Ошибка распознавания определяется, по-видимому, «наводками» при распознавании сложного изображения, которые затрудняют процесс правильной априорной сегментации. Доказательством тому служит эксперимент: если на вход программы подать изображение только *домика*, то MyScript Notes выдаст правильные результаты (*прямоугольник, параллелограмм, треугольник* с соответствующими параметрами).

В этом случае КВ-система выдвигает гипотезу о «дораспознавании» на реальном изображении. Для вызова соответствующих процедур PR-системы она формирует управляющую информацию, которая включает: что надо искать (это известно из онтологической структуры-гипотезы); где надо искать (вычисленный сегмент-«ареал» по модели изображения); с какими ограничениями, т.е. с необходимыми параметрами (из онтологической структуры).

После получения управляющей информации PR-система реально анализирует выделенную область и реализует процедуру распознавания только в этом сегменте, как будто на целом изображении присутствуют только компоненты этого объекта.

Следует отметить, что здесь проявляется более оптимальное решение задачи сегментации, когда сегментация осуществляется на основе частичных результатов распознавания нижнего уровня. В дальнейшем расширение этой стратегии можно использовать более «плотно», т.е. реализовать стратегию «сверху-вниз» (от онтологии к изображению) и при «стандартном» распознавании.

Если в результате повторного распознавания будет выделен объект класса *треугольник*, то в выходных данных КВ-системы для этого объекта

будет представлена онтологическая структура, как показано на рисунке 6 (означенные параметры выделены сплошными кружками).

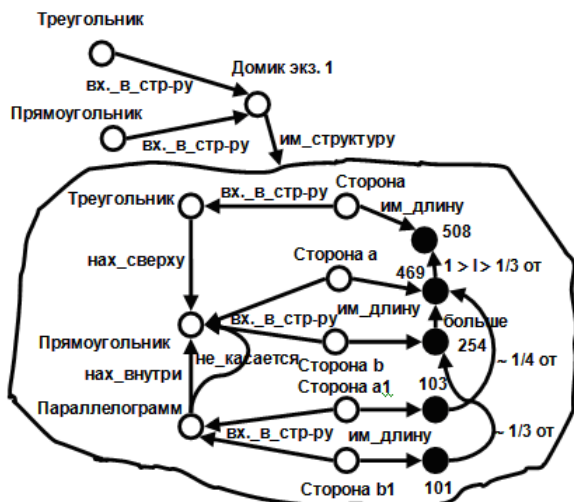


Рисунок 6– Означенная онтологическая структура объекта «домик»

3. Трансформация описания структуры образа для передачи описания в терминах «непроизводных объектов» на уровень числового распознавания.

В такой гибридной системе необходимо иметь средства трансформации из одного (символьного) представления в другое (числовое) представление. При передаче информации от KB-системы к PR-системе необходимо из онтологических описаний структур объектов уметь вычленять существенные признаки, образовывать из них кортежи и «придумывать» обучающее множество примеров и контр примеров. Контролировать этот процесс может пробное распознавание при наблюдении и сопровождении учителя. Для этого можно привлекать синтезатор изображений интегральной системы.

4. Обучение подсистем анализа изображений

Гибридная система работает в двух режимах: обучение (символьное и статистическое – разные методы) и собственно распознавание.

Для обучения PR-системы существует много методов обучения. Приведем краткое описание процедур обучения с учителем [Найденова и др., 2008].

Методы машинного обучения (MachineLearning) являются методами обучения классификациям объектов, представленных описаниями в признаковых пространствах. Цель обучения есть получение необходимых и достаточных правил, с помощью которых можно произвести классификацию новых объектов, сходных с теми, которые составляли обучающую выборку (обучение с учителем – supervised learning). При этом каждый обучающий пример (описание объекта) имеет метку, показывающую, к какому классу он принадлежит.

Ключевыми моментами машинного обучения являются:

1. Выбор и формирование признакового пространства;
2. Проверка гипотез о различимости/сходстве объектов и классов объектов; задание бинарных операций сходства-различия объектов; задание мер сходства-различия для классов объектов;
3. Формирование обучающей выборки;
4. Формирование контрольной выборки;
5. Адекватный выбор алгоритма обучения.

Если выбор признакового пространства определяет задачу обучения, главным образом, содержательно, то формирование обучающей и контрольной выборок отвечает за точность, быстроту и эффективность обучения. С помощью правильно выбранных примеров можно направлять процесс обучения. Пошаговые процедуры обучения и выбор последовательности примеров (от простого к сложному) позволяет также минимизировать число примеров, необходимых для обучения. Контрольная выборка необходима не только для проверки правильности работы классификатора, но и для целенаправленного «доучивания» классификатора, его исправления, модификации, придания ему требуемых свойств.

К широко известным методам относятся вероятностные методы машинного обучения.

Метод Байеса определяет вероятность наступления события в условиях, когда наблюдается лишь некоторая частичная информация о событиях, например, по наблюдаемым признакам определяется принадлежность некоторого объекта к одному из заданных классов.

Метод опорных векторов (Support Vector Machine) относится к линейным разделяющим методам. Каждый вектор признаков (объект) представляется точкой в многомерном пространстве признаков. При заданной классификации объектов строятся две параллельные гиперплоскости (границы), разделяющие объекты разных классов, таким образом, что расстояние между этими гиперплоскостями максимизируется. Примеры, расположенные вдоль гиперплоскостей, называют опорными векторами. Значения признаков – вещественные числа.

К вероятностной методам относится скрытая Марковская модель (Hidden Markov chains). Это статическая модель, имитирующая некоторый последовательный процесс, в котором на наблюдаемые переменные оказывают влияния скрытые состояния. Переход из одного состояния в другое происходит с некоторой вероятностью. По последовательности наблюдений можно получить информацию о последовательности состояний.

Для обучения KB-системы можно применить логический метод и метод концептуального обучения [Найденова, 2009].

К логическим методам относятся методы, при

которых в процессе обучения строятся логические правила в форме продукций или в форме решающих деревьев, в узлах которых проверяются значения отобранных при обучении признаков и принимается решение о разбиении объектов на подклассы. Логические методы работают на описаниях, которые имеют как символьную, так и целочисленную природу, или в пространстве булевых признаков. С этой точки зрения эти методы можно отнести к методам концептуальных классификаций или методам построения правил рассуждений на уровне концептов. Но формирование правил в практических задачах часто влечет за собой громадную ручную работу и включает в себе творческую и неавтоматизированную компоненту методов машинного обучения.

К методам концептуального обучения (извлечение концептов и их иерархических отношений) относится направление машинного обучения, называемое формальным концептуальным анализом. Концептуальное обучение представляет собой особый класс методов, основанных на порождении и использовании концептуальных знаний, элементами которых являются объекты, атрибуты (значения атрибутов), классификации (разбиения объектов на классы) и связи между ними. Эти связи выражаются через имплицитивные отношения вида «объект ↔ класс», «объект ↔ значение атрибута (свойство)», «значение атрибута (ов) ↔ класс», «подкласс ↔ класс».

Индуктивный вывод концептуальных знаний сводится к задаче вывода хороших диагностических тестов для заданных классификаций объектов. Хороший диагностический тест определяется как совокупность атрибутов (значений атрибутов), которая порождает наилучшую аппроксимацию заданной классификации (разбиения) некоторого заданного множества объектов. К поиску хороших тестов сводятся почти все хорошо известные логические методы машинного обучения. Эти методы охватывают вывод имплицитивных и функциональных зависимостей из данных, вывод логических правил (правила «если-то», «грубые» множества), вывод ассоциативных правил, конструирование решающих деревьев, извлечение иерархических концептуальных классификаций и ряд других. Все перечисленные задачи отличаются по форме представления выделяемых отношений включения на множестве всех подмножеств объектов и решаются с использованием одной и той же структуры данных и одних и тех же алгоритмов.

Подход *lazylearning* основывается на гипотезе, что решение когнитивных задач (обучение языку в частности) базируется на построении выводов на основе аналогий, а не на основе абстрактных правил, полученных из экспериментов. Этот подход используется в различных дисциплинах искусственного интеллекта и лежит в основе таких методов, как выводы на основе сходства, выводы на

основе примеров, выводы на основе аналогии, вывод на основе прецедентов (*case-based reasoning*) и пр.

При "ленивом" обучении обучающие примеры добавляются в память без обобщений и реструктурирования. Сходство нового примера с остальными вычисляется по метрике сходства, и правило большинства сходных примеров используется как базовое для предсказания категории нового примера.

Основными методами подхода *greedy learning* являются обучение на основе деревьев решений и индуктивного вывода. Обучение на основе деревьев решений основывается на предположении, что сходство примеров может быть использовано для автоматического построения деревьев решений, на основе которых порождаются обобщения и объяснения. Целью индуктивного вывода является построение ограниченного множества интерпретируемых правил на основе обучающих примеров или деревьев решений.

Обучение KB-системы осуществляется с помощью: 1) разработчика онтологии; 2) пользователя посредством ЕЯ-интерфейса и при посредстве системы синтеза изображений. Разработчик вводит новые описания в онтологию непосредственно на языке представления знаний (язык семантического гиперграфа). Пользователь может вводить новые описания посредством текстов на ЕЯ. Но для проверки корректности введенных описаний и в том и другом случае используется система онтологического синтеза изображений, по результатам работы которой разработчик и пользователь корректируют свои описания.

Для иллюстрации этого процесса на рисунке 7 приведено множество синтезируемых изображений, по которым видно, что в онтологической структуре объекта «домик» (в отличие от рисунка 6) не заданы некоторые ограничения по параметрам подобъектов (например, соотношение длин сторон «сруба домика» и «крыши»).

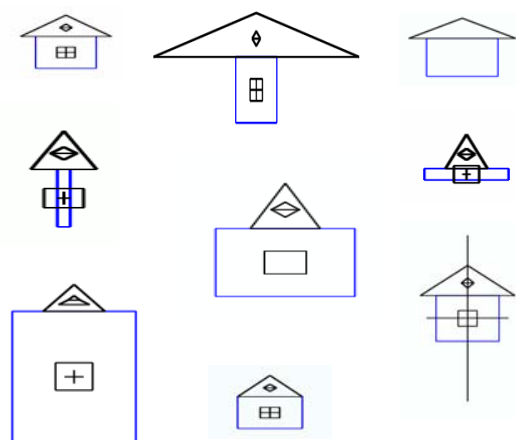


Рисунок 7 - Множество синтезируемых изображений

ЗАКЛЮЧЕНИЕ

Подчеркнем особенности гибридной системы анализа (двумерных) изображений:

- Интеграция числового (статистического) PR-подхода с KB-подходом, и использование преимуществ и того и другого.
- Новые + старые методы обучения (индуктивные и дедуктивные методы обучения). Машинные методы обучения концептуальным знаниям представляют собой модель правдоподобных индуктивных и дедуктивных рассуждений, в которых вывод знаний и их использование неотделимы друг от друга.
- Процесс анализа изображений на PR-уровне может управляться KB-системой.
- Трансформация структурного описания класса образов (объектов) в атрибутивное описание и передача этой информации с верхнего на нижний уровень распознавания.
- Для верификации процессов обучения и самого распознавания привлечение системы концептуального синтеза изображений.
- Автоматизация процесса пополнения онтологии с использованием процедуры верификации (синтеза изображений): а) за счет введения структурных описаний разработчиком непосредственно на языке семантических гиперграфов; б) за счет введения описаний пользователем на проблемно ориентированном ЕЯ.
- Интеграция анализа/синтеза изображений с ЕЯ-анализатором (лингвистическим транслятором).
- Онтологические описания ситуаций, получаемые от изображений или с текстом на ЕЯ, представляются на общем языке представления знаний – семантическом гиперграфе.
- Достаточно мощный и гибкий язык представления знаний – семантический гиперграф, как развитие семантических сетей.
- Процедура сегментации изображения на фрагменты, соответствующие распознаваемым объектам, в отличие от априорной сегментации носит «осмысленный» целенаправленный характер, т.к. она проводится на основе частичных результатов распознавания, выдаваемых PR-системой, в рамках проверяемых гипотез, которые формируются KB-системой.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Баранович, 2011] Баранович, А.Е. Семантико-хроматические гипертопосети: унифицированная модель представления знаний // OSTIS-2011. Труды конференции. С. 71-86
- [Горелик и др., 2004] Горелик, А.Л., Скрипкин, В. А. Методы распознавания. Учебное пособие. – М.: Высшая Школа, 2004. - 261 с.
- [Животников и др., 2005] Животников, Г.С., Пытьев, Ю.П., Фаломкин, И.И. Итеративный морфологический алгоритм локализации неизвестного объекта на изображении в реальном времени // ММРО-2005. Труды конференции. С. 322-325
- [Журавлёв, 2000] Журавлёв, Ю.И. Распознавание образов и анализ изображений / Ю.И. Журавлёв, И.Б. Гуревич // Искусственный интеллект: Модели и методы. Т. 1. – М.: Радио и связь, 2000. – 310 с.

[Катыс, 1990] Катыс, Г.П. Обработка визуальной информации. – М.: Машиностроение, 1990. – 320 с.

[Курбатов и др., 2010] Курбатов, С.С., Найденова, К.А., Хахалин, Г.К. О схеме взаимодействия в комплексе «анализ и синтез естественного языка и изображений» // КИИ-2010. Труды конференции. С. 234-242

[Курбатов и др., 2011] Курбатов, С.С., Найденова, К.А., Хахалин, Г.К. Интегрирование интеллектуальных систем анализа/синтеза изображений и текста: контуры проекта INTEGRO // OSTIS-2011. Труды конференции. С. 213-232

[Найденова и др., 2008] Найденова, К.А., Невзорова, О.А. Машинное обучение в задачах обработки естественного языка: обзор современного состояния исследований // Новости Казанского университета, 2008, № 1, С. 3-24

[Николаев и др., 2003] Николаев, А.Б., Фоминых, И.Б. Нейросетевые методы анализа и обработки данных. Учебное пособие. – М.: МАДИ (ТГУ), 2003, 95 с.

[Новикова, 2008] Новикова, Н.М. Структурное распознавание образов. Учебно-методическое пособие для вузов. – Воронеж: Издательско-полиграфический центр Воронежского ГУ, 2008. – 29 с.

[Павлидис, 1986] Павлидис, Т. Алгоритмы машинной графики и обработки изображений / Т. Павлидис; пер. с англ. Н.В. Петрова; под ред. М.С. Гуревича. – М.: Радио и связь, 1986. – 400 с.

[Пытьев, 2007] Пытьев, Ю.П. Математические методы и адаптивные алгоритмы эмпирического построения теоретико-возможностной модели стохастического объекта // ММРО-13. Труды конференции. С.54–56.

[Хахалин, 2009] Хахалин, Г.К. Прикладная онтология на языке гиперграфов // ЗОНТ-09. Труды конференции. С. 223-231

[Bhanu et al., 2005] Bhanu Bir, PengJing, Huang Thomas and Draper Bruce A.. Introduction to the Special Issue on Learning in Computer Vision and Pattern Recognition // IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics, vol. 35, no. 3, June 2005, pp. 391-395

[IcadGT, 2007] Режим доступа: <http://www.recsoft.ru>. – [Электронный ресурс].

[Khakhalin et al., 2012] Khakhalin, G., Kurbatov, S., Naidenova, K., and Lobzin, A. Integration of the Image and NL-text Analysis/Synthesis Systems. In book: “Intelligent Data Analysis for Real-Life Applications: Theory and Practice” (Ed. Rafael Magdalena et al.). N.Y., USA: IGI Global, 2012 (в печати).

[MyScriptNotes, 2007] Режим доступа: <http://www.visionobjects.com/en/webstore/myscript-studio/description/>. – [Электронный ресурс].

[Naidenova, 2009] Naidenova, X. A. Machine Learning Methods for Commonsense Reasoning Processes: Interactive Models. N.Y., USA: IGI Global. 2009.

[Pellegritti et al., 1994] Pellegritti Paolo, Roli Fabio, Serpico Sebastian B., and Vemazza Gianni. Supervised Learning of Descriptions for Image Recognition Purposes // IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 1. January 1994, pp. 92- 98

[Raster Arts, 2008] Режим доступа: <http://www.csoft.ru>. – [Электронный ресурс].

[Roli et al., 1996] Roli Fabio, Serpico Sebastiano B., and Vemazza Gianni. A Hybrid System for Two-Dimensional Image Recognition // Proceedings of the IEEE, VOL. 84, NO. II, November 1996, pp. 1659-1680

THE HYBRID CIRCUIT OF THE IMAGE ANALYSIS

S.S. Curbatov, A.P. Lobzin, K.A. Naidenova, G.K. Khakhalin

An image analysis scheme connecting the advantages of numerical and symbolic methods of pattern recognition is examined. The crossbreeding of two approaches with the use of applied ontology in an integrated system of linguistic translator and synthesizer of images makes it possible to achieve the higher level of understanding images. Semantic hypergraph is used as the language for knowledge representation in the framework of symbolic methods of image processing.



УДК 004.822:514

РАЗРАБОТКА ИНТЕРФЕЙСА ИНТЕЛЛЕКТУАЛЬНОГО ПОМОЩНИКА ПРОЕКТАНТА

Боргест Н.М., Чернов Р.В., Шустова Д.В.

Самарский государственный аэрокосмический университет имени академика С.П. Королева
(национальный исследовательский университет), г. Самара, Россия

borgest@yandex.ru

necroromnt@yandex.ru

ShustovaDV@yandex.ru

В работе приводятся результаты разработки многоэкранного и многооконного интерфейса интеллектуального помощника проектанта, реализуемого в предметной области предварительного проектирования самолета. Реализация подобного интерфейса позволит наглядно представить проектанту весь ход выполнения работ над проектом. Показана возможность в перспективе автоматического решения типовых проектных задач.

Ключевые слова: интеллектуальный помощник, интерфейс, проектирование, робот.

ВВЕДЕНИЕ

Успехи в области распознавания образов, речевых технологий, разработки баз данных, промышленных САПР позволили приступить к созданию интеллектуальных помощников

развитым многоэкранным и многооконным интерфейсом приближают наступление эры интеллектуальных систем в проектировании.

1. Робот-конструктор

Интеллектуальный помощник проектанта в

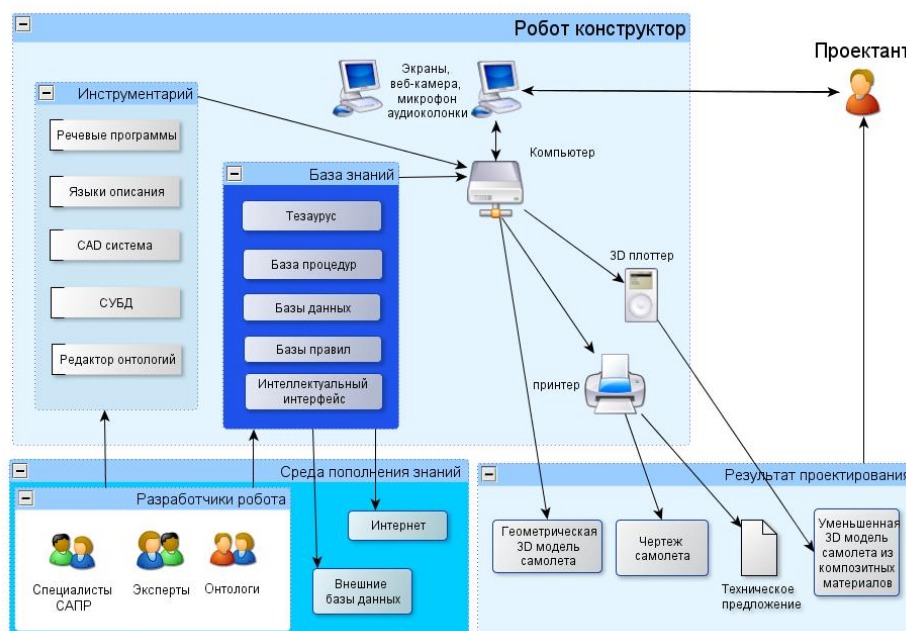


Рисунок 1 – Структурная схема робота конструктора

проектанта. Тезаурус предметной области, формализованный опыт проектирования, мощные базы данных уже созданных проектов в сочетании с

пределе может выполнять функции робота конструктора, способного в автоматическом режиме решать поставленные перед ним проектные задачи.

На рисунке 1 показаны основные компоненты работы. Главной сущностью взаимодействия с роботом является проектант. Ориентируясь, с одной стороны, на потребителя, который в большей степени способен ставить проектные задачи, чем их решать, с другой стороны, на специалиста-предметника, который в большей степени озабочен содержательной стороной проектной задачи, чем техническими вопросами ее реализации, можно предположить, что перспективным способом коммуникации с роботом будет естественный язык. Это определяет важную компоненту в работе - это наличие речевых программ, их связь с тезаурусом.

Сам робот – это компьютер с периферийными устройствами, инструментари, включающие в себя названные речевые программы, языки описания, СУБД, САД системы, редакторы онтологий, и база знаний, как совокупность тезауруса, баз данных, правил и процедур, со сценариями проектирования.

Важной компонентой в работе является среда пополнения знаний. Организация ее автоматизированного функционирования – самостоятельная задача для интеллектуальных систем разного класса.

Результатом проектирования традиционно является чертеж объекта, его 3D модель, а также «отпечатанная» на 3D плоттере модель спроектированного объекта, которую можно предоставить заказчику для обозрения.

Особое место в работе – многоэкранный и многооконный интерфейс, который как раз и является предметом исследования в данной работе. Но прежде, несколько слов о матрице проекта.

2. Матрица проекта

Условно проектирование какого-либо объекта или системы можно представить как процесс заполнения некой абстрактной матрицы, которая по её наполнению будет содержать всё описание этого

объекта или системы.

На начальном этапе матрица проекта пуста и контурно представляет собой некое подобие морфологической таблицы, содержащей все возможные структурные варианты будущего объекта.

Далее, после получения технического задания на проектирование, начинается заполнение этой матрицы данными, первые из которых как раз и являются данные самого технического задания.

Проектирование процесс всегда итерационный. Это связано с тем, что начиная от этапа постановки задачи, до принятия окончательных решений невозможно определить и согласовать все взаимосвязанные и взаимозависимые данные между собой. Отсюда потребность для проектанта иметь возможность видеть весь процесс проектирования объекта или в нашем случае заполнения матрицы проекта.

В случае автоматического проектирования, когда робот «самостоятельно», на основе заложенных в него правил, схем, предпочтений и алгоритмов принимает решения, также важно видеть как и на каком основании осуществляется принятие тех или иных решений. Метафористически или образно можно говорить о необходимости или желательности не только охватить весь процесс проектирования, но и иметь возможность наблюдать, словно через сканер или рентген, внутреннюю «кухню» процесса.

Реализация демонстрации этой внутренней «кухни» возможно лишь тогда, когда все принятые ответственные решения будут доступны для проектанта. В нашем случае налицо потребность в многоэкранном и многооконном интерфейсе, который позволит не только следить, но и при необходимости активно вмешиваться в процесс проектирования (заполнения матрицы проекта), наблюдая его в динамике.

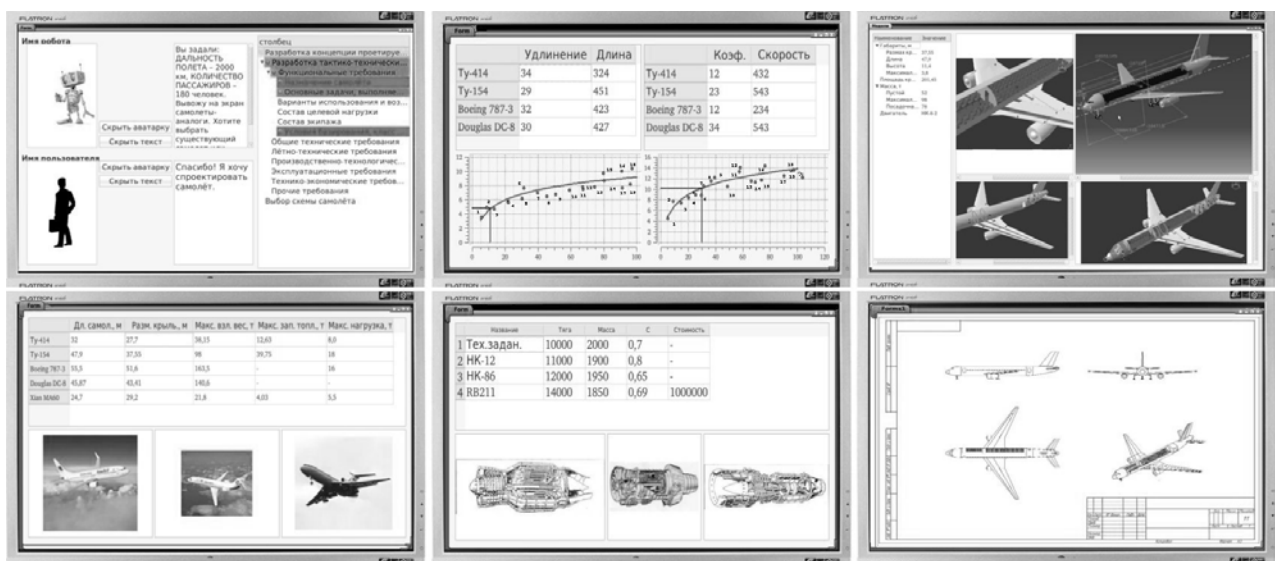


Рисунок 2 – Многоэкранный интерфейс робота конструктора самолетов

3. Многоэкранный интерфейс

Разработке пользовательских интеллектуальных интерфейсов, анализа данных и семантики текстовой информации посвящены работы ученых, работающих в Киеве, Минске, Владивостоке, Ульяновске и др. городах ([Sosnin, 2011], [Грибова, 2011], [Голенков, 2011], [Гладун, 2011] и др.). Представленный ниже интерфейс является развитием работ [Боргест, 2011a], [Боргест, 2011b] и базируется, в том числе и на работах указанных выше авторов.

На рисунке 2 показана панель многоэкрannого и многооконного интерфейса интеллектуального помощника проектанта для предметной области предварительное проектирование самолета.

Верхний слева экран – экран, на котором идентифицируются робот и проектант, дублируется в текстовом виде диалог робота и проектанта, а также в иерархическом виде представлен сценарий проектирования в виде перечня решаемых последовательно задач (см. рисунок 3).

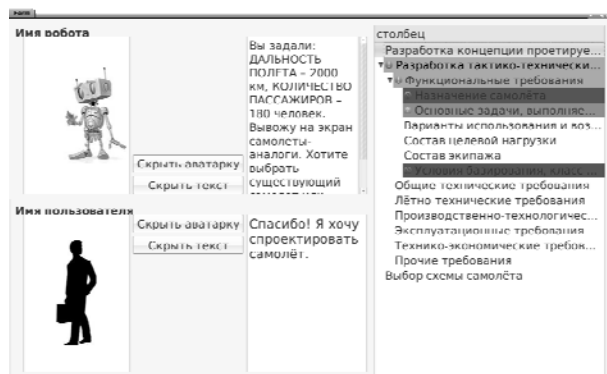


Рисунок 3 – Экран диалога

На рисунке 4 представлен экран прототипов. После формулировки проектантом постановки задачи и идентификации ее роботом, последний осуществляет поиск возможных решений в базе данных. Вполне вероятен сценарий, когда робот может не дойти до этапа непосредственно проектных работ, а найти приемлемое решение в своей базе данных готовых проектов и проектант может удовлетвориться этим решением. На рисунке 4 показаны аналогичные проекты самолетов, которые по заданным техническим условиям близки к требуемому самолету.

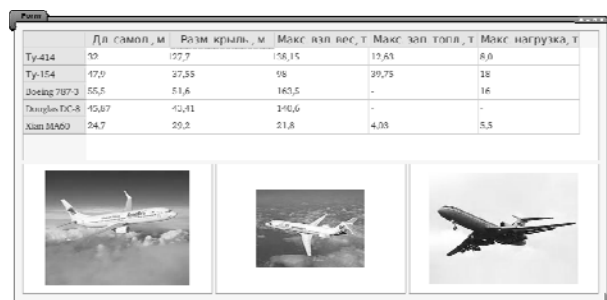


Рисунок 4 – Экран прототипов

На рисунке 5 показан типовой экран принятия

решений на основе обработки статистических данных по тем или иным характеристикам объекта проектирования. Заложенный в робот сценарий проектирования «предполагает» принятие решений по некоторым характеристикам, используя при этом знания физической природы закономерностей и взаимосвязи параметров. Проектант, наблюдая принятые на основе обработанной статистики значения параметров и понимая суть решаемой задачи, имеет возможность внести свои коррективы.

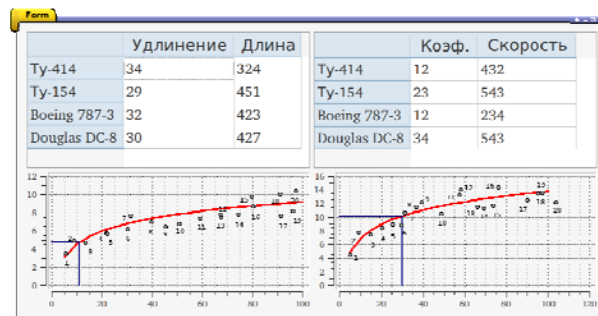


Рисунок 5 – Экран анализа статистики

Важной компонентой проектирования сложных систем является рациональный подбор комплектующих, готовых подсистем и узлов. Для самолетов и других транспортных средств такой типовой задачей является выбор двигателя, удовлетворяющего многим показателям и техническим требованиям. На рисунке 6 представлен экран выбора двигателя под проектируемый самолет на основе анализа технических и стоимостных характеристик.

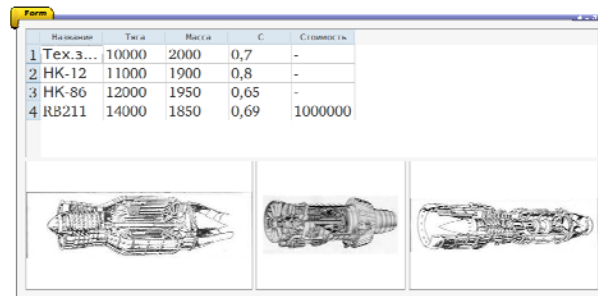


Рисунок 6 – Экран выбора комплектующих

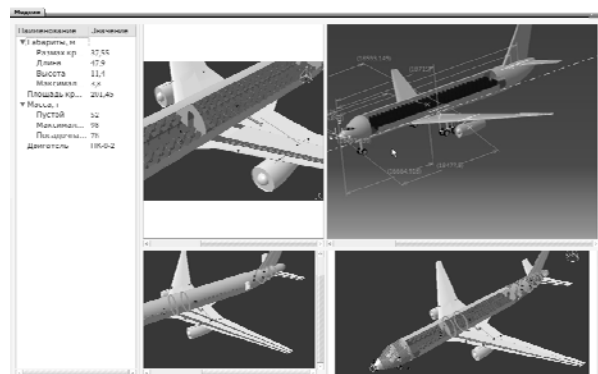


Рисунок 7 – Экран построения 3D модели

Формирование 3D модели технического объекта осуществляется в CAD системе, в том числе в автоматическом режиме, на основе полученных

значений основных параметров после выполнения необходимых расчетных проектных процедур.

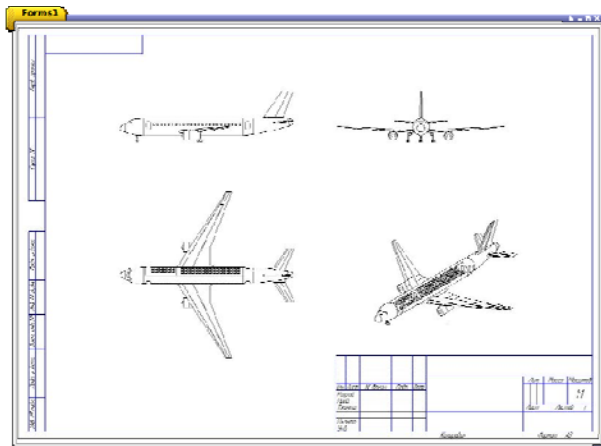


Рисунок 8 – Экран получения чертежа

Рисунок 7 иллюстрирует работу автоматической центровки и компоновки самолета, которая завершается получением 3D модели объекта и чертежа общего вида самолета (см. рисунок 8).

Представленный далеко не полный перечень экранов и окон в интерфейсе интеллектуального помощника позволяет сократить время и повысить качество проектирования.

4. Обеспечение открытости

Интеллектуальный помощник способен взаимодействовать с самыми распространенными инженерными программами, такими как Catia, Ansys. Большинство из них являются модульными, и каждый модуль может работать самостоятельно, а так же могут запускаться и использоваться без собственного пользовательского интерфейса.

Для реализации взаимодействия в разрабатываемом интерфейсе интеллектуального помощника с другими программами планируется использовать язык скриптов, плагины и подключаемые исполняемые файлы (модули). В виде подключаемых модулей возможно реализовать дополнительные диалоговые окна, функции, парсеры для распознавания файлов разных форматов и баз данных, конвертеры файлов интеллектуального помощника в файлы для других программ, средства передачи данных напрямую в независимую программу. Сценарий работы с модулями можно будет запрограммировать используя скрипты. Как возможный язык скриптов может быть взят Perl, который имеет широкий спектр применения. Для языка скриптов будут разработаны объекты, процедуры и функции взаимодействия с интеллектуальным помощником.

В базах данных храниться информация о пользователях, тезаурус (словарь), готовые детали, узлы и агрегаты, списки доступных скриптов, сценариев и плагинов. Сам проект сохраняется в виде базы данных или в виде файла на языке разметки .xml. Сохранение в виде базы данных

позволит упростить работу с массивными проектами и их объединение в единый проект, а также защитить проект от несанкционированного доступа. Сохранение в виде .xml файлов позволит читать сохранённые проекты без использования интеллектуального помощника.

ЗАКЛЮЧЕНИЕ

Разработанный прототип многоэкранного и многооконного интерфейса интеллектуального помощника проектанта, тестируется в предметной области предварительного проектирования самолета. Реализация подобного интерфейса позволит наглядно представить проектанту весь ход выполнения работ над проектом. Показана возможность в перспективе автоматического решения типовых проектных задач.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Боргест, 2011a] Боргест, Н.М. Модель интеллектуального интерфейса робота «конструктор самолета» / Н.М. Боргест // XI международная научная конференция имени Т.А. Таран «Интеллектуальный анализ информации ИАИ-2011», Киев, Просвіта, 2011, С. 297-302
- [Боргест, 2011b] Боргест, Н.М. Онтология проектирования самолета/ Н.М. Боргест // Искусственный интеллект, 2011, №4. С.260-265.
- [Гладун, 2011] Гладун, В.П. Когнитивные структуры для машинного анализа семантически-сложных текстов / В.П. Гладун, Л.А. Святогор// XI международная научная конференция имени Т.А. Таран «Интеллектуальный анализ информации ИАИ-2011», Киев, Просвіта, 2011, С.4-14.
- [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, Минск, БГУИР, 2011, С.21-58.
- [Грибова, 2011] Грибова, В.В. Процессы управления интеллектуальными системами/ В. В. Грибова, А.С. Клещев // Онтология проектирования, 2011, № 1. С.22-31.
- [Sosnin, 2011] Sosnin, P.I. Precedent-oriented shell for personal expert systems. Interactive Systems and Technologies: the Problems of Human-Computer Interaction. – Collection of scientific papers. Ulyanovsk: UISTU, 2011, P.10-33

INTERFACE OF INTELLECTUAL ASSISTANT OF DESIGNER

Borgest N.M., Chernov R.V., Shustova D.V.

Samara State Aerospace University named after academician Korolev S.P. (National Research University), Samara, Russia

borgest@yandex.ru

necromont@yandex.ru

ShustovaDV@yandex.ru

The paper presents results of development and multi-window interface multi-screen intelligent assistant of the designer realized by in the subject area the preliminary design of aircraft. Implementation of this interface allows designers to visualize the entire progress of the work on the project. The possibility to run automatically solves common design problems.



УДК 004.822:514

СЕМАНТИЧЕСКИЕ МОДЕЛИ МУЛЬТИМОДАЛЬНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ И СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ ИХ ПРОЕКТИРОВАНИЯ

Корончик Д. Н.

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

denis.koronchik@gmail.com

В работе описана технология проектирования пользовательских интерфейсов интеллектуальных систем. Приводится описание семантической модели пользовательских интерфейсов, которая является основной описываемой технологии. Предложенная в данной работе технология позволяет проектировать мультимодальные пользовательские интерфейсы на основе готовых совместимых компонентов.

Ключевые слова: интеллектуальный пользовательский интерфейс, мультимодальный пользовательский интерфейс, семантическая модель, семантическая технология.

ВВЕДЕНИЕ

Эффективность использования программной системы зависит от ее пользовательского интерфейса. В большинстве случаев разработка пользовательского интерфейса в современных системах отнимает большую часть времени затрачиваемого на разработку всей системы [Myers В.А., 1992]. Трудоемкость разработки обусловлена не столько сложностью пользовательского интерфейса, сколько отсутствием хорошо продуманных технологий их проектирования.

Проектируемые с помощью современных технологий, пользовательские интерфейсы представляют собой достаточно сложные системы. Основной проблемой в них является то, что пользователю, имеющему низкий уровень квалификации, сложно работать с ними. 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. Кроме того для их освоения необходимо читать большое количество документации.

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 семантической технологии проектирования пользовательских интерфейсов

В основе предлагаемой семантической технологии проектирования пользовательских интерфейсов, лежат следующие принципы:

- разработка унифицированной модели пользовательских интерфейсов согласованной с общей теорией интеллектуальных систем (что не только сокращает сроки разработки, но и упрощает интеграцию пользовательского интерфейса в состав системы). В основе предлагаемой модели пользовательских интерфейсов лежит семантическая модель, основу которой составляет унифицированное представление информации с помощью SC-кода, описанного в [Голенков и др, 2001];

- повторное (многократное) использование уже разработанных фрагментов пользовательских интерфейсов (что обеспечивает не только сокращение сроков разработки благодаря многократному повторному использованию типовых компонентов пользовательских интерфейсов, но и снижает требования к квалификации разработчиков, поскольку это переводит проектирование на более высокий уровень – на уровень модульного сборочного проектирования). Фрагменты пользовательского интерфейса, которые являются достаточно унифицированными и могут быть использованы многократно в различных системах, называются компонентами пользовательского интерфейса. Для хранения, компонентов используется библиотека компонентов пользовательских интерфейсов;
- повышение уровня автоматизации и последующей интеллектуализации процесса проектирования (что сокращает сроки разработки, упрощает интеграцию пользовательского интерфейса в состав интеллектуальной системы). Для проектирования пользовательских интерфейсов используются инструментальные средства визуального проектирования, а также средства верификации и отладки, которые также являются интеллектуальными системами;
- создание интеллектуальной help-системы обеспечивающей консультацию разработчиков пользовательских интерфейсов по всем аспектам технологии их проектирования (что не только снижает сроки проектирования, но и снижает требования, предъявляемые к начальной квалификации разработчиков). Интеллектуальная help-система является sc-системой, основной задачей которой является формирование у разработчика навыков по проектированию пользовательских интерфейсов.

Таким образом, технология проектирования пользовательских интерфейсов включает в себя:

- семантическую модель пользовательских интерфейсов;
- библиотеку совместимых компонентов;
- инструментальные средства проектирования пользовательских интерфейсов:
 - интеллектуальные средства визуального проектирования;
 - интеллектуальные средства отладки и верификации;
- методику проектирования пользовательских интерфейсов;
- методику обучения проектированию пользовательских интерфейсов;
- help-систему по проектированию пользовательских интерфейсов.

2. Семантическая модель мультимодального пользовательского интерфейса

Основной частью технологии проектирования пользовательских интерфейсов является семантическая модель. Она описывает принципы, которые лежат в основе проектируемых интерфейсов, что позволяет унифицировать их. В основу семантической модели положены следующие принципы:

- 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 (базе 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.

Выделяются 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-операций.

Графический интерфейс интеллектуальных систем представляет собой мультимодальный оконный интерфейс. Под окном будем понимать скроллируемую область экрана, которая ограничена

прямоугольником произвольного размера.

Взаимодействие пользователя с системой осуществляется в рамках главного окна. Главное окно принадлежит к множеству *sc.g-окно* (окна, внутри которых для визуализации знаний используется SCg-код). В рамках главного окна могут присутствовать другие окна. Их будем называть дочерними окнами. Визуализируются они с помощью *sc.g-рамки* (рисунок 1) внутри которых отображается содержимое окна на одном из внешних языков.



Рисунок 1 – Пример изображения *sc.g-рамки*

С помощью *sc.g-рамки* отображаются окна в развернутом виде. В свернутом виде они изображаются с помощью *sc.g-узла* с квадратом в правом нижнем углу. При работе с системой пользователь может создавать окна, относящиеся к различным классам (*sc.g-окна*, *видео-окна* и т.д.). Такие окна будем называть дочерними окнами. Они являются частью *sc.g-конструкции* в рамках некоторого *sc.g-окна* (в частности, главного окна). Главное окно не может быть дочерним окном по отношению к другим окнам. И одно окно не может иметь несколько родительских окон. Таким образом, они образуют древовидную иерархию, которая, в рамках базы знаний пользовательского интерфейса, задается с помощью отношения *дочернее окно** (рисунок 2).

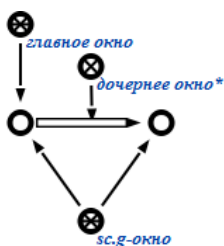


Рисунок 2 – Пример использования отношения *дочернее окно**

Графический пользовательский интерфейс строится с использованием уже разработанных компонентов. К компонентам вывода информационных конструкций относятся просмотрщики, которые в свою очередь могут быть двух типов:

- просмотрщики содержимого *sc-ссылок*. Позволяют просматривать содержимое *sc-ссылок* (файлов) записанное в некотором формате (не SC-код);
- просмотрщики фрагментов базы знаний представленных с помощью SC-кода. Позволяют просматривать с помощью внешнего языка информационные конструкции, представленные

в SC-коде (к примеру, фрагмент базы знаний, описывающий геометрический чертеж с помощью SC-кода может изображаться в виде *sc.g-текста* и в виде геометрического чертежа).

К компонентам ввода информационных конструкций относятся редакторы. Они позволяют редактировать содержимое некоторой *sc-ссылки*. Главное окно представляет собой просмотрщик фрагмента базы знаний с помощью SCg-кода и набора команд редактирования базы знаний. Другими словами при инициировании команд редактирования в рамках главного окна происходит редактирование не на уровне внешнего представления, а напрямую в базе знаний, на уровне SC-кода, а эти изменения отображаются с помощью просмотрщика.

При таком подходе основной проблемой является проблема интеграции компонентов между собой. Для ее решения предлагается осуществлять взаимодействие между компонентами через базу знаний. Таким образом, взаимодействуя между собой, компоненты могут лишь использовать общие ключевые узлы (понятия) в базе знаний. Такой способ интеграции компонентов позволяет разрабатывать их параллельно с минимальными зависимостями, что значительно сокращает сроки проектирования.

Диалог пользователя с системой осуществляется с использованием SCg-кода. Пользователь формирует сообщения системе, явно рисуя их с помощью SCg-кода (рисунок 3) и иницируя команду погрузки в базу знаний. Такой способ формирования сообщений назовем базовым. Ответы, как и вопросы, выводятся с помощью SCg-кода (в виде *sc.g-конструкций*). Частным видом ответа является *sc.g-конструкция*, состоящая из одной *sc.g-рамки*, содержимое которой представляется на каком-то внешнем языке. Другими словами диалог сводится к обмену сообщениями (представленными с помощью SCg-кода) между пользователем и системой.

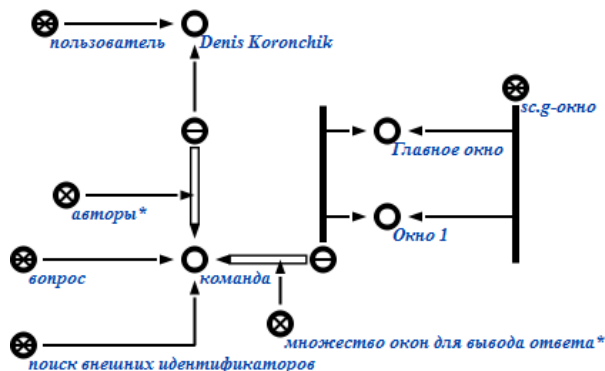


Рисунок 3 – Пример сформированного вопроса с помощью SCg-кода.

На рисунке 3, задается вопрос: *Какие внешние идентификаторы имеются у узла с основным идентификатором "команда"*. Помимо самой сути вопроса, к нему еще добавляется информация об

авторе вопроса (в нашем случае это пользователь, поэтому ответ необходимо будет выдать ему). Кроме этого дополнительно указывается множество окон, в которые необходимо вывести ответ (в качестве таких окон могут выступать окна не принадлежащие множеству *sc.g*-окон, что дает возможность получать ответ на любом внешнем языке). Таким же образом формируются различного рода команды (сворачивание, разворачивание, создание, удаление окна и т. д.). Такой способ формирования команд и вопросов является базовым и универсальным.

В процессе формирования сообщений таким способом, пользователю приходится выполнять часто повторяющиеся элементарные пользовательские действия (самые простейшие манипуляции с устройствами ввода: нажатие клавиши, перемещение мыши и т.д.). Очевидно, что постоянный набор сообщений вручную с помощью *SCg*-кода отнимает много времени и не является очень эффективным, по причине выполнения большого количества повторяющихся элементарных пользовательских действий. Можно ускорить процесс набора сообщений, за счет улучшения и расширения команд редактирования *sc.g*-конструкций. Но это не даст значительного прироста, потому что при формировании сообщения, необходимо будет указывать дополнительную информацию (к примеру, для каждого вопроса надо указывать авторов и множество окон для вывода).

Введем в модель такое понятие как пользовательская команда. Пользовательская команда - последовательность элементарных пользовательских действий, инициирующая некоторое действие системы. К элементарным действиям пользователя относятся: перемещение мыши, нажатие клавиш мыши и клавиатуры. Выделяются следующие классы пользовательских команд по типу инициируемого действия:

- команды, которые инициируют вопрос в системе;
- команды, которые инициируют действия связанные с редактированием *sc*-текстов;
- команды, которые инициируют действия связанные с просмотром *sc*-текстов;
- команды, которые инициируют вывод *декомпозиции** или *разбиения** объекта.

По способу определения аргументов, выделяются следующие классы пользовательских команд:

- команды с заранее определенными аргументами (объектами, над которыми будет производиться инициируемое действие);
- команды с дополнительно указываемыми аргументами.

Как и вся другая информация, пользовательские команды представлены с помощью *SC*-кода в базе знаний. При визуализации с помощью *SCg*-кода, они представляются в виде элементов управления. Рекомендуется изображать элементы управления в виде прямоугольников, внутри которых

присутствует графический или текстовый идентификатор команды. Чтобы различать различные классы команд кроме идентификатора команды в прямоугольнике изображается графический идентификатор класса команды (пример на рисунке 4)

↓ Просмотр БЗ

Рисунок 4 – Пример изображения команды, которая инициирует вывод *декомпозиции** или *разбиения** объекта.

Инициирование пользовательской команды происходит при нажатии левой клавиши мыши на *sc.g*-узле, который её изображает. Все инициируемые пользователем команды попадают в базу знаний, где хранится их четкая последовательность. Это дает возможность их последующей обработки и анализа. Для анализа действий пользователя выделяется специализированная система – система управления диалогом. Именно в этой системе принимаются решения о том, каким образом строить диалог между пользователем и системой.

В современных пользовательских интерфейсах существующих приложений, как и в предлагаемом подходе, также есть команды, для инициирования которых необходимо указывать аргументы и команды, для которых аргументы уже заранее определены. В качестве примера команд с известными аргументами (объектами действий) можно привести команды закрытия окна (окно заранее известно), команды сворачивания окна и т. д. В качестве команд, для которых необходимо указать аргументы, можно привести следующие: указание фрагмента текста при форматировании в редакторе Word и т. д. Проблемой является то, что в различных системах указание аргументов происходит различным. Это в свою очередь требует от пользователя некоторого времени для освоения. В предлагаемой модели указание аргументов команд происходит явно (при этом пользователь всегда может узнать, какие аргументы необходимы для инициирования команды, задав соответствующий вопрос системе). Таким образом, процесс инициирования команды сводится к следующей последовательности действий:

- указание аргументов команды (в настоящий момент это делается левой клавишей мыши, с зажатой клавишей Alt);
- инициирование команды нажатием левой клавиши мыши (без зажатой клавиши Alt) на *sc.g*-узле изображающем её.

База знаний пользовательского интерфейса состоит из следующих разделов:

- Описание синтаксиса и семантики всех используемых внешних языков;
- Описание пользовательских команд;
- Описание принципов работы самого пользовательского интерфейса;
- Описание используемых компонентов – используются интерфейсом для организации

диалога с пользователем. К примеру, при установке содержимого sc-рамки пользователю отображается список редакторов, которые позволяют его редактировать;

- Протокол действий пользователя.

Машина обработки знаний интерфейса состоит из набора sc-операций. Кроме sc-операций, работающих только над sc-памятью, она содержит эффекторные и рецепторные операции. Эффекторные sc-операции, реагируя на изменения в sc-памяти, производят изменения во внешней среде (вывод информации). Рецепторные sc-операции, реагируя на изменения во внешней среде, производят изменения в sc-памяти (ввод информации).

Машина обработки знаний пользовательского интерфейса состоит из следующих типов sc-операций:

- Операции работы с окнами:
 - создание окна указанного типа;
 - удаление окна;
 - открытие/закрытие окна;
- Эффекторные sc-операции:
 - инициирование вывода ответа на вопрос заданный пользователем;
 - вывод информационной конструкции пользователю (на экран, в наушники и т. д.);
- Операции эмуляции элементарных действий пользователя:
 - Перемещение мыши;
 - Нажатие клавиш на клавиатуре;
 - Вращение колеса прокрутки.
- Рецепторные sc-операции (список может быть расширен в зависимости от используемых внешних устройств):
 - нажатие кнопки на клавиатуре;
 - нажатие клавиши мыши;
 - вращение колеса прокрутки мыши;
 - sc-операции трансляции информации из SC-кода во внешнее представление и обратно.

Описанные выше принципы построения пользовательского интерфейса позволяют проектировать пользовательские интерфейсы достаточно быстро и легко. При этом созданные таким образом пользовательские интерфейсы с точки зрения пользователя внешне мало чем отличаются от современных пользовательских интерфейсов:

- проектируемый пользовательский интерфейс является оконным, что присутствует во всех системах. Главное окно можно сравнить с рабочим столом операционной системы. В рамках этого окна создаются дочерние окна, в которых происходит просмотр или редактирование некоторых файлов;
- в рамках главного окна (рабочего стола) присутствуют некоторые команды, которые могут быть инициированы пользователем с указанием аргументов аргументами или же без них.

Однако предлагаемый подход имеет ряд

преимуществ. Так как в основе предлагаемых графических пользовательских интерфейсов лежит SCg-код, то у пользователя появляется возможность указывать и элементы управления в качестве аргументов команд. Это значительно снижает требования, предъявляемые к начальной квалификации пользователя. Ему лишь необходимо знать, как задать вопрос. Умея задавать вопросы, он может легко изучить команды и научиться работать с пользовательским интерфейсом (в современных системах для этого приходится читать документацию). Унификация процесса формирования пользовательских команд, также упрощает освоение пользовательского интерфейса конечным пользователем, так как при переходе от пользовательского интерфейса одной системы к интерфейсу другой ему не нужно заново учиться этому. Благодаря тому, что база знаний пользовательского интерфейса содержит описание команд и используется SCg-код, появляется возможность производить демонстрации работы с теми или иными командами. Пользователь может попросить систему показать, как пользоваться той или иной командой. При этом система, основываясь на описании команды, построит последовательность элементарных пользовательских действий, которая потом будет выполнена специализированными sc-операциями. Другими словами пользователю будет явно показано, как это сделать (система возьмет под свой контроль перемещение мыши и клавиатуру). Это также дает целый ряд преимуществ, при тестировании проектируемых интерфейсов. Так как для этого не надо записывать огромные протоколы с явным указанием элементарных действий (как это делается при тестировании современных интерфейсов). Достаточно лишь описать последовательность команд, остальное сделает сама система.

Полное описание пользовательского интерфейса с помощью SC-кода в базе знаний позволяет также решить еще одну очень важную задачу. Сейчас очень остро стоит проблема проектирования документации к разрабатываемым системам и их пользовательскому интерфейсу. По сути происходит следующее:

- проектировщик (дизайнер) описывает пользовательский интерфейс (создает техническое задание), в котором описываются все команды, внешние языки и т. д.;
- разработчик (программист) реализует описанные команды и принципы;
- технический писатель пишет по ним документацию.

На всех трех этапах одни и те же вещи описываются на различных языках. В предлагаемом подходе, описания команд, которые делаются разработчиком на формальном языке, при описании проектируемого интерфейса, сразу же и является частью справочной системы, которая в свою очередь позволяет задавать вопросы и получать ассоциативный доступ к хранимой информации (чего нет в современных справочных системах). Это

значительно сокращает сроки проектирования. Так как уже разработанный интерфейс и является большей частью справочной системы по его эксплуатации.

Разбиение пользовательского интерфейса на компоненты, которые взаимодействуют между собой через общую sc-память (базу знаний), используя общие ключевые узлы и набор sc-операций, сводит зависимость между ними к минимуму. Снижение зависимостей между проектируемыми компонентами значительно сокращает их срок разработки, и упрощает дальнейшую поддержку.

Благодаря тому, что все действия пользователя семантически интерпретируются и записываются в протокол, появляется возможность их анализа. Это делает возможным помощь пользователю при освоении пользовательского интерфейса (к примеру, система может подсказать пользователю, что некоторые действия можно делать гораздо проще). Также это дает возможность системе подстраивать пользовательский интерфейс под конкретного пользователя (показывать наиболее часто используемые команды, использовать более удобные принципы размещения элементов управления и т. д.)

3. Библиотека совместимых компонентов пользовательских интерфейсов

Большое разнообразие пользовательских интерфейсов влечет за собой разработку большого числа компонентов (под компонентом понимается часть пользовательского интерфейса, которая может быть интегрирована и использована в различных системах). Стоит отметить, что компоненты могут состоять из других компонентов (уровень вложенности не ограничен). Таким образом, в качестве компонентов, могут выступать уже спроектированные пользовательские интерфейсы. Большое число компонентов создает проблему их хранения и поиска. Чтобы решить эту проблему, в технологию включена библиотека совместимых компонентов пользовательских интерфейсов (далее библиотека компонентов). Она представляет собой специализированную интеллектуальную систему, которая решает следующие задачи: хранение, поиск, добавление, удаление (поддержка актуальности) и сравнение компонентов пользовательских интерфейсов.

Одним из важных типов компонентов, который включается в библиотеку, является типовое ядро пользовательских интерфейсов. Этот вид компонентов нужен для погружения спроектированного пользовательского интерфейса на определенную платформу. Такие компоненты включают в себя набор необходимых рефепторных и эффекторных sc-операций и обеспечивает взаимодействие компонентов пользовательских интерфейсов между собой.

В рамках библиотеки, все компоненты специфицируются и классифицируются. Их классификация производится по различным признакам. К примеру, в зависимости от решаемых задач компоненты делятся на следующие классы:

- просмотрщики содержимого sc-ссылок;
- редакторы содержимого sc-ссылок;
- трансляторы содержимого sc-ссылки в SC-код;
- трансляторы из SC-кода в содержимое sc-ссылок.

Технология позволяет интегрировать в качестве компонентов редакторы и просмотрщики, разработанные с использованием других технологий (далее их будем называть инородными компонентами пользовательского интерфейса). В основном они используются для просмотра и редактирования содержимого sc-ссылок. Это позволяет сэкономить время на разработке такого класса компонентов.

База знаний библиотеки компонентов включает в себя следующую информацию:

- спецификация компонентов;
- история версий компонентов;
- статистика использования компонентов.

Машина обработки знаний состоит из следующего набора операций:

- поиск компонента, по его спецификации;
- добавление нового компонента;
- удаление компонента;
- сравнение двух компонентов.

Пользовательский интерфейс библиотеки компонентов строится на основе SCg-интерфейса (комплекс информационно-программных средств обеспечивающих общение интеллектуальных систем с пользователями на основе SCg-кода, как способа внешнего представления информации). Однако, это не исключает возможность использования других способов диалога пользователя с библиотекой компонентов.

Очевидным является то, что такая система должна быть реализована в интернет варианте, так чтобы она постоянно пополнялась и все имели доступ к самой актуальной версии компонентов.

В рамках библиотеки компонентов могут содержаться различные версии и модификации какого-либо компонента. К примеру, компонент просмотра sc.g-конструкций может иметь двумерную, трехмерную или же многослойную модификации, при этом каждая из модификаций компонента может иметь различные версии.

Использование библиотеки компонентов при проектировании пользовательского интерфейса прикладной системы позволяет значительно сократить сроки проектирования, а также снизить требования, предъявляемые к начальной квалификации разработчика. Это достигается за счет проектирования пользовательского интерфейса из уже заранее заготовленных модулей, что также

позволяет повысить качество проектируемого интерфейса.

4. Интегрированные средства проектирования пользовательских интерфейсов интеллектуальных систем

Интегрированные средства проектирования пользовательских интерфейсов интеллектуальных систем решают задачу автоматизации и интеллектуализации процесса проектирования. В их состав входят инструментальные средства визуального проектирования, средства отладки и верификации проектируемых интерфейсов, библиотека компонентов пользовательских интерфейсов и help-система. Интеграция указанных систем в рамках интегрированной среды, позволяет значительно ускорить процесс проектирования, а также повысить качество проектируемых интерфейсов.

В основе инструментальных средств визуального проектирования лежит пользовательский SCg-интерфейс. Это дает разработчику возможность проектирования пользовательских интерфейсов с использованием известной технологии WYSIWYG (What You See Is What You Get – «что вы видите то и получите»), то есть результат проектирования выглядит так же, как и прототип во время разработки).

Инструментальные средства верификации и отладки, пользовательских интерфейсов, представляют собой специализированную интеллектуальную систему, которая позволяет находить и устранять ошибки в проектируемом пользовательском интерфейсе. Эта система направлена на автоматизацию и последующую интеллектуализацию процесса верификации и отладки проектируемого интерфейса. Для нее выделено три уровня интеллектуализации:

- на первом уровне средства верификации и отладки позволяют осуществлять поиск ошибок в проектируемом интерфейсе;
- на втором уровне интеллектуализации добавляется возможность анализа и оптимизации проектируемого интерфейса;
- на третьем уровне интеллектуализации появляется возможность автоматического исправления ошибок.

5. Методика проектирования пользовательских интерфейсов

Методика проектирования пользовательских интерфейсов является важной частью технологии, так как она описывает сам процесс проектирования.

На первом этапе разработчику необходимо специфицировать проектируемый пользовательский интерфейс. Спецификация включает в себя список задач решаемых интерфейсом, описание внешних языков представления знаний.

Следующим этапом является создание задачно-ориентированной декомпозиции проектируемого пользовательского интерфейса. На этом этапе он разбивается на интерфейсные подсистемы, которые могут разрабатываться параллельно. Это позволяет сократить сроки проектирования пользовательского интерфейса. Целесообразно проводить разбиение таким образом, чтобы максимальное количество подсистем уже имелось в библиотеке совместимых компонентов пользовательских интерфейсов.

После уточнения функциональных возможностей разрабатываемых подсистем производится их разработка с использованием семантической технологии проектирования интеллектуальных систем.

После разработки пользовательского интерфейса разработчик выделяет из него типовые фрагменты и специфицируя их необходимым образом включает в библиотеку компонентов, тем самым внося вклад в развитие технологии.

Таким образом, сама методика проектирования направлена на развитие технологии за счет разработки новых компонентов и пополнения ими библиотеки.

6. Интеллектуальная help-система по технологии

Интеллектуальная help-система, по семантической технологии проектирования пользовательских интерфейсов интеллектуальных систем, решает задачу помощи разработчикам в процессе проектирования пользовательских интерфейсов. Данная интеллектуальная система строится с использованием семантической технологии проектирования интеллектуальных систем. Её база знаний содержит следующие разделы:

- унифицированная семантическая модель пользовательских интерфейсов интеллектуальных систем. В данном разделе содержится полное формальное описание модели пользовательских интерфейсов, это дает возможность пользователю (разработчику) значительно быстрее усвоить теорию и понять принципы лежащие в основе проектируемых им интерфейсов;
- библиотека совместимых компонентов пользовательских интерфейсов интеллектуальных систем. В этом разделе приводится полное формальное описание библиотеки компонентов. Это позволяет разработчику значительно быстрее освоить принципы работы с библиотекой и тем самым сократить сроки проектирования;
- интегрированные средства проектирования пользовательских интерфейсов. Раздел базы знаний посвященный описанию средств, которые позволяют автоматизировать и интеллектуализировать процесс проектирования;

- методика проектирования пользовательских интерфейсов интеллектуальных систем. Содержит детальное описание процесса проектирования, что в свою очередь позволяет пользователю получить ответ на такой вопрос: *Что и как делать дальше?*;
- методика обучения проектированию пользовательских интерфейсов интеллектуальных систем. Данный раздел базы знаний полностью посвящен процессу обучения проектированию. Он содержит информацию, которая должна помочь пользователю в освоении самой технологии;
- help-система по семантической технологии проектирования пользовательских интерфейсов интеллектуальных систем. Раздел, который призван помочь пользователю освоить саму help-систему.

Включение интеллектуальной help-системы в состав технологии позволяет не только сократить сроки проектирования, но и снизить требования, предъявляемые к начальной квалификации разработчика.

ЗАКЛЮЧЕНИЕ

Описанные выше принципы и приемы позволяют проектировать интеллектуальные пользовательские интерфейсы для интеллектуальных систем, которые легко интегрируются в них и строятся на основе уже имеющихся компонентов. На основе предлагаемой технологии уже проектируются пользовательские интерфейсы некоторых прикладных систем. Что дает возможность говорить о работоспособности предлагаемого подхода. Все результаты, в том числе и исходные коды компонентов и ядра пользовательских интерфейсов, представлены на сайте проекта OSTIS.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Голенков и др, 2001] Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков, [и др]; – Мн. : БГУИР, 2001.
- [Грибова и др, 2005] Использование методов искусственного интеллекта для проектирования пользовательского интерфейса / В.В. Грибова, А.С. Клещев // Информационные технологии. №8. — 2005. — с.58-62
- [Грибова и др, 2011] Автоматизация разработки пользовательских интерфейсов с динамическими данными / В.В. Грибова, Н.Н. Черкезишвили // Материалы международной научно-технической конференции “Открытые семантические технологии проектирования интеллектуальных систем” - 2011. – с. 287-292
- [Касьянов, 2003] Касьянов, В.Н. Графы в программировании: обработка, визуализация и применение/ В. Н. Касьянов, В. А. Евстигнеев // СПб.: ВНУ, 2003. – 1104 с.
- [Курзанцева, 2008] Об адаптивном интеллектуальном интерфейсе “Пользователь – система массового применения” / Л.И. Курзанцева // Комп’ютерні засоби, мережі та системи №7, 2008. – с. 110 – 116
- [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.
- [Fei, 2001] Multi-Modal Human Interactions with an Intelligent Interface Utilizing Images, Sounds and Force Feedback / Fei He,

Arvin Agah // Journal of Intelligent and Robotic Systems. – Kluwer Academic Publisher, 2001.

[Myers B.A., 1992] Survey on User Interface Programming, Proceedings SIGCHI'92: Human Factors in Computing Systems / Myers B.A., Rosson M.B. - Monterey, CA, 1992. - P. 195-202.

[OSTIS, 2011] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2011. - Режим доступа: <http://ostis.net>. – Дата доступа: 10.11.2011

[Patrick, 2003] Intelligent user interfaces: introduction and survey / Patrick A.M. – Delft University of Technology, 2003

SEMANTIC MODELS OF MULTIMODAL USER INTERFACES AND SEMANTIC TECHNOLOGY FOR THEIR DESIGN

Koronchik D. N.*

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

denis.koronchik@gmail.com

This article describes semantic technology for user interface design. This article contains description of semantic user interface model, which is a main part of that technology.

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 that allow to create user interfaces.

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 design 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;
- design 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, allows to create intelligent user, that designed by using components.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.81:159.942.52

ЗАДАЧА СОЗДАНИЯ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО РАСПОЗНАВАНИЯ ЭМОЦИЙ

Заболеева-Зотова А.В., Орлова Ю.А., Розалиев В.Л., Бобков А.С.

*Волгоградский государственный технический университет,
г. Волгоград, Россия*

zabzot@vstu.ru

vladimir.rozaliiev@gmail.com

В статье рассматривается актуальное направление определения эмоциональных реакций человека по мимике, движениям, голосу. Приведено краткое описание современных систем распознавания эмоциональных реакций человека. Описаны подходы и решения авторов для построения системы автоматизированного определения эмоциональных реакций человека. Рассматриваются перспективные направления использования комплексной системы распознавания эмоций.

Ключевые слова: распознавание эмоциональных реакций, автоматизированные системы, перспективы использования.

ВВЕДЕНИЕ

Основное направление развития внедряемых в жизнь человека новых информационных технологий – это улучшение человеко-машинного взаимодействия. Создание роботов способных заменить в некоторых отраслях реальных людей невозможно без преодоления барьера взаимоотношений человек-машина. Неспособность машин испытывать и проявлять эмоции является одним из факторов препятствующих быстрому внедрению роботов в повседневную жизнь людей. Количество проводимых исследований и объемы финансирования, говорят о том, что данное направление является очень перспективным. [Заболеева-Зотова и др., 2011] Развитие телекоммуникационных технологий меняет межлическую коммуникацию и это в скором времени приведет к тому, что люди, все чаще, будут использовать виртуальное общение, которое станет легким в освоении и эффективным, однако при этом выражение эмоций будет теряться. А поскольку эмоции человека влияют на когнитивные процессы, и в том числе на процесс принятия решения о совершении каких-либо действий, то такие системы приобретают всё большее значение.

1. Существующие системы распознавания эмоциональных реакций

В последнее время активно развиваются веб-решения на основе модели SaaS (Software as a

Service – программное обеспечение как услуга) для определения эмоций пользователя по мимике. Компании Google, Twitter, FaceBook, Flickr и YouTube и многие другие проводят исследования и внедрения таких систем, так Google Inc. применяет технологии распознавания лиц и эмоций для более умного поиска, Twitter оценивает настроение пишущих в онлайн-блогах людей, FaceBook недавно представил обществу усовершенствование под названием Tagger, которая автоматически распознает, отмечает и подписывает лица друзей пользователя социальной сети.

Не отстают и компании, представляющие данные инновации в своих продуктах, таких как цифровые фото- и видекамеры. Таких компаний очень много, среди них Ricoh, Fujifilm, Canon, Nikon и другие.

Хотя некоторые компании утверждают, что их продукт чтения эмоций лица является наиболее правильным, важно отметить тот факт, что на сегодняшний день технологии есть, но они распознают только черты, присущие и наиболее подходящие данному выражению лица. Но далеко не всегда можно сказать, действительно такую эмоцию испытывает человек или же это просто натренированная улыбка. Если же добиваться результата такого, чтобы внешнее выражение соответствовало внутренним (реально переживаемым) эмоциям пользователя программы, то, важно убедиться, что участнику действительно удобно и комфортно выражать свои эмоции изнутри и проявлять их на своем лице, и что он этого хочет.

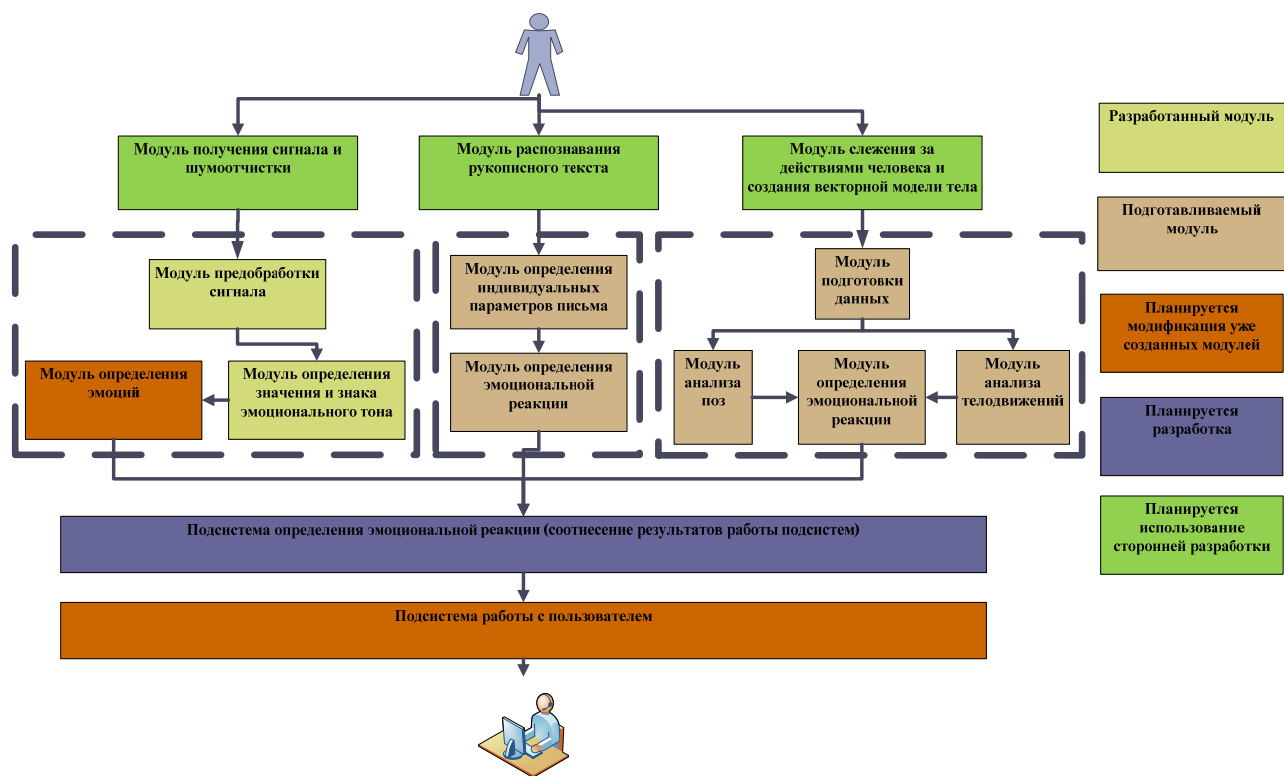


Рисунок 1 – Архитектура системы определения эмоциональных реакций

Это достигается с использованием полного комплекса технологий, каковыми являются распознавание выражения лица, распознавание речи и интонаций во время разговора данного человека, определение смысловой нагрузки и эмоциональных переживаний при письме и компьютерной переписке, распознавание телодвижений и походки.

В результате проведенного авторами обзора разработок компаний Ugobe, Machine Perception, NeuroSky, VibrImage, Sound Intelligence, TruMedia, FaceReader, Federal Express, лабораторий ERIC, Affective Computing Research, Массачусетского технологического университета (MIT), институт Фраунхофера, университеты Женева и Токио, Microsoft, Apple, Sony было определено, что существующие системы анализируют голос человека, его мимику, телодвижения, физиологические показатели, однако на сегодняшний момент не существует системы, полностью реализующей анализ всех средств передачи эмоциональных реакций человека.

2. Подход к решению проблемы

В своих работах мы предлагаем модели, методики и построенную на их основе систему, способную решить эту проблему. [Заболеева-Зотова и др., 2011; Розалиев, 2010] В состав такой системы входят уже разработанные блоки и подготавливаемые к разработке (рисунок 1). На вход системы подается видеосигнал, звуковой сигнал и образцы рукописного текста. На выходе оператору системы сообщается об эмоциональном состоянии исследуемых людей.

Нами были проанализированы теории эмоций, рассмотрены фундаментальные и современные работы ученых. [Анохин, 1980; Изард, 2003; Ильин, 2008] Различные теории рассматривают различные аспекты эмоций, это и физиологический, и эволюционный, и когнитивистский подходы. Безусловно, одна теория не может описать такие сложные явления как эмоции, поэтому важно рассмотрение всей области теорий эмоций. За теоретическую основу разработки автоматизированной системы определения эмоциональных реакций взята теория базовых эмоций, дополненная представлениями из теории физиологического и эволюционного происхождения эмоций, а так же теории оценок. [Ильин, 2008] Такая совокупность знаний сможет описать возникновение и протекание эмоций. Так эволюционная теория поможет выявить описания эмоций, свойственные всем людям, физиологическая поможет выявить изменения происходящие в организме человека и их внешние проявления, базисными эмоциями, возможно разделить все множество эмоций на несколько групп, а оценивание за счет прошлого опыта укажет на отличия эмоций от других состояний (например, от признаков болезни).

Мы принимаем, что эмоция является реакцией на какое-либо ощущение, а ощущение возникает при всяком событии. Таким образом, эмоция возникает как следствие всякого события. [Ильин, 2008] Тогда, процесс идентификации эмоциональной реакции по речи, переживаемой человеком, осуществляется в соответствии со следующей методикой: на первом этапе – определение знака эмоции (как функция от

параметров речевого сигнала) [Розалиев, 2010; Nozjan, 2003], на втором этапе – построение модели эмоции (в виде модели иммунного ответа организма) [Dasgupta, 1999], на третьем этапе – построение модели совокупности эмоций (на основе аппарата алгебры эмоций) [Фоминих, 2006].

Распознавание человека на видео изображении для определения переживаемой эмоциональной реакции осуществляется при помощи технологий безмаркерного захвата движений (требуемая информация собирается от сенсора Microsoft Kinect). После этого анализируется активность субъекта на видео, чтобы определить участки, где наблюдаются статические позы или динамические микро и макро движения. Формализации движений человека осуществляется посредством нечеткой логики. [Batyshin, 2008] Вводятся лингвистические переменные, которыми сможет оперировать не только разрабатываемая автоматизированная система, но и эксперт, описывающий характерные телодвижения на ограниченном естественном языке для поиска информации на видео изображении. Рассматриваются следующие параметры: часть тела, продолжительность движения, скорость движения и направление движения.

Подсистема поиска характерного эмоционального движения на видео изображении основана на интерпретирующей модели. Входные данные для алгоритма поиска – выражения на ограниченном естественном языке, а выходные – границы найденного движения, сегменты, на которые оно разбиты, а также степень достоверности полученных результатов. [Ковалев, 2011]

Позы в системе представлены переменными, описывающими положения частей тела относительно друг друга в определенный момент времени. Таким образом, мы имеем статические данные, которые необходимо классифицировать, используя аппарат нейронных сетей [Хайкин, 2006], в соответствие с заданным конечным числом поз или соответствующих им эмоциональных реакций, например гнев, радость, грусть и т.д.

3. Текущая работа

Одна из задач – поиск нечетких границ разбиения событий. [Заболеева-Зотова, 2011] Располагая границами событий, формами функций принадлежности для переменных и правилами перехода между ними, можно дефазифицировать значения, чтобы получить анимацию векторной модели человека. Таким образом, если правила построены и описывают паттерны, которые были получены при помощи захвата движений людей, а не сгенерированы в специально предназначенных пакетах, например Poser, 3ds Max, Motionbuilder или др., то в результате мы получим более

правдоподобные эмоциональные характерные жесты персонажей компьютерных игр и приложений.

Растущие объемы рынка индустрии компьютерных игр и приложений делают актуальной обратную задачу – по заданному текстовому описанию, построить анимацию векторной модели человека. В этом случае имеем на входе описание на ограниченном естественном языке, на выходе – анимированное движение в виде bvh файла.

Еще одна из задач, которая в настоящее время находится в процессе решения, состоит в том, чтобы из трех нечетких последовательных темпоральных высказываний, каждое из которых описывает движения вокруг одной из осей, построить одно высказывание, которое будет интегрально описывать все три, при этом семантика останется неизменной. [Заболеева-Зотова, 2011]

Так же, планируется развитие методов определения эмоциональных реакций по письменной речи, что существенно увеличит сферы использования разработок. Так контроль эмоциональности высказываний можно ввести на форумах, блогах и других местах обсуждения происходящих событий.

4. Трудности в реализации

Конечно же, есть и трудности. Изображения людей, передвигающихся в толпе, непросто изолировать, чтобы получить достаточно исходных данных для анализа, а при перемещении людей в направлениях, отличных от перпендикуляра к оптической оси камеры, полезная информация об их походке частично теряется. Однако можно использовать систему при пропуске людей через узкий проход поодиночке, к примеру, если поток людей движется через пост паспортного или таможенного контроля в аэропорту.

Но при этом возникает еще одна проблема – проблема понимания, что такое нормальное поведение. Количество возможных вариантов «нормального» поведения столь велико, что сложно провести грань между допустимым и недопустимым, и потому не исключается риск подвергнуть обыскам и допросам ни в чем неповинных людей. Если автоматика самостоятельно подает сигнал тревоги, окончательное решение должно оставаться за людьми. Это позволит избежать многих проблем, по крайней мере, на ранних этапах.

ЗАКЛЮЧЕНИЕ

Рассмотрим области, где применение системы определения эмоциональных реакций может быть полезно и необходимо. Такими областями являются: контроль общественных мест для противодействия противоправным и

экстремистским действиям; контроль поступающих звонков в службы доверия, экстренные службы (выявление излишне эмоциональных звонков); криминалистическая экспертиза – отсев фонограмм, содержащих излишне эмоциональную речь; отслеживание переживаний при снятии денег с банкомата, кассы; контроль состояния водителей; контроль сотрудников аэропортов, атомных станций и других важных объектов для недопущения к работе людей с неустойчивым эмоциональным состоянием; контроль действий сотрудников для предотвращения неврозов, депрессивных состояний; применение в играх, например в играх, разрабатываемых министерством обороны, для повышения реалистичности происходящих действий; замена полиграфа.

Контроль общественных мест может осуществляться на основе возможности получения информации от одного из модулей системы, так, если получение видео картинки сильно ограничено, на помощь должны прийти данные от звукозаписывающих устройств. Так, повышение громкости речи или наличие в речи определенного набора слов могут показать эмоциональные реакции говорящих и обратить внимание оператора на них. И наоборот, если получение звука ограничено из-за сильной зашумленности, то основу работы системы должны обеспечивать видеоканалы. Анализируя язык жестов, походку и телодвижения человека, система распознавания эмоциональных реакций может помочь в поимке преступников, а также быть полезной в борьбе с терроризмом, и применяться в медицинских целях.

Жизнь стремительно информатизируется и внедрение систем контроля эмоциональности не только по голосу, но и по видеоряду, передаваемому по сетям 3G или через Интернет (т.е. без существенных задержек и в хорошем качестве) позволит внедрять системы, способные контролировать жизнь общества, выявляя агрессию, злость, депрессивные состояния.

Работа выполнена при поддержке грантов РФФИ 10-01-00135-а, 10-01-00165-а, 10-01-90012-Бел_а, 11-01-97023-р_поволжье_а и Президента Российской Федерации №МК-3281.2011.9.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [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, 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.
- [Анохин, 1980] Анохин, П.К. Узловые вопросы теории функциональной системы / П.К. Анохин.-М.: Наука, 1980.– 290 с.
- [Изард, 2003] Изард К. Психология эмоций. СПб.: Питер, 2003. 464 с.

[Ильин, 2008] Ильин, Е.П. Эмоции и чувства. «Питер», СПб, 2008

[Hozjan, 2003] Hozjan V., Zdravko K. “Improved Emotion recognition with Large Set of Statistical Features”, Eurospeech 2003, 2003.

[Dasgupta, 1999] Artificial Immune Systems and Their Applications, D. Dasgupta (Ed.) Springer – Verlag. 1999

[Фоминих, 2006] Фоминих, И.Б. Эмоции как аппарат оценок поведения интеллектуальных систем. Десятая национальная конференция по искусственному интеллекту с международным участием КИИ-2006 [Электронный ресурс] / И.Б. Фоминих. - [2006]. - Режим доступа: <http://www.raai.org/resurs/papers/kii-2006/doklad/Fominykh.doc>

[Batyrrshin, 2008] Batyrrshin I.Z., Sheremetov L.B. Perception-based approach to time series data mining. - Applied Soft Computing 8, 3, 2008.

[Ковалев, 2011] Ковалев, С.М., Идентификация дискретно-динамической системы с изменяющейся структурой в стохастической среде / С.М. Ковалев, С.В. Соколов // Обозрение прикладной и промышленной математики. - Москва, 2011. - Т. 18, вып. 4. - С. 545-549.

[Хайкин, 2006] Хайкин, С. Нейронные сети: полный курс, 2-е издание. : Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1104 с.: ил.

[Заболеева-Зотова, 2011] Применение нечетких 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.

APPROACH TO THE PROBLEM OF CREATING AN AUTOMATED EMOTION-RECOGNITION SYSTEM

Zaboleeva-Zotova A.V., Orlova Y.A.,
Rozaliev V.L., Bobkov A.S.

*Volgograd State Technical University,
Volgograd, Russia*
zabzot@gmail.com

vladimir.rozaliev@gmail.com

The article deals with the actual direction - definition of emotional reactions on human facial expressions, movements and voice. Brief description of current systems for recognition human emotional responses given. Describes the authors' approaches and solutions to build the system automatically determine the emotional reactions. The emotional-recognition system may be useful and necessary in some areas: the control of public places to deal with illegal and extremist actions; control of incoming calls to the service trust, emergency services (identifying overly emotional connection); forensic examination - screening of phonograms containing overly emotional speech; tracking experiences when withdrawing money from ATMs, cash; condition monitoring of drivers; control of airport, nuclear power plants and other critical facilities to prevent the work of people with unstable emotional state; control actions of employees to prevent neuroses, depression; using in games, such as games developed by the Department of Defense, to improve the realism ongoing operations; replacement of the polygraph.



УДК 004.81:159.942.52

РЕШЕНИЕ ЗАДАЧИ ПРЕДСТАВЛЕНИЯ ДВИЖЕНИЙ В ВИДЕ НЕЧЕТКИХ ТЕМПОРАЛЬНЫХ ВЫСКАЗЫВАНИЙ ДЛЯ ОПРЕДЕЛЕНИЯ ЭМОЦИОНАЛЬНЫХ РЕАКЦИЙ ЧЕЛОВЕКА

Заболеева-Зотова А.В., Орлова Ю.А., Розалиев В.Л., Бобков А.С.

*Волгоградский государственный технический университет,
г. Волгоград, Россия*

zabzot@vstu.ru

vladimir.rozaliiev@gmail.com

В статье рассматриваются задачи автоматизации процесса распознавания эмоционального состояния человека. Подробно рассматриваются решения по движениям и позам. Приводится формализация активности движений, фазификация динамической информации, грануляции 1-го и 2-го уровней. Рассматривается модель представления характерных эмоциональных жестов и движений в виде нечетких последовательных темпоральных высказываний.

Ключевые слова: эмоции человека, автоматизация, характерные жесты, нечеткая темпоральная переменная, лингвистическое значение, фазификация, грануляция, нечеткое темпоральное событие, нечеткое последовательное темпоральное высказывание, нечеткий темпоральный гиперграф

ВВЕДЕНИЕ

Эмоции играют важнейшую роль в человеческой жизни. По тому, как выражаются и проявляются эмоции, можно многое сказать об отношении человека к различным объектам. Эмоции влияют на когнитивные процессы, и в том числе на процесс принятия решения о совершении каких-либо действий, поэтому системы определения эмоциональных реакций приобретают всё большее значение. [Заболеева-Зотова и др., 2011а]

Работа авторов статьи направлена на определение эмоциональных реакций людей по речи и телодвижениям. В статье подробно рассматривается подход, используемый в подсистеме определения эмоциональных реакций по движениям человека.

1. Формализация активности движений

1.1. Активность

Формализуем понятие активности.

$$A(\Delta t) = \sum_{n=1}^m (T_n(\Delta t) * k_n), \quad (1)$$

где m – количество временных рядов,

характеризующих движение, $T_n(\Delta t)$ – изменение n -го временного ряда за момент времени Δt , k – коэффициент чувствительности для n -го временного ряда (ВР).

Активность «А» выражается в количестве телодвижений человека: чем меньше телодвижений, и как следствие меньше изменений в каналах bvh файла, хранящего записанное движение, тем значение активности «А» – меньше. [Заболеева-Зотова и др., 2011b]

Активность телодвижений человека за момент времени Δt определяется формулой (1) и зависит непосредственно от того, какой частью тела субъект совершил движение.

1.2. Коэффициент чувствительности узла

Коэффициент чувствительности зависит от степени влияния n -го временного ряда на перемещение массы тела человека в пространстве.

Таким образом, опишем коэффициент чувствительности в следующем виде:

$$k_n = \sum_{i=0}^j (p_i * m_i), \quad (2)$$

где i – номер дочернего элемента, j – максимальный номер дочернего элемента, m_i –

коэффициент массы тела, показывает, какую долю от всей массы тела занимает данная часть, p_i - коэффициент пропорциональности массы тела.

Учёными биомеханиками были произведены исследования по определению массы и длины частей тела человека. В результате усреднения полученных результатов для взрослого мужчины получены данные, представленные на рисунке 1.



Рисунок 1 - Распределение массы человека

Так как, векторная модель имеет иерархическое описание, то это накладывает определённые особенности на расчёт активности. Приведём расчёт коэффициента чувствительности узла, который отвечает за плечо, при изменении угла вращения относительно оси X:

$$k_n = p_0 * m_0 + p_1 * m_1 + p_2 * m_2, \quad (3)$$

где k_n - коэффициент чувствительности канала, отвечающий за поворот в узле «плечо» относительно оси X; m_i - коэффициент массы определенной части тела; p_i - для среднего взрослого мужчины по всем элементам будет равен единице.

2. Фазификация и гранулирование

2.1. Лингвистические переменные

Для дальнейшей обработки информации о телодвижениях человека авторами была произведена фазификация. [Заболеева-Зотова и др., 2011b] В каждом `bvh`- файле есть параметр, который отвечает за его скорость воспроизведения и соответственно записи. Таким образом, время можно рассчитывать либо в секундах, либо в кадрах.

Введем нечеткую темпоральную переменную «Продолжительность». Множеству термов соответствуют следующие понятия: «нулевая»,

«очень короткая», «короткая», «умеренная», «длительная», «очень длительная». В дальнейшем количество лингвистических значений, которое может принимать данная переменная в системе может быть настроено. Это зависит от ошибки, которую мы в дальнейшем получим при конвертации из четкого значения в нечеткое и обратно.

График функций принадлежности к термам переменной «Продолжительность» представлены на рисунке 2:

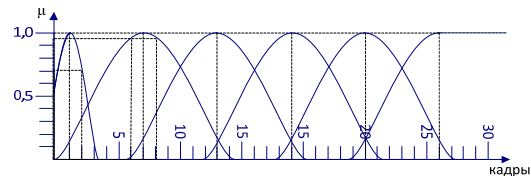


Рисунок 2 – График функций принадлежности к термам переменной «Продолжительность»

Для характеристики величины движения введем семейство лингвистических переменных (ЛП) «Нечеткое изменение угла».

2.2. Определение лингвистических значений переменных

Количество лингвистических значений (ЛЗ), которое будет принимать переменная, будет постоянное, а вот сами значения различные. Это зависит от максимальной подвижности (МП) и типа сустава (с тремя, с двумя, с одной осью вращения) или другой части тела, в которой происходит изменение угла поворота.

Векторная модель скелета состоит из 22 точек, чтобы определить ЛЗ, которые будут описывать эти точки необходимо найти соответствия между узлами и анатомическими местами изменения углов при движении человека. Требуемое соответствие представлено в статье [Заболеева-Зотова и др., 2011a]. В результате анализа полученного соответствия было выявлено, что соответствующими местами сгиба являются суставы и отделы позвоночника.

Для корректного составления функция принадлежности для ЛЗ, которые принимает переменная «Нечеткое изменение угла» были проанализированы величины нормальной подвижности, которые не считаются отклонениями, у людей, не занимающихся спортом. В результате анализа источников было выявлено, что в морфологии величины подвижности измеряются в терминах углов разгибания-сгибания, приведения-отведения, внутренней-наружной ротации. МП для суставов, определяем, в следующем виде: $МП_{сустава} = \text{Max}(\text{Сгиб} + \text{Разгиб}, \text{Прив} + \text{Отвед}, \text{Вн} + \text{Нар})$. Сводные результаты МП суставов в градусах представлены в статье [Заболеева-Зотова и др., 2011a]. Также было выявлено, что максимальная подвижность отделов позвоночника может быть рассчитана как сумма углов МП поворота налево и

направо вокруг продольной оси тела человека.

График функций принадлежности к термам переменной «Нечеткое изменение угла» представлены на рисунке:

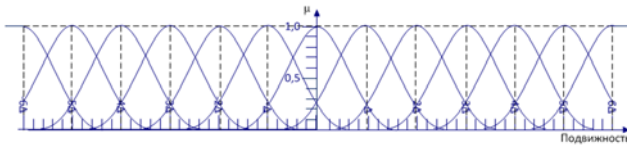


Рисунок 3 – График функций принадлежности к термам переменной «Нечеткое изменение угла»

Для каждого узла введем ЛЗ каждые Δ градусов. В зависимости от того, какого вида движения человека мы будем описывать, данное значение может настраиваться. Для описания малых периодических колебаний, например постукивание по столу пальцами руки или покачивание, «переминание» с ноги на ногу, в результате практических экспериментов было принято решения рассчитать Δ как $1/12$ от максимальной подвижности сустава. Данное значение позволило наиболее точно описать направление и скорость движение, не потеряв при этом семантики.

В качестве примера возьмём ЛЗ для узла, который соответствует локтевому суставу, максимальная подвижность которого равно 180 градусов. Для него $\Delta = \pi / 12$, таким образом, имеем следующие значения: «Около минус $\pi/2$ », ..., «Около минус $\pi/6$ », «около минус $\pi/12$ », «Около нуля», «Около $\pi/12$ », «Около $\pi/6$ », ..., «Около $\pi/2$ ».

После чего проводим гранулирование второго уровня, вводя ЛП «Скорость изменения угла». Функции принадлежности определены на множестве ЛЗ переменной «Нечеткое изменение угла». В результате практических исследований было выявлено, что малые периодические колебания не совершаются строго с определенной скоростью, происходит часто перескакивание скорости на соседние значения.

Множество термов ЛП состоит из следующих элементов: «Стабилизация», «Очень медленное увеличение», «Медленное увеличение», «Среднее увеличение», «Быстрое увеличение», «Очень медленное уменьшение», «Медленное уменьшение», «Среднее уменьшение», «Быстрое уменьшение», «Очень быстрое уменьшение».

Для более компактного описания в высказываниях было принято решение представить графики функций принадлежности к термам переменной «Скорость изменения угла» в следующем виде:

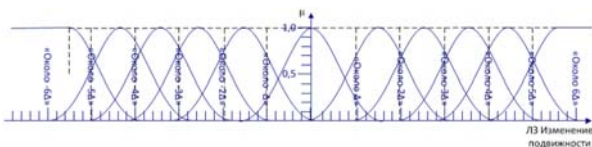


Рисунок 4 – График функций принадлежности к

термам переменной «Скорость изменения угла»

Таким образом, используя данные лингвистические переменные, можно описать движение сустава вокруг одной из осей в виде нечеткого темпорального события.

3. Нечеткие темпоральные высказывания

Так как события расположены на временной оси сразу друг за другом, то движение можно представить в виде нечеткого последовательного темпорального высказывания (НТВ). [Ковалев, 2011]

Например, на графике (рисунок 5) представлено положение угла поворота вокруг оси X для узла «правая нога» в bvh файле. Опишем отрезок $[t_4; t_{12}]$ в виде последовательного нечеткого темпорального высказывания: «Наблюдается среднее уменьшение угла очень короткой продолжительностью, за которым следует практически нулевая стабилизация изменения угла, за которой следует среднее увеличение угла очень короткой продолжительностью».

Моделью данного описания является следующее НТВ:

$$W = (B2rtfLa)rtsn(B0rtfLb)rtsn(B5rtfLa), \quad (4)$$

где ЛЗ нечеткой переменной «Тенденция ВР»: В2 - Среднее уменьшение, В0 – Стабилизация, В5 - Среднее увеличение; ЛЗ нечеткой темпоральной переменной «Продолжительность»: La - Очень короткая, Lb – Нулевая; rtsn – темпоральное отношение непосредственного следования.



Рисунок 5 – График изменения величины угла

Одна из задач – поиск нечетких границ разбиения событий. Решим её следующим способом. Примем маркером начала НТВ W , нечеткую точку $TNW = t_1$. Геометрической интерпретацией НТВ W является нечеткое разбиение

$$\sum_W TN = \{ [TN(a), TK(b)] [TN(b), TK(b)] [TN(c), TK(c)] \}, \quad (5)$$

Если значения функций принадлежности для ЛЗ

«очень короткая» и «нулевая» равны соответственно $\mu_{La}(6) = 0,92$, $\mu_{La}(7) = 1$, $\mu_{La}(8) = 0,92$, $\mu_{Lb}(0) = 0,7$, $\mu_{Lb}(1) = 1$, $\mu_{Lb}(2) = 0,7$, то подставляя их в рекуррентные формулы [Ковалев, 2011] найдём нечеткие границы разбиения.

Построим нечеткий темпоральный гиперграф для нашего высказывания.

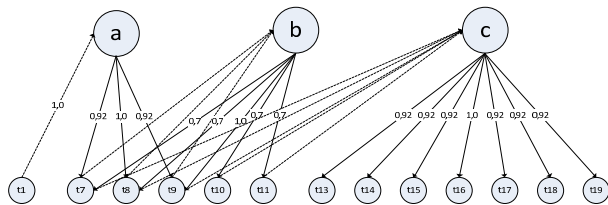


Рисунок 6 – Нечеткий темпоральный гиперграф

Располагая границами событий, формами функций принадлежности для переменных «Продолжительность», «Нечеткое изменение угла», «Тенденция ВР» и правилами перехода между ними, дефазифицируем значения чтобы получить анимацию векторной модели человека.

Таким образом, если правила построены и описывают паттерны, которые были получены при помощи захвата движений людей, а не сгенерированы в специально предназначенных пакетах, например Poser, 3ds Max, Motionbuilder или др., то в результате мы получим более правдоподобные эмоциональные характерные жесты персонажей компьютерных игр и приложений.

ЗАКЛЮЧЕНИЕ

На данном этапе совместно с психологами нами были разработаны высказывания на ограниченном естественном языке, которые описывают поведение человека, когда он испытывает недовольство или нетерпение. [Розалиев, 2010] Таким образом, система, имея набор высказываний, которые получены при помощи данного метода, и их интерпретацию, анализирует движения человека и строит описание на ограниченном естественном языке. Затем методом скользящего окна обнаруживает характерные паттерны и сигнализирует об этом пользователю автоматизированной системы.

Также, обращая внимание на всевозрастающие объёмы рынка индустрии компьютерных игр и приложений, можно говорить об актуальности еще одной задачи - по заданному текстовому описанию, построить анимацию векторной модели человека. В этом случае имеем на входе описание на ограниченном естественном языке, на выходе – анимированное движение в виде bvh файла.

Для автоматизации процесса распознавания эмоционального состояния человека по движениям и позам разработаны: классификация характерных поз, и их соответствие эмоциональным реакциям; база данных, хранящая интерпретацию, словесное

описание и изображение характерных поз; нейросетевая модель идентификации позы человека и эмоциональной реакции. Автоматизирован процесс предобработки данных и формализована активность движений человека. Произведена фазсификация динамической информации, а также грануляции 1-го и 2-го уровней. Разработана модель представления характерных жестов и телодвижений в виде нечеткого последовательного темпорального высказывания. Построены правила соответствия полученных описаний интерпретациям психологов.

Работа выполнена при поддержке грантов РФФИ 10-01-00135-а, 10-01-00165-а, 10-01-90012-Бел_а, 11-01-97023-р_поволжье_а и Президента Российской Федерации №МК-3281.2011.9.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Заболеева-Зотова и др., 2011а] 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.

[Заболеева-Зотова и др., 2011б] 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.

[Ковалев, 2011] Ковалев, С.М., Идентификация дискретно-динамической системы с изменяющейся структурой в стохастической среде / С.М. Ковалев, С.В. Соколов // Обозрение прикладной и промышленной математики. - Москва, 2011. - Т. 18, вып. 4. - С. 545-549.

[Розалиев, 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.

SOLVING THE PROBLEM OF FORMALIZING HUMAN MOVEMENT AS A FUZZY TEMPORAL EXPRESSIONS IN ORDER TO DETERMINE THE EMOTIONAL REACTIONS OF THE PERSON

Zaboleeva-Zotova A.V., Orlova Y.A., Rozaliev V.L., Bobkov A.S.

*Volgograd State Technical University,
Volgograd, Russia*

zabzot@gmail.com

vladimir.rozaliev@gmail.com

The article deals with the problem automation the process of recognition emotional state Detail the solutions found by the movements and postures. We present a formalization of the activity of movements, fuzzification dynamic information, granulation of the 1st and 2nd levels. Our model shows the characteristic gestures and movements in the form of fuzzy sequential temporal expression.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.934.2

СИСТЕМА ОПРЕДЕЛЕНИЯ ЭМОЦИОНАЛЬНОГО СОСТОЯНИЯ ДИКТОРА ПО ГОЛОСУ

Киселёв В.В., Давыдов А.Г., Ткачя А.В.

ООО «Речевые технологии», г. Минск, Республика Беларусь

info@spetech.by

В статье рассматривается анализ речевой коммуникации в интеллектуальных диалоговых системах телекоммуникационной сферы. Речевая аналитика – новое направление в области речевых технологий ориентированное, на автоматический анализ разговора с целью выявления удовлетворенности собеседника. В статье кратко представлены теоретические аспекты системы определения эмоционального состояния диктора по голосу и практическая реализация предложенных методов на примере программного комплекса.

Ключевые слова: голосовой анализ, машинное обучение, распознавание эмоций, эмоциональное состояние диктора.

ВВЕДЕНИЕ

Первые попытки определения эмоциональных состояний с использованием статистических закономерностей, присущих определенным акустическим особенностям голоса, были предприняты в середине 80-х годов [Van Bezooijen, 1984] и [Tolkmitt and Scherer, 1986]. В дальнейшем интерес к задаче автоматического распознавания эмоций по голосу только возрастал. Количество ежегодно публикуемых статей, относящихся к данной тематике, с середины 90-х годов до настоящего момента, возросло более чем в 10 раз [Schuller et al., 2011], а эффективность распознавания приблизилась к человеческой.

Однако, несмотря на достигнутые успехи в сфере голосового детектирования эмоциональных состояний, проблема еще очень далека от своего окончательного решения. Существует ряд серьезных препятствий, значительно усложняющих разработку эффективных систем распознавания эмоционального состояния.

Прежде всего, нет четкого определения понятия «эмоция». При любых попытках его формализации мы натываемся на многообразие психологических моделей эмоциональных процессов, демонстрирующих различные ракурсы восприятия и модели описания эмоционального состояния диктора.

Вторая сложность – отсутствие единой теории, связывающей внутренние состояния диктора с особенностями его речи. Несмотря на достигнутые

в этой сфере успехи, общепринятого подхода пока не существует [Scherer, 2003].

1. Система определения эмоционального состояния

Задача определения эмоционального состояния диктора широко востребована в современном мире, особенно в сфере оказания справочных услуг абонентам контакт-центров, для которых уровень удовлетворенности клиента оказанной ему услугой является главным критерием высокого качества работы.

Однако, в связи с большим количеством обрабатываемых запросов, невозможно производить ручную оценку удовлетворенности клиентов, отслеживать негативные звонки, «тихие» рекламации и пр. В этом случае для решения задачи определения эмоционального состояния диктора прибегают к использованию интеллектуальных автоматических систем распознавания эмоций.

В последние годы использование наукоемких алгоритмов голосового анализа, и в частности развитие методов распознавания эмоций, позволили получать оценку эмоций в реальном времени. Это позволяет использовать разрабатываемые системы не только для получения конечной оценки удовлетворенности абонента контакт-центра, но и для отслеживания изменения эмоционального состояния клиента во время его звонка.

Использование таких систем дает возможность четко контролировать работу операторов контакт-центров, с одной стороны своевременно сигнализируя им об малейших изменениях в эмоциональном состоянии абонента, позволяя

оператору быстро скорректировать свои дальнейшие действия, тем самым, предотвратив дальнейшее развитие конфликтной ситуации. А с другой стороны запись и анализ полученных эмоций могут быть использованы для выявления недобросовестности оператора контакт-центра.

Так же система определения эмоционального состояния может быть использованы в обучающих целях манеры ведения телефонных разговорах.

2. Определение информативных признаков

Важнейшим звеном системы автоматического детектирования эмоций по голосу диктора является выделение оптимального набора информативных признаков, коррелированных с эмоциональными состояниями. Выбор информативных признаков оказывает значительное влияние на эффективность классификации.

Акустические характеристики голоса могут быть условно разделены на пять категорий [El Ayadi et al., 2011]: просодические (частота основного тона, темп речи и т.д.), динамические (фонетическая функция), фонационные (отношение гармоник основного тона к шуму, джиттер, шиммер, и др.), спектральные (линейные спектральные частоты, кепстральные коэффициенты линейной шкалы частот, кепстральные коэффициенты мел-шкалы частот, и др.) и энергетические (отношение мощностей в спектральных полосах, оценка мощности сигнала и другие, как правило, основанные на энергетическом операторе Тигера). Каждая группа показателей предназначена для описания отдельных аспектов голоса, и находит свое применение в распознавании эмоциональных состояний.

Большинство исследователей уверены, что просодические характеристики речи, такие, как частота основного тона и энергия, передают значительную часть эмоционального содержания высказывания [Cowie et al., 2001], [Busso et al., 2009] и [Bosch, 2003]. Вильямс и Стивенс показали [Williams and Stevens, 1981], что степень возбуждения диктора влияет на суммарную энергию, распределение энергии по частотному спектру и на частоту и длительность пауз в речевом сигнале.

В качестве просодических характеристик можно выделить параметры, связанные с основным тоном, формантами, энергией, длительностями и артикуляцией [Cowie et al., 2001], [Murray and Arnott, 1993] и [Lee and Narayanan, 2005].

Значительные успехи в распознавании эмоций были достигнуты за счет генерации статичных векторов информативных признаков, получаемых из просодических характеристик сигнала после применения набора статистических функционалов. Предполагается, что данный факт косвенно свидетельствует о суперсегментной природе эмоциональной речи.

Джонстон и Шерер [Johnstone and Scherer, 2000]

проанализировали результаты большого числа исследований, проведенных в этой области, и представили результаты в форме таблицы:

Таблица 1 – Акустические паттерны базовых эмоциональных состояний

	Стресс	Гнев
Интенсивность	↑*	↑
F0 уровень/среднее	↑	↑
F0 вариабельность		↑
F0 диапазон		↑
Контурсы выражения		↓
Энергия ВЧ компонент		↑
Скорость речи и артикуляции		↑

продолжение Таблицы 1

Страх	Печаль	Радость	Скука
↑	↓*	↑	
↑	↓	↑	
	↓	↑	↓
↑(↓)	↓	↑	↓
	↓		
↑	↓	(↑)	
↑	↓	(↑)	↓

где символ «↑» обозначает увеличение параметра, символ «↓» – уменьшение.

3. Выбор наиболее информативных и помехоустойчивых параметров речи

Число всевозможных информативных признаков, выделяемых из звукового сигнала, может достигать нескольких тысяч. Далеко не все из них эффективны для решения задач распознавания эмоционального состояния, а немалая часть потенциально полезных характеристик оказывается избыточными. Перед построением и обучением классификаторов проводится предварительная процедура отбора информативных признаков. Конечной целью данного этапа работы является выделение релевантного набора характеристик звукового сигнала и декорреляция пространства информативных признаков.

При разработке современных систем распознавания эмоционального состояния диктора по голосу для минимизации набора информативных признаков широко применяются метод главных компонент и метод выделения признаков (feature selection). Их использование позволяет значительно сократить объем работы эксперта для построения наиболее эффективной системы. Вероятно, на данный момент наиболее популярной является стратегия линейного последовательного поиска (*sequential forward search*) [Pudil et al., 1994].

4. Алгоритмы классификации эмоционального состояния диктора

Системы автоматического распознавания эмоций состоят из двух основных блоков – первый осуществляет акустическую обработку входного речевого сигнала, выделяя из него набор

информативных признаков, отобранных в ходе предварительно проведенного исследования, а второй содержит классификатор, распознающий на их основе эмоциональное состояние диктора. Практически во всех предложенных системах распознавания эмоций были задействованы традиционные классификаторы, однако многие современные исследователи фокусируют свое внимание на разработке новых подходов.

Наиболее популярными техниками классификации являются следующие [Pantic and Rothkrantz, 2003]: поиск ближайших соседей (kNN), модель скрытых марковских процессов (HMM), модель смеси нормальных распределений (GMM), модели на основе нечеткой логики, искусственных нейронных сетей, байесовские классификаторы максимума вероятности, метод опорных векторов (SVM). На данный момент отсутствует какое-либо общепризнанное мнение по поводу того, какой классификатор лучше использовать.

При условии предварительно проведенного отбора информативных признаков, высокую эффективность дают такие классификаторы как линейный дискриминантный классификатор (LDC) либо поиск ближайших соседей (kNN), показавших свою эффективность как на идеальных [Dellaert et al., 1996] и [Petrushin, 1999], так и на более реалистичных базах [Batliner et al., 2000], [Kwon et al., 2003], [Lee and Narayanan, 2005] и [Shami and Verhelst, 2007].

5. Система автоматического определения эмоционального состояния

На основе вышеизложенной теории для разработки системы автоматического распознавания эмоций было принято решение использовать просодические характеристики речи в качестве информационных признаков эмоций, а классификацию эмоций проводить на основе поиска ближайших соседей (kNN).

Выбранная теория была реализована в программном комплексе, предназначенном для определения эмоционального состояния диктора по голосу.

В качестве просодических признаков эмоций были выбраны следующие характеристики голоса:

- интонированность: характеризует изменение производной частоты основного тона (ЧОТ). ЧОТ отражает высоту голоса диктора. Изменения (производная) ЧОТ определяют интонации голоса. Для монотонной речи характерны малые абсолютные значения производной ЧОТ (группировка значений около нуля) рисунок 1а. Излишне интонированная речь характеризуется значительным разбросом производной ЧОТ рисунок 1б;

- громкость – мощность сигнала;
- ритмичность – параметр, который отражает среднюю длительность фраз диктора;

- мелодичность – параметр, отражающий долю голосовых (вокализованных) фрагментов в речи;
- скорость – параметр, характеризующий темп речи, количество произносимых звуков в единицу времени.

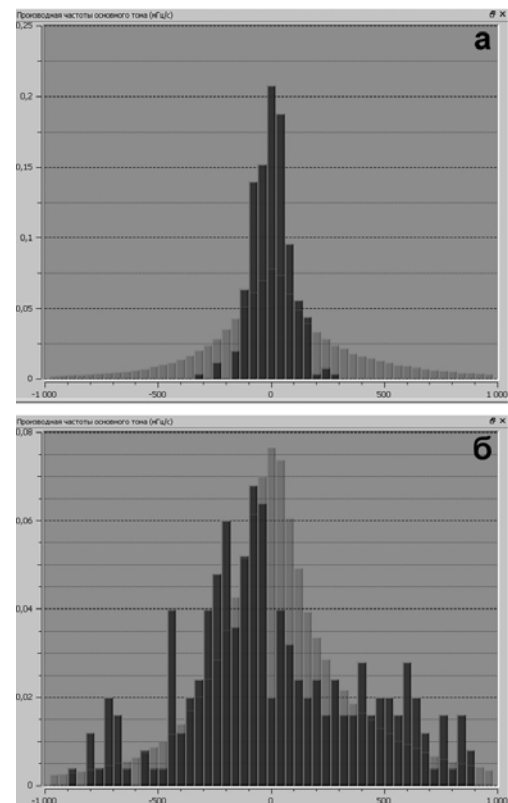


Рисунок 1 – Гистограмма интонированности речи (а – монотонная, б – излишне интонированная речь).

Гистограмма для достаточно выразительной речи показана зеленым цветом (более светлый оттенок)

Детектирование эмоций осуществляется по входу каждого из значений моментальных просодических характеристик голоса в соответствующие им диапазоны, заданные для каждой из эмоций.

В окне настройки просодических параметров эмоции также присутствуют параметры минимальной продолжительности эмоции и относительный коэффициент эмоции, который характеризует степень удовлетворенности (от 0 – полное отсутствие положительных эмоций, до 100 – максимально комфортное психоэмоциональное состояние). Оба этих параметра также учитываются при определении эмоции и вычислении общей оценки удовлетворенности диктора.

На рисунке 2, представлен фрагмент речи с распознанными эмоциональными состояниями.

Для увеличения достоверности принятия решения в программе используются модули определения пола и возраста диктора. Полученные параметры значений пола и возраста учитываются при оценке удовлетворенности диктора, так как

изменение этих параметров приводит с дисперсии величин просодических характеристик голоса.

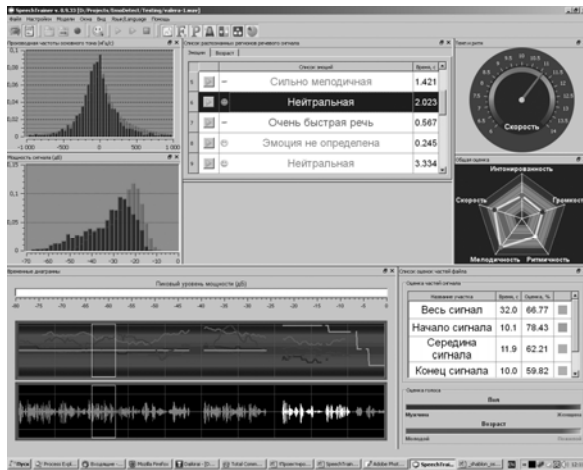


Рисунок 2 – Интерфейс программного комплекса

На ряду с определением эмоционального состояния диктора, программный комплекс позволяет просматривать текущие значения просодических характеристик голоса, создавать и настраивать просодические модели эмоций, а так же изменять параметры определения самих информативных просодических признаков. В связи с этим на базе программного комплекса можно проводить глубокий анализ просодических характеристик голоса.

ЗАКЛЮЧЕНИЕ

Речевые технологии предлагают пользователям широкий спектр автоматизированных услуг, одной из которых является автоматическая оценка эмоционального состояния диктора.

На основе изложенной в статье теории был разработан программный комплекс, который предназначен для повышения качества работы контакт-центра с клиентами при обслуживании обращений абонентов, нуждающихся в получении справочной информации. Преимущество использования разработанного ПО заключается в том, что при непрерывном притоке клиентов он обеспечивает постоянный контроль уровня качества обслуживания на любом этапе общения оператора с абонентом. Это позволяет центру обработки вызовов значительно повысить основные требования, предъявляемые к нему со стороны клиентов: скорость реакции на запросы абонентов, уровень удовлетворенности клиента и стремление к установлению эмоциональной связи с каждым абонентом.

Библиографический список

- [Van Bezooijen, 1984] The Characteristics and Recognizability of Vocal Expression of Emotions / Van Bezooijen, Dordrecht // The Netherlands: Foris 1984.
- [Tolkmitt and Scherer, 1986] Effect of experimentally induced stress on vocal parameters / Tolkmitt, F. J., Scherer, K. R. // J. Experimental Psychology: Human Perception and Performance 12(3), P. 302–313.
- [Schuller et al., 2011] Schuller, B., Batliner, A., Steidl, S. and Seppi, D. Recognising realistic emotions and affect in speech // State

of the art and lessons learnt from the first challenge. Speech Communication, In Press.

[Scherer, 2003] Vocal communication of emotion / Scherer, K.R. // A review of research paradigms. Speech Communication, 40(1-2), P. 227-256.

[El Ayadi et al., 2011] Survey on speech emotion recognition: Features, classification schemes, and databases / El Ayadi, M., Kamel, M.S. and Karray, F. // Pattern Recognition, 44(3), P. 572-587.

[Cowie et al., 2001] The description of naturally occurring emotional speech / Douglas-Cowie, E., Cowie, R. and Schroder, M. // Proc. 15th Internat. Conf. on Phonetic Sciences, Barcelona, Spain, P. 2877–2880.

[Busso et al., 2009] Analysis of emotionally salient aspects of fundamental frequency for emotion detection / Busso, C., Lee, S., Narayanan, S. // IEEE Trans. Audio Speech Language Process. 17(4), P. 582–596.

[Bosch, 2003] Emotions, speech and the asr framework / Bosch, L.; Speech Commun. 40, P. 213–225.

[Williams and Stevens, 1981] Vocal correlates of emotional states / Williams, C. and Stevens, K. // In: J. Darby (Ed.), Speech evaluation in psychiatry. New York: Grune & Stratton., P. 189–220.

[Murray and Arnott, 1993] Toward a simulation of emotions in synthetic speech: A review of the literature on human vocal emotion / Murray, I., Arnott, J. // J. Acoust. Soc. Am. 93(2), P. 1097–1108.

[Lee and Narayanan, 2005] Analysis of emotionally salient aspects of fundamental frequency for emotion detection / Busso, C., Lee, S., Narayanan, S. // IEEE Trans. Audio Speech Language Process. 17(4), P. 582–596.

[Johnstone and Scherer, 2000] Vocal communication of emotion / Johnstone, T., Scherer, K.R. // In: Lewis, M., Haviland, J. (Eds.), Handbook of emotion, second ed. Guilford, New York, P. 220–235.

[Pantic and Rothkrantz, 2003] Toward an Affect-Sensitive Multimodal Human-Computer Interaction / Pantic, M. and Rothkrantz, L.J.M., // Proceedings of the IEEE, 91(9), P. 1370-1390.

[Dellaert et al., 1996] Recognizing emotion in speech / Dellaert, F., Polzin, T., Waibel, A. // In: Proc. ICSLP, Philadelphia, PA, USA, P. 1970–1973.

[Petrushin, 1999] Emotion in speech: recognition and application to call centers / Petrushin, V. // In: Proc. Artificial Neural Networks in Engineering (ANNIE'99), St. Louis, MO, USA, P. 7–10.

[Batliner et al., 2000] Desperately seeking emotions: actors, wizards and human beings / Batliner, A., Fischer, K., Huber, R., Spiker, J. and Noth, E. // In: Proceedings of the ISCA Workshop Speech Emotion, P. 195–200.

[Kwon et al., 2003] Emotion recognition by speech signals / Kwon, O.-W., Chan, K., Hao, J., Lee, T.-W. // In: Proc. Interspeech, P. 125–128.

[Shami and Verhelst, 2007] An evaluation of the robustness of existing supervised machine learning approaches to the classification of emotions in speech / Shami M. and Verhelst W. // Speech communication, 49(3), P. 201-223.

[Pudil et al., 1994] Pudil, P., Novovicova, J., Kittler, J. (1994) Floating search methods in feature selection. Pattern Recognition Lett. 15, pp. 1119–1125.

SYSTEM OF DETERMINATION OF THE EMOTIONAL STATE OF THE SPEAKER OF A SOFTWARE TO THE VOICE

Kyselov V.V., Davydov A.G., Tkachenja A.V.

LLC "Speech Technology", Minsk, Belarus
info@speetech.by

The article describes the analysis of speech communication in intellectual dialogue systems of telecommunication. Speech analytics is a new direction in the field of speech technologies which is focused on the automatic analysis of conversation aimed at detecting the satisfaction of the interlocutor. The article briefly describes the theoretical aspects of the emotion recognition system and practical realization of the suggested methods through the example of bundled software.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.896

ОБРАБОТКА КОНТЕКСТНЫХ ДАННЫХ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

А.Г.Киселёва, Г.Д.Киселёв

*Национальный технический университет Украины «Киевский политехнический институт»
г. Киев, Украина*

kiseleva_anna@ukr.net

kiselev_gd@bigmir.net

В статье описана библиотека программ позволяющая повысить достоверности исходных данных в контекстных приложениях. Приводится анализ применения различных методов фильтрации и сглаживания временных рядов. Оценивалась эффективность методов экстраполяции и линейного предсказания.

Ключевые слова: контекстно-зависимые приложения, контекст, сглаживание, предсказание.

ВВЕДЕНИЕ

Контекстно-зависимыми приложениями (КЗП) называются интеллектуальные системы, в которых обеспечивается эффективный интерфейс с пользователем и с другими программными системами за счет использования контекстных данных, получаемых от разных информационных источников. Под контекстными данными или контекстом понимают информационные потоки (временные ряды), идущие от датчиков (сенсоров), сканирующих окружающее пользователя пространство (оборудование, программное обеспечение, физическое пространство и социальная среда). На физическом уровне входные и выходные информационные потоки переносятся электрическими сигналами.

Сложность решения задач обработки физических сигналов поступающих от различных датчиков обусловлена тем, что для регистрации информационных потоков поступающих от пользователя и снимаемых с датчиков, помимо полезного сигнала, фиксируются электрические и магнитные сигналы от посторонних источников. Такие помехи, неизбежно возникающие в реальных условиях, рассматриваются как аддитивный шумовой компонент, искажающий полезный сигнал. В общем случае, создатели КЗП не имеют данных о характеристиках шумовой компоненты, которые могут изменяться в зависимости от типа

сенсоров и условий эксплуатации системы. В связи с этим, актуальной является задача использования средств выделения полезного сигнала на фоне помехи, используя различные алгоритмы фильтрации и сглаживания сигналов, которые в минимальной степени искажают форму информативных фрагментов физических сигналов. В настоящей статье исследуется один из возможных подходов к решению задачи выбора эффективных алгоритмов фильтрации и сглаживания, а так же прогнозирования входных сигналов в распределенных интеллектуальных системах, в которых основным потребителем и источником контекстной информации является современные устройства, например, многофункциональный мобильный телефон. Для таких систем на алгоритмы подавления шумов и прогнозирования накладываются требования по минимизации потребляемых вычислительных ресурсов и времени вычислений.

Оцифрованные входные сигналы от контекстных источников входных сигналов, в частности, от датчиков мобильного телефона, описываются в виде временных рядов, членами которых являются вещественные числа. Поэтому, используемые алгоритмы прогнозирования, сглаживания и фильтрации сигналов должны эффективно работать с временными рядами. Краткосрочный прогноз значений членов временного ряда используется не только для плавного изменения выходных характеристик устройств, но и для

контроля достоверности контекстообразующих временных рядов.

Предмет исследования

Авторами статьи были разработаны и протестированы программы фильтрации и сглаживания временных рядов для следующих известных методов:

1. Метод простого скользящего среднего (ПСС) (SMA — SimpleMovingAverage);
2. Метод двойного скользящего среднего (ДСС) (DMA — DoubleMovingAverage)
3. Метод экспоненциального сглаживания (ЭС) (ES — ExponentialSmoothing);
4. Метод двойного экспоненциального сглаживания (ДЭС) (DES - DoubleExponentialSmoothing);
5. Фильтра Кальмана(КФ)(FilterKalmana - FK)

Методы скользящего среднего и экспоненциального сглаживания строятся на предположении, что при определении средних значений в некотором числовом ряду случайные отклонения погашаются. При сглаживании этими методами фактические значения членов временного ряда заменяются средними значениями, которые определяют срединную точку периода скользящего [2].

При реализации метода простого скользящего среднего (ПСС) используется формула:

$$s_t = \frac{1}{k} \sum_{n=0}^{k-1} x_{t-n} = s_{t-1} + \frac{x_t - x_{t-k}}{k}, \quad \text{где } k -$$

порядок скользящего среднего или количество предыдущих значений временного ряда, для которых вычисляется среднее значение; x_t – предыдущее зашумленное значение временного ряда; s_t – сглаженное значение очередного элемента временного ряда.

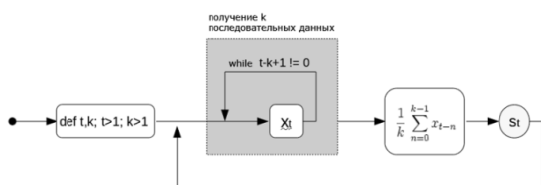


Рисунок1 –функциональная схема алгоритма вычисления простого скользящего среднего

В методе двойного скользящего среднего (ДСС) сглаженное значение элементов временного ряда вычисляется в два этапа. На первом этапе рассчитывается первичное сглаженное значение очередного элемента временного ряда. На втором этапе это значение уточняется, при этом, каждое зашумленное значение суммируется с рассчитанным на первом этапе сглаженным значением.(рис. 2).

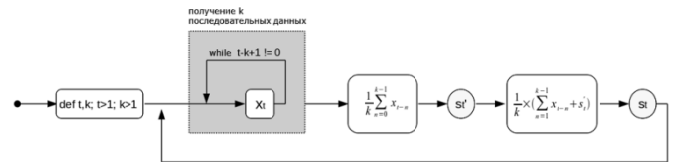


Рисунок2 – функциональная схема алгоритма вычисления двойного скользящего среднего

Скользящее среднее не даст точного сглаживания, если временные ряды монотонно возрастают или убывают. Этот метод лучше подойдет для ряда с небольшими случайными отклонениями данных от некоторого постоянного или медленно изменяющегося значения.

В методе простого экспоненциального сглаживания (ПЭС) весовые коэффициенты предыдущих наблюдаемых значений временного ряда увеличиваются по мере приближения к текущему (по времени) элементу в соответствии с формулой[3]:

$$s_t = \alpha x_t + (1 - \alpha) s_{t-1}, \quad \text{где } \alpha - \text{ параметр сглаживания.}$$

Выбор параметра сглаживания α представляет собой достаточно сложную проблему. Чем ближе параметр сглаживания к единице, тем больше влияние последних наблюдений и тем больше скорость убывания весов предыдущих наблюдений. Однако, если высокочастотная (шумовая) компонента ряда имеет достаточно большую дисперсию, не следует использовать большие значения параметра сглаживания из-за ухудшения качества прогноза.

Метод экспоненциального сглаживания не дает удовлетворительных результатов, если значения временного ряда монотонно возрастают или убывают. В таких случаях может быть применен метод экспоненциального сглаживания с учетом тренда. В частности, модель Хольта [4], в которой используется два уравнения – первое для сглаживания элементов ряда и второе для сглаживание тренда:

$$s_t = \alpha \gamma_t + (1 - \alpha)(s_{t-1} + b_{t-1}) \quad 0 \leq \alpha \leq 1, \\ b_t = \gamma(s_t - s_{t-1}) + (1 - \gamma)b_{t-1} \quad 0 \leq \gamma \leq 1,$$

где S_t — сглаженное значение прогнозируемого элемента для времени t ; b_t - оценка прироста тренда, показывающая возможное возрастание или убывание значений за одно приращение времени; α, γ - параметры сглаживания ($0 \leq \alpha \leq 1; 0 \leq \gamma \leq 1$); k - количество периодов времени, на которые на которые производится прогноз.

Параметры сглаживания α и γ выбираются. Чем больше значения параметров, тем большему сглаживанию подвергаются данные. При $\alpha=\gamma$ имеет место особый случай, поскольку в одинаковой мере производится сглаживание текущего значения и тренда. Такой вариант называется двойным экспоненциальным сглаживанием (ДЭС) Брауна [5].

Фильтр Кальмана (ФК), для расчета сглаженного значения очередного элемента

временного ряда, использует его зашумленное значение, которое пересчитывается с использованием уточняемого для каждого последующего значения временного ряда параметра ошибки [6]. Достоинства ФК:

- имеет рекуррентную формулу;
- легко обобщается для многомерных и нестационарных временных рядов;
- легко модифицируется на случай ненулевых математического ожидания шума коррелированных шумов.

Уравнения ФК имеют вид:

$$\text{начальные условия: } P_{k,k} = P_{k,k-1}(I - K_k); \quad x_{1,1} = z_1,$$

$$x_{2,1} = z_2, \quad P_{1,1} = R$$

$$\text{экстраполяция: } \hat{x}_{k,k-1} = \hat{x}_{k-1,k-1} + q$$

$$P_{k,k-1} = P_{k-1,k-1} + Q$$

$$\text{фильтрация: } \hat{x}_{k,k} = \hat{x}_{k,k-1} + K_k(z_k - \hat{x}_{k,k-1});$$

$$K_k = P_{k,k}(R - P_{k,k}); \quad P_{k,k} = P_{k,k-1}(I - K_k);$$

где $\hat{x}_{k,k}$ - оценка вектора состояний по k измерениям в k -й момент времени (оценка фильтрации); $P_{k,k}$ - ковариационная матрица ошибки оценки фильтрации; $\hat{x}_{k,k-1}$ - экстраполированная оценка вектора состояний на k -й момент времени по $k-1$ измерениям; $P_{k,k-1}$ - ковариационная матрица ошибки оценки вектора экстраполяции; K_k - коэффициент усиления ФК; q - оценка среднего значения шума состояния; Q - оценка дисперсии шума состояния; R - оценка дисперсии шума измерения.

Для оценки применимости в контекстно-зависимых приложениях мобильного телефона предсказаний значений членов временного ряда полученных с применением методов линейного предсказания и экстраполяции, авторами статьи были разработаны и протестированы алгоритмы и программы для следующих известных методов:

1. Метода линейного предсказания (ЛП);
2. Метода линейной экстраполяции (ЛЭ);
3. Метода полиномиальной экстраполяции (ПЭ);
4. Метода сплайн экстраполяции (СЭ).

Основная задача при построении модели линейного предсказания (в частности модели авторегрессии) состоит в том, чтобы правильно отыскать параметры этой модели [7]. Для этой цели используется система уравнений Юла-Уолкера.

Каждый элемент выборки ЛП умножается на соответствующий весовой коэффициент. Величина зависимости x_{n+1} члена временного ряда от предыдущих n членов называется коэффициентом автокорреляции.

Экстраполяцией называют особый вид аппроксимации, при котором функция аппроксимируется вне заданного интервал [8].

1. Метод линейной экстраполяции

В данном методе, искомое значение вне интервала $[x_0, x_n]$ ищется с помощью уравнения прямой,

построенной на 2-х точках лежащих на границах интервала: $f(x) = y_n + (x - x_n)/(x_0 - x_n)(y_0 - y_n)$.

2. Метод полиномиальной экстраполяции

Полиномиальная экстраполяция [4] – метод, в котором вместо уравнения прямой линии на наборе из N точек $[x_0, x_n]$ строится полиномиальная функция P , которая проходит через эти точки: $y_i = P(x_i)$, для всех $1 \leq i \leq N$ и вычисляется значение функции в точке x_{n+1} .


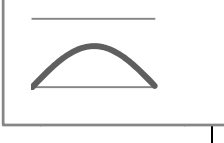


3. метод сплайн экстраполяции

В данном методе экстраполяция выполняется на основе сплайнов [9,10], с помощью которых строится результирующий полином, проходящий через все точки входного ряда.

Тестирование методов обработки контекстных данных

Для проверки эффективности работы алгоритмов фильтрации и сглаживания входных сигналов была создана тестовая среда, в которой использованы тестовые временные ряды, показанные в таблице 1. Созданная тестовая среда использовалась для отладки и анализа разработанных авторами программ, для рассмотренных в работе методов.

Таблица 1. Тестовые временные ряды

Номер теста	Тестовый временной ряд	Графическая интерпретация
1	$y = \text{tresh}$	
2	$y = \text{tresh} * \sin(3.14 * i / N)$	
3	$y = \text{tresh} + 2 * \sin(i/10) * \cos(i/10) + 2 * \sin(i)/i$	
4	$y = 10 * ([1/\text{tresh} * x] / 10 - [[1/\text{tresh} * x] / 10])$	

Для тестирования программ фильтрации и сглаживания временных рядов был выбран генератор шума на основе источника случайных чисел с нормальным (Гаусовский) распределением с заданным математическим ожиданием μ и дисперсией σ^2 для компоненты шума. В блоке сравнения тестовой среды сравнивается сглаженный зашумленный временной ряд с исходным тестовым временным рядом. Результаты сравнения характеризуют качество алгоритмов сглаживания.

Результаты тестирования

На рисунке 3 показаны результаты тестирования фильтра Кальмана для зашумлённого тестового временного ряда полученной с акселерометра мобильного телефона (тест номер 3). Дисперсия шума (сигма) равна 0.5. Из-за небольшого уровня зашумлённости и особенностей тестового ряда (имеет резкие скачки), фильтр Кальмана сглаживает не только шум, но и саму функцию, из-за чего сильно сгладилась её форма и появился эффект запаздывания. Такой же результат показал и метод простого скользящего среднего. Отфильтрованная функция сохраняет периодичность исходной тестовой функции.



Рисунок 3 - Результаты тестирования фильтра Кальмана сгенератором шума с дисперсией 0.5

На рисунке 4 показан тот же тестовый временной ряд, как и в первом случае, при этом дисперсия шума (сигма) равна 5. Как видно из рисунка, уровень шума практически больше уровня значений тестового временного ряда. Поскольку процент сглаживания высок, форма исходного тестового ряда практически не различима на уровне шумов, так же не сохраняется периодичность повторения и уровни значений его амплитуды. Таким образом, для сложных, сильно зашумленных временных рядов алгоритмы сглаживания/фильтрации показывают плохой, зачастую неприемлемый, результат.

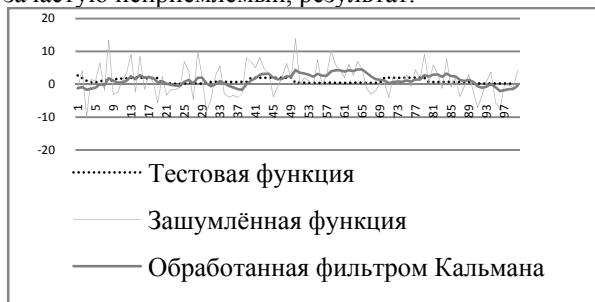


Рисунок 4 - Результаты тестирования на генераторе шума с дисперсией 5

Результаты тестирования программ для всех исследованных методов фильтрации и сглаживания временных рядов приведены в таблице 2 на примере тестов 1-4. Программы тестировались на компьютере с процессором с тактовой частотой 2.26ГГц. Длина тестового временного ряда 1000 элементов.

Таблица 2. Среднеквадратическое отклонение

Дисперсия шума Sigma=0.5				
Метод сглаживания	Тест1	Тест2	Тест3	Тест4
ФК	0,158017	0,349812	0,467901	0,325037
ПСС	0,132832	0,309358	0,788688	0,888607
ДСС	0,0767471	0,295795	0,696336	0,334083
ПЭС	0,0796013	0,299081	0,75302	0,889265
ДЭС	0,0763433	0,296688	0,785822	0,889241
Дисперсия шума Sigma=5				
Метод сглаживания	Тест1	Тест2	Тест3	Тест4
ФК	1,57463	1,56321	1,61398	1,65909
ПСС	1,2332	1,06842	1,09635	1,19313
ДСС	0,76762	0,785053	0,646097	1,10649
ПЭС	0,75865	0,696137	0,979228	0,897497
ДЭС	0,76054	0,67862	0,769288	0,846703

Кроме того, в процессе тестирования оценивалось ресурсопотребление исследуемых программ. Испытания проводились на компьютере с процессором с тактовой частотой 2.26ГГц. Контролировалось время работы программы соответствующего алгоритма и объем используемой памяти для временных рядов длиной 1000 членов. Полученные результаты приведены в таблице 3.

Таблица 3. Ресурсопотребление программ фильтрации и сглаживания временных рядов

	ФК	ПСС	ДСС	ПЭС	ДЭС
Время обработки временного ряда(мс)	1.157	2.181	2.865	3.021	5.543
Важествовано памяти(Б)	540	454	596	468	954

Тестирование программ для всех методов сглаживания с дисперсией шума равной 0,5 приведено на рисунке 5.

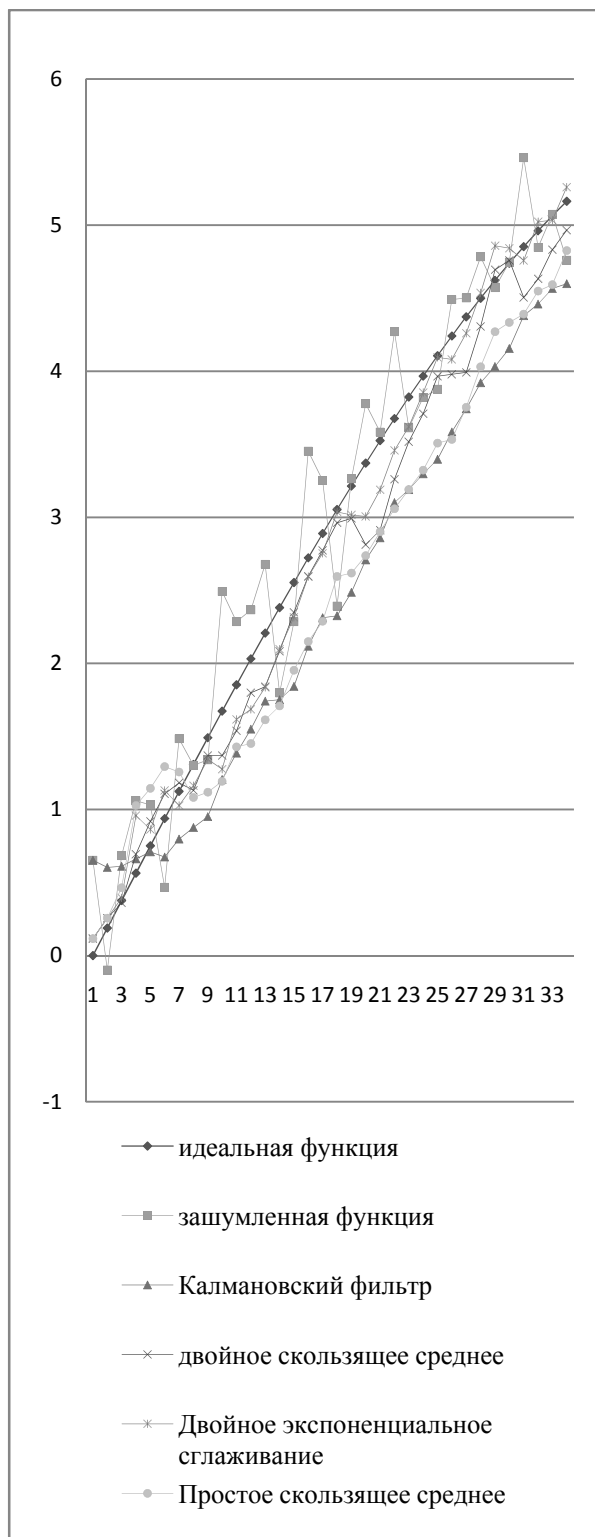


Рисунок 5 – Сглаживание исходного временного ряда с применением методов ПСС, ДСС, ДЭС, КФ

Из рисунка видно, что при гладких входных функциях методы ДСС и ДЭС дают наиболее приемлемый результат, но при этом они наиболее ресурсоёмкие.

Результаты тестирования программ реализующих исследуемые методы предсказания временных рядов приведены в таблице 4. Качество прогнозирования оценивается по величине среднеквадратической ошибки прогноза (табл.4).

В первом разделе таблицы приведены результаты прогнозирования для обучающей выборки равной двум, во втором - десяти.

Таблица 4. Среднеквадратическая ошибка прогноза

Количество элементов в обучающей выборке P=2				
Метод предсказания	тест1	тест2	тест3	тест4
ЛП	0,3333	0,6382	0,422482	2,50000
ЛЭ	0,0000	0,0033	2,524410	7,50000
ПЭ	1,0000	0,0047	3,524410	7,50000
СЭ	0,0000	0,0047	2,524410	7,50000
Количество элементов в обучающей выборке P=10				
Метод предсказания	тест1	тест2	тест3	тест4
ЛП	0,0909	0,6204	0,366496	1,66667
ЛЭ	0,0000	0,0204	3,040490	-
СЭ	0,0000	0,0204	0,197941	9,46078

ЗАКЛЮЧЕНИЕ

В контекстно-зависимых приложениях, как правило, необходимо использовать предсказания возможных значений членов временного ряда снимаемого с датчиков телефона. Использование для этих целей метода линейного предсказания ограничено тем, что предсказание возможно только для одного следующего члена временного ряда. Размер выборки ЛП сильно влияет на качество предсказания. При малой выборке, могут пропускаться долгосрочные тренды временного ряда.

Методы экстраполяции более эффективны для краткосрочных прогнозов, при которых требуется предсказание нескольких членов временного ряда. Они достаточно просты и потребляют мало ресурсов процессора мобильного телефона. Линейная экстраполяция применима для задач, в которых временной ряд изменяется без резких скачков, практически линейно. Данный метод имеет высокое быстродействие, но низкую точность на нелинейных участках входного ряда. Экстраполяция сплайнами более гибкая, чем полиномиальная, т.к. при добавлении нового члена временного ряда в интервал экстраполяции не требуется построение полинома на всём интервале, а лишь на ближайших к искомой точках.

Контекстно-зависимые приложения, работающие с зашумленными исходными данными, формируют ошибочные результаты. Фильтрация или сглаживание соответствующих временных рядов является обязательным этапом, повышающим достоверность исходных данных. Проведенные эксперименты подтвердили эффективность применения рассмотренных методов подавления случайной шумовой компоненты для реальных сигналов, снимаемых с датчиков мобильных устройств. Созданная библиотека программ сглаживания и фильтрации, а так же прогнозирования временных рядов позволяет создавать новые приложения для решения контекстно-зависимых задач.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Oviatt, 2003] Oviatt S. L. Multimodal Interfaces/Oviatt S. L. // The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, Jacko J. and Sears A. (Eds.). Mahwah, NJ: Lawrence Erlbaum Assoc. 2003. P. 286–304.
- [Kendall, 1983] Kendall M. G., Kendall's advanced theory of statistics/ Kendall, M. G., Stuart, A., Ord, J. K. // Vol. 3, Hodder Arnold, London
- [Лукашин, 2003] Лукашин Ю. П. Адаптивные методы краткосрочного прогнозирования временных рядов / Лукашин Ю. П. // М. : Финансы и статистика, 2003. – 415 с.
- [Gardner, 1980] Gardner Jr. Forecasting with Exponential Smoothing: Some Guidelines for Model Selection/ Gardner Jr., Everette S. and David G. Dannenbring // In Decision Sciences, 11., 370-383, 1980.
- [Bowerman, 1993] Bowerman, Forecasting and Time Series: An Applied Approach/Bowerman, Bruce J. and Richard T. O'Connell. // Duxbury Thomson Learning, 1993.
- [Згуровский, 1995] Згуровский М. З. Аналитические методы калмановской фильтрации для систем с априорной неопределённостью / М. З. Згуровский, В. Н. Подладчиков – К. : Наукова думка, 1995. – 283 с.
- [Грэй, 1980] Грэй А. Х. Линейное предсказание речи/Грэй А. Х. // М. связь.-1980.-25 стр.
- [Казаков, 1987] Казаков И.Е. Методы оптимизации стохастических систем/Казаков И.Е., Гладков Д.И. // М. 1987.- 304
- [Brezinski, 1991] Brezinski C. Extrapolation Methods. Theory and Practice/ Brezinski C. Redivo Zaglia M. // North-Holland. - 1991.
- [Роджерс, 2001] Роджерс Д. Математические основы машинной графики/Роджерс Д., Адамс Дж. // М.: Мир, 2001. — ISBN 5-03-002143-4

CONTEXT DATA PROCESSING FOR INTELLIGENT SYSTEM

Kyselova A.G, Kyselov G.D.

*National Technical University of Ukraine
"Kiev Polytechnic Institute"*

kiselev_gd@bigmir.net

kiseleva_anna@ukr.net

INTRODUCTION

Context-aware system is employed in a variety of contemporary information systems in order to enhance the naturalness, flexibility and convenience of user interface. Context is a mode of communication according to human senses (sight, touch, hearing, smell, and taste) or type of computer input devices (input devices equivalent to human senses: cameras (sight), haptic sensors (touch), microphones (hearing), olfactory (smell) and input devices that do not map directly to human senses: keyboard, mouse, writing tablet, motion input and others). A multimodal interface provides several distinct tools for input and output of data. Data fusion input data is much researched in the field of sensor networks where the primary motivation is to reduce communication costs by the integration of similar data sources.

Keywords – context-aware system, context, smoothing, prediction.

RAW DATA-PROCESSING PACKAGE

The raw data-processing package to evaluate the reliability of data information for sensors environment. That it consist of algorithms that smoothed data because the data information influences by environment status may be imperfect.

An often-used technique in industry is "smoothing". This technique, when properly applied, reveals more clearly the underlying trend, seasonal and cyclic components.

There are three distinct groups of smoothing methods

- Averaging Methods
- Exponential Smoothing Methods
- Kalman filter

And prediction method:

- Linear prediction
- Spline extrapolation
- Polynomial extrapolation
- Line extrapolation

CONCLUSION

The goal of testing is to estimate algorithm's quality, trace functionality issue, and estimate resources consumption and estimate of difference between the data after the algorithm with original data.

The result of testing is choose algorithm in which general restrictions typical for mobile environments shall be kept in mind:

- low memory consumption
- error tolerance for the input data
- speed



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 681.3.06

COGNITIVE LINGUISTIC PRESENTATIONS OF LANGUAGE STRUCTURES IN ARTIFICIAL INTELLIGENCE AND MACHINE TRANSLATION SYSTEMS

Kozerenko E.B.

Institute of Informatics Problems of the Russian Academy of Sciences

kozerenko@mail.ru

The paper focuses on the issues of establishing semantic content of syntactic structures in the English and Russian languages for the tasks of machine translation and knowledge management. The problem of establishing transferable language phrase structures is considered. The approach employed is based on generalized cognitive entities manifested in the categorial systems of a subset of natural languages (English and Russian in our case) and functional roles of language units in a sentence. A declarative module of syntactical processor was designed and implemented within the framework of machine translation system “Cognitive Translator” and a number of intelligent knowledge-based systems.

Keywords: machine translation, phrase structures, syntax, semantics, transfer.

INTRODUCTION

The present state of research and development in the field of machine translation and multilingual systems design requires new methods of linguistic reality presentations capturing the intricate features of natural languages and comprising the facilities of the already existing approaches. The crucial problem to be faced is *categorization* of linguistic phenomena. Of special concern are the syntactic-semantic structures since neither constituency grammar nor dependency grammar alone gives the complete expressive means for such natural language properties as syntactic ambiguity and synonymy.

Translation is a creative and sophisticated human activity, hence, producing automatically a high-quality translation of an arbitrary text from one language to another is a task too far from its complete implementation. However, for simpler tasks, such as acquiring information on the Web, getting acquainted with subject domain information, etc., rough translation output without post editing can be quite adequate. One of the domains where MT works best is scientific discourse.

Of the three forms of translation performed by man: written translation, consecutive interpretation and simultaneous interpretation, the one which is nearest to the real-time machine translation is simultaneous interpretation (SI). Therefore, the recommendations for SI are of prime interest to MT designers, as they propose more implementable solutions for lexical grammatical transformations than the first two forms.

Syntactically languages are most different in the basic word order of verbs, subjects, and objects in declarative clauses. English is an SVO language, while Russian has a comparatively flexible word order. The syntactic distinction is connected with a semantic distinction in the way languages map underlying cognitive structures onto language patterns, which should be envisaged in MT implementations [Nirenburg, et al.,1992]. Besides, there exist syntactic constructions specific of a given language (such as, for example, English constructions with existential “there” and “it” as formal subjects). Sometimes, a word may have translation to a word of another part-of-speech in the target language, a word combination, or even a clause, as the English *implementable* is best translated into Russian as *kotoryi vozmozhno realizovat'* (*which can be implemented*). To overcome these differences the categorial and functional features of the two languages were considered, and structures of the input were made conformed to the rules of the target language by applying contrastive linguistic knowledge for implementation of the transfer model.

A suitable formalism is indispensable for an algorithmic presentation of the established language transfer rules, and the language of Cognitive Transfer Structures (CTS) was developed based on rational mechanisms for language structures generation and feature unification. The formalism developed for presentation of syntactic structures for the English-Russian machine translation is a variant of unification grammar and comprises over two hundred rules and it was implemented within the framework of machine translation system “Cognitive Translator” and a number of intelligent knowledge-based systems.

1. SI Techniques for Handling Syntactic Structures

Segmentation and unification of utterances in the course of translation is a major task for human professional interpreters. They would even say that syntax is “interpreter’s enemy”. The selectivity of languages as to the choice of specific characteristics of description of one and the same situation results in numerous distinctions, and one of the most crucial of them is the degree of particularity in conveying a referential situation. Therefore, a situation which in one language is described by means of one specific feature, in another language may require two or more characteristics. Thus, in many cases the English language is more economical (about thirty percent, according to the reports of simultaneous interpreters) [Visson, 1989, Visson, 1991] in expressing a thought than Russian.

A very good illustration of this phenomenon is attributive word combinations of the “stone wall” type which when being translated into Russian in many cases require numerous additions. On the other hand, Russian input in some cases may result in an expanded English translation.

In practice the technique applied to overcome this problem is *utterance segmentation* which consists in sectioning a source Russian sentence into two or more utterances in the resulting English sentence.

Another important rule is the least possible change of word order. But this inflicts other unavoidable transformations, and not all of them are implementable within the framework of machine translation. For example, the general rule for interpreters: a Russian noun which appears at the very beginning of a sentence and has the form of an oblique case, i.e. indirect object standing at the beginning of a Russian sentence, should be transformed into the subject of an English sentence notwithstanding its initial syntactic role

e.g. *Na vstreche dogovorilis’...* (At the meeting agreed...)

should be translated as -

The meeting reached an agreement...

This transformation performed in the course of human simultaneous interpretation appears to be unattainable to a machine translator at the present state of the art. The requirement of denotational equivalence involves numerous lexical grammatical shifts which cause transformations of the semantic structure of an utterance [Visson, 1989, Visson, 1991]. Another regular semantic shift, that of substituting a predicate of action by the predicate of state.

e.g. *He is a member of the college team.* (A predicate of state).

On igraet v studencheskoi komande. (He plays in the students’ team. A predicate of action).

Moreover, the existence of such shifts within the real text corpora inflicts complications for one more computational linguistics problem, that is text alignment, which in some cases may appear even intractable.

The following SI techniques appeared to be of use for MT design in the course of our development.

(1) Full translation of lexical grammatical forms is applied when these forms completely correspond to each other both in the source and the target languages as to their form, function and meaning.

(2) Null translation is applied when a grammatical form exists in the source and target languages but is used differently for explicating a certain referential situation.

(3) Partial translation is used when one and the same grammatical form has several content functions which differ in the source and target languages.

(4) Functional substitution is employed when the functions and meanings of grammatical forms in the source and target languages differ. In that case the source form can be substituted by a form of another type in the target language on the basis of their functional identity.

(5) Assimilation is a device applied for translating grammatical forms constituting compound structure, and the combinability features of these forms differ in the source and target languages.

(6) Conversion is used for substituting a form of one category by a form of another category, and is conditioned by the combinability rules difference in the source and target languages.

(7) Antonyms employment is used for eliminating a conflict between lexical and grammatical combinability of language units in the source and target languages.

Thus it is obvious that the search for equivalence should be carried out starting with the establishment of semantic equivalence of patterns notwithstanding their structural dissimilarity. Pattern-matching approach for the English – Russian transfer was assumed, and the segmentation of structures of the source language was performed on the basis of functional transfer fields which were established via contrastive study of the two languages.

2. Cognitive and Functional Aspects of Transfer Modelling

The machine translation technique employed presupposes three stages: analysis, transfer and generation. The stage of analysis results in parse representing the structure of the input sentences. Transfer is a bridge between the parse structure of the source language and the input to the generation procedure for the target language. At this stage the transformation is performed of one parse tree (applicable for the source language presentation) into another tree (presenting the target language). Thus syntactic transformations imply the mapping of one tree structure to another.

It is very important that *a parse for MT differs from parses required for other purposes*. Thus the grammar formalisms developed for a unilingual situation (phrase structure rules systems for the English language) [Grover et al., 1993] would give an untransferable parse in many crucial situations. For example, just one English phrase structure rule for simple sentence would

suffice for grammar parse without translation, but for the English – Russian transfer a multiple structure of possible parses is required depending on the specific finite verbal form constituting the sentence. And to overcome this, an accurate scheme for all the particular verbal form cases should be designed.

The segmentation of phrase patterns used for the input language parse was carried out with the consideration of semantics to be reproduced via the target language means. Both the most important universals such as enumeration, comparison, modality patterns, etc., and less general structures were singled out and assigned corresponding target language equivalents.

Consider an example of a phrase structure conveying the modal meaning of obligation: “...*the task to be carried out...*”. In other words, the meaning of this phrase can be rendered as “...*the task that should be carried out...*”. The Infinitive phrase in the English language gives the regular way of expressive means compression without the loss of semantic value. A literary translation in Russian requires the second way of presenting the same idea of obligation. However in this specific case a “reduced” translation variant is also possible which consists in the introduction of the subordinate conjunction “*chtoby*” – “*so that*”, between the noun and the modifying Infinitive. The parse rule would look like:

NP(to) → NP VPto

And the generation rule would be presented as:

NP(to) → NP Punct. {comma} Conj.(*chtoby*) VPto

Special attention is required for the problem of passive constructions transfer. As in the phrase “*was considered*”. The rules for simultaneous translation (which in many cases is similar to the real time machine translation performance and can be a source of compromise decisions for phrase structure design) requires the transformation of the English Subject into the Direct Object (Russian, Accusative Case) standing in the first position in a sentence and the passive verbal form would produce an impersonal verbal form in Russian. However such transformation proved to be of considerable danger to the whole sentence structure and might cause an unpredictable generation result. Hence, for many cases a more clumsy, though robust method of a passive construction generation was accepted: the one similar to the English “*be + Past Participle*”:

$V(aux_ppt) \rightarrow V(aux) PPt$

Actually the process of transfer goes across the functional – categorial values of language units. A language structure which can be subjected to transfer has to be semantically complete from the point of view of its function. The cases of categorial shifts, in particular, when the technique of conversion is employed, require special treatment: the categorial shift of a syntax unit is determined by the functional role of this unit in a sentence (e.g. noun as a modifier → adjective). *Only by creating the centaur concepts.. 'constituency-dependency', 'linearity-nonlinearity', 'form-function', etc. can we get a reasonably clear picture of linguistic reality* [Shaumyan, 1987].

The starting idea for the language structures segmentation strategy was the notion of functional semantic fields. The system of grammar units, classes and categories with generalized content supplementary to the content of lexical units, together with the rules of their functioning, is a system which in the end serves for transmission of generalized categories and structures of mental content which lie the foundation of utterance sense, and constitute the basis of language grammar formation [Bondarko, 2001].

As it was exhibited in [Kibrik, 2001] language coding technique is to a great extent determined by the deep semantic structure, and of considerable advantage is such a presentation method which takes for the starting point the semantic level, and particular semantic units are confronted with the coding devices expressing them. The approach of functional semantics concurs in many aspects with the categorial grammar. The system of sentence members (functional roles) is being modified, but its essence is preserved in the new facts qualification via the traditional categories [Zolotova, 2001].

The transferability of phrase structures is conditioned by the choice of language units in the source and target languages belonging to the same Cognitive Transfer Spaces (CTS), notwithstanding the difference or coincidence of their traditional categorial values. A set of basic CTS was singled out and language patterns employed for conveying the functional meanings of interest were examined.

- Primary Predication CTS (non-inverted) bearing the Tense – Aspect – Voice features; this field mainly includes all possible complexes of finite verbal forms and tensed verbal phrase structures.
- Secondary Predication CTS bearing the features of verbal modifiers for the Primary Predication CTS. Included here are the non-finite verbal forms and constructions, and subordinate clauses comprising the finite verbal forms. All these are united by the functional meanings they convey, e.g. qualification, circumstance, taxis (ordering of actions), etc.
- Nomination and Relativity CTS: language structures performing the nominative functions (including the sentential units) comprise this field.
- Modality and Mood CTS: language means expressing modality, subjunctivity and conditionality are included here. Here the transfer goes across the regular grammatical forms and lexical means (modal verbs and word combinations) including phrasal units.
- Connectivity CTS: included here are lexical – syntactic means employed for concatenation of similar syntactic groups and subordination of syntactic structures.
- Attributiveness CTS: adjectives and adjectival phrases in all possible forms and degrees comprise the semantic backbone of this field; included here are also other nominal modifiers, such as nominative language units and structures (*stone wall* constructions, prepositional genitives – *of* –phrases),

and other dispersed language means which are isofunctional to the backbone units.

- Metrics and Parameters CTS: this field comprises language means for presenting entities in terms of parameters and values, measures, numerical information.
- Partition CTS: included in this field are language units and phrase structures conveying partition and quantification (e.g. *some of, part of, each of, etc.*).
- Orientation CTS: this field comprises language means for rendering the meaning of space orientation (both static, and dynamic).
- Determination CTS: a very specific field which comprises the units and structures that perform the function of determiner (e.g. the Article, which is a good example for grammar – lexical transfer from English into Russian, since in Russian there exist no such grammatical category; demonstrative pronouns, etc.).
- Existentiality CTS: language means based on *be*-group constructions and synonymous structures (e.g. sentential units with existential *there* and *it* as a subject: *there is...; there exists...; etc.*).
- Negation CTS: lexical – syntactic structures conveying negation (e.g. *nowhere to be seen, etc.*).
- Reflexivity CTS: this field is of specific character since the transfer of reflexivity meaning goes across lexical - syntactic – morphological levels.
- Emphasis – Interrogation CTS: language means comprising this field are grouped together since they employ grammar inversion in English.
- Dispersion CTS: individual language structures specific for a given language are included here; these are presented as phrasal templates which include constant and variable elements.

The set of functional meanings together with their categorial embodiments serve the source of constraints for the unification mechanism in the formal presentation of our grammar. The formalism developed employs feature-based parse, and head-feature inheritance for phrase structures which are singled out on the basis of functional identity in the source and target languages. To implement the feature-valued inheritance sometimes broader contexts are taken.

3. Statistical Approach to Machine Translation

In statistical machine translation (SMT) the task of translating from one natural language into another is treated as a machine learning problem. This means that via training on a very large number of hand-made translation samples the SMT algorithms master the rules of translation automatically. The application of statistical models has considerably advanced the area of machine translation since the last decade of the previous century, however now new ideas and methods appear aimed at creating systems that efficiently combine symbolic and statistical approaches comprising different models.

Both the paradigms move towards each other:

more and more linguistics is being introduced into stochastic models of machine translation, and the rule-based systems include statistics into their linguistic rule systems. The procedures of analysis and translation are enhanced by the statistical data, which taken into consideration by the “translation engine” for disambiguation of language structures. The stochastic approach to natural language processing originates from the projects in speech and characters recognition and spellcheckers. The main method for solving numerous problems, including the part of speech establishment and tagging, is the Bayesian approach. The architecture of stochastic systems is based on the dynamic programming algorithm.

Machine learning is rooted in the stochastic research paradigm. The training algorithms can be of the two types: supervised and unsupervised. An unsupervised algorithm should infer a model capable for generalization of the new data, and this inference should be based on the data alone. A supervised algorithm is trained on a set of correct responses to the data from the training set so that the inferred model provides more accurate decisions. The object of machine learning is the automatic inference of the model for some subject area basing on the data from this area. Thus a system learning, for example, syntactic rules should be supplied with a basic set of phrase structure rules.

The widely used methods lately have been the *N*-grams which capture many intricacies of syntactic and semantic structures, *N*-grams of variable length in particular, introduction of semantic information into *N*-grams. The statistical models are built on the data obtained from the parallel corpora in different languages. Usually the texts are compared within language pairs. The text in the language from which the translation should be done is called the *source text*, and the text which is its translation is called the *target text*. Correspondently the languages are also called the source language and the target language (i.e. the language of translation).

The main method of extracting the data about the matches between the source and target languages and texts is the alignment of parallel texts. The result of this procedure is also called *alignment* and it is designated by *A*. The probability characteristics of alignments are employed in the algorithms of statistical machine translation. Hence, the *alignment* and the probability distribution are the key notions in these models description.

The following notations are employed in this paper: the symbol *P* denotes the probability distributions in the most general sense, and the symbol *p* denotes the probability distribution based on some particular model. The main attention in this paper is given to the description of various methods employed for parallel texts alignment, as the results of the alignment procedure determine the accuracy and adequacy of translation. We focus on the *linguistic filters* that are being introduced in the form of data structures and rules into the statistical translation models. The models under consideration are illustrated basing on the bilingual model for the Russian and

English language pair. However, the similar methods are applicable for the alignments and translations of the Russian texts into the French and German languages, as well as other European languages.

4. Methods of parallel texts alignment

The statistical approaches to parallel texts alignment are aimed at establishing the most probable alignment A for the two given parallel texts S and T :

$$\arg \max_A P(A|S,T) = \arg \max_A P(A,S,T) \quad (1)$$

For estimation of the probability values indicated in this expression the most frequently used methods present the parallel texts in the form of aligned sentence sequences (B_1, \dots, B_K) . The probability of each sequence is independent from the probabilities of other sequences, and it depends on the sentences in the given sequence only [Gale, Church, 1993]. Then

$$P(A, S, T) \approx \prod_{k=1}^K P(B_k) \quad (2)$$

This method takes into account the length of sentences in the source language and in the target language measured in symbols. The longer sentence in one languages will correspond to the longer sentence in the other language. This approach gives stable results for similar languages and literal translation. The more finely tuned mechanisms of matching are provided by the methods of lexical alignment. Thus in [Chen, 1993] the method of alignment by means of creating the model for consecutive word-by-word translation is presented. The best alignment result will be the one which maximizes the probability of a corpus generation with the given translation model. For the alignment of the two texts S and T they should be split into the sequences of sentence chains. A chain contains zero or more sentences in each of the two languages, and the sequence of chains covers the whole corpus

$$B_k = (S_{a_k}, \dots, S_{b_k}; t_{c_k}, \dots, t_{d_k}) \quad (3)$$

Then the most probable alignment $A = B_1, \dots, B_{m_A}$ of the given corpus is determined by the following expression, and the chains of sentences do not depend on each other:

$$\arg \max_A P(S,T,A) = \arg \max_A P(L) \prod_{k=1}^{m_A} P(B_k) \quad (4)$$

where $P(L)$ denotes the probability of the L chains being generated. The translation model employed in this approach is extremely simplified and does not take into account the factor of the word order in a sentence and the possibility of the fact that a word in the source text can correspond to more than one word in the text of translation. In this model the word chains are used, and they are limited to the 1:1, 0:1 и 1:0 matches. The essence of the model consists in the idea that if one

word is usually translated by the word of another language, then the probability of the word chains matches 1:1 will be very high, and much higher than the product of probabilities of the 1:0 and 0:1 word chains matches where the given word occurs. And the program chooses the most probable alignment variant.

The translation model based on the word-by-word alignment (we employ this model for the Russian and English parallel texts) will be as follows:

$$P(r|e) = \frac{1}{Z} \sum_{a_1=0}^l \dots \sum_{a_m=0}^l \prod_{j=1}^m P(r_j | e_{a_j}) \quad (5)$$

where e is a sentence in English; l is the length of e expressed in words; r is a sentence in Russian; m is the length of r ; r_j is the j -th word in r ; a_j is the position in e , with which the r_j is aligned; $P(w_r | w_e)$ is the probability of translation, i.e. the probability of the w_r appearing in the Russian sentence if the corresponding w_e occurs in the English sentence, and Z is the normalization constant. For a particular alignment m probabilities of translations are multiplied, and the individual translations are independent one from another.

However, the above stated approach based on the word-by-word comparison and in no way accounting for the links between words and phrases does not give optimal results for the alignment of the Russian language and the English language texts, for there are certain structural differences between these languages, and in translation there can be considerable transformations. If the languages under consideration are structurally different, the methods are used oriented at the introduction of grammar knowledge, for example, the alignment methods based on the words that belong to particular parts of speech [Masahiko, Yamazaki, 1996] are employed. In this case the auxiliary words are not taken into account. For the employment of these methods the part of speech tagging of the parallel texts should be performed. The methods of parallel texts alignment for creating statistical translation models were, as a rule, developed on the basis of word matches: each word in the chain of a source text had to be matched with the corresponding word in the chain of the target text (in the language of translation) and vice versa. However, quite often it is difficult to establish which words of the target and source chains correspond to each other. Special problems arise when attempting to align the words inside idioms, in case of translational paraphrases, in free translation and when the auxiliary words are omitted. The alignment of two word chains can be quite sophisticated. It is necessary to take into account various transpositions of words, omissions, insertions, and the alignments between different language levels: when a word in the source text corresponds to a phrase in the target text, and the opposite situation. The most general definition of the word-based alignment is given in [Och, Ney, 2000]. The phrase-based translation model, or the alignment

template model [Och, Ney, 2004] and other similar approaches have greatly advanced the development of machine translation technology due to the extension of the basic translation units from words to phrases, i.e. the substrings of arbitrary size. However, the phrases of this statistical machine translation model are not the phrases in the meaning of any existing syntax theory or grammar formalism, thus, for example, a phrase can be like «alignments the», etc.

5. Linguistic filters on the basis of the Cognitive Transfer Grammar

The key idea of our linguistic framework is cognitive cross-linguistic study of what can be called *configurational semantics*, i.e. the systemic study of the language mechanisms of patterns production, and what meanings are conveyed by the established types of configurations. We explore the sets of meanings fixed in grammar systems of the languages under study. Our studies are focused on the types of meanings outside the scope of lexical semantics, and we consider the lexical semantics when the meanings which we denote as configurational, have expression at the lexical level. The importance of this aspect is connected with the fact that natural languages are selective as to the specific structures they employ to represent the referential situation. However, it is always possible to establish configurations which perform the same function across different languages (i.e. isofunctional structures). The parse aimed at transfer procedures requires a semantic grammar and cannot be efficiently implemented through a combination of monolingual grammars.

In the previously formulated Cognitive Transfer Grammar (CTG) [Kozerenko, 2003], [Kozerenko, 2008] the functional meanings of language structures are determined by the categorial values of head elements. The probability characteristics are introduced into the rules of the unification grammar as weights assigned to the parse trees.

In the Cognitive Transfer Grammar the basic structures are the *transfemes* [Kozerenko, 2008]. A *transfeme* is a unit of cognitive transfer establishing the functional semantic correspondence between the structures of the source language L_s and the structures of the target language L_T . For the alignment of parallel texts the transfemes are given as the rewrite rules in which the left part is a nonterminal symbol, and the right part are the aligned pairs of chains of terminal and nonterminal symbols which belong to the source and target languages :

$$T \rightarrow \langle \rho, \alpha, \sim \rangle, \quad (6)$$

where T is a nonterminal symbol, ρ and α are chains on terminal and nonterminal symbols which belong to the Russian and English languages, and \sim is a symbol of correspondence between the nonterminal symbols occurring in ρ and the nonterminal symbols occurring in α . In the course of parallel texts alignment on the basis of the CTG the derivation process begins with a pair of the linked starting symbols S_ρ and S_α , then at each

step the linked nonterminal symbols are rewritten pairwise with the use of the two components of a single rule.

For automatic extraction of the rules on the basis of CTG from parallel texts these texts should be previously aligned by sentences and words. The extracted rules base on the wordwise alignments in such a way that at first the the starting phrase pairs are identified with the use of the same criterion as the majority of statistical models of translation employing the phrase-based approach [Och, Ney, 2004], which means that there should be at least one word inside a phrase in one language aligned with some word inside a phrase in another language, but no word inside a phrase in one language can be aligned with any word outside its pair phrase in another language.

Cognitive Transfer Grammar is a generative unification grammar having a hierarchical structure and reflecting a major part of language transformations employed in the process of translation from one language into another. Besides, basing on the experimental data obtained from the corpora study the CTG rules are supplied with the weights of possible derivation variants.

Definition.

Cognitive Transfer Grammar G_{CT} is a set

$$G_{CT} = \{T_{L_1}, T_{L_2}, N_{L_1}, N_{L_2}, P_{CA}, P_{CT}, S_{L_1}, S_{L_2}, M, D\} \quad (7)$$

Where T_{L_1}, T_{L_2} are the sets of terminal symbols of the languages L_1 and L_2 ; N_{L_1}, N_{L_2} are the sets of nonterminal symbols of the languages L_1 and L_2 ; P_{CA}, P_{CT} are the rules of analysis and synthesis on the basis of the cognitive transfer; S_{L_1}, S_{L_2} are a pair of the starting symbols of the languages L_1 и L_2 with which the process of analysis and alignment of sentences is initiated; M is the function of establishing the correlations between the structures of the languages L_1 and L_2 ; D is the function assigning the probability values to each rule from the sets P_{CA}, P_{CT} .

Ambiguity is an immanent feature of the natural language and it is a cause of major difficulties in machine translation implementation. Ambiguous and polysemous syntactic structures are taken into account in the further development of the CTG mechanisms, which is the *multivariant* CTG, and the implementations of the multivariant CTG data structures are used as linguistic filters in statistical translation models. These data structures are called *multivariant cognitive transfer structures* (MCTS). The general presentation of the MCTS syntax is as follows :

MCTS { MCTS <identifier> MCTS <weight> MCTS <tag> } →

<Input phrase structure and the set of its features and values > →

<Head-driven transfer scheme> →
 <Generated phrase structure and its set of features and values – variant 1> < weight 1>
 <Generated phrase structure and its set of features and values – variant 2> < weight 2>
 <Generated phrase structure and its set of features and values – variant N> < weight N> .

The new multivariant CTG captures the polysemy of syntactic structures and the mechanisms of disambiguation basing on the statistical data are introduced into the systems of parse and transfer rules, possible contexts of language structures are taken into account.

The multivariant CTG provides an extensible platform for the development of machine translation and knowledge extraction systems. At present the CTG principles are employed for development of the rule systems for the Russian-French and Russian-German language pairs. A new hybrid approach to construction of the models for machine translation and other natural language processing systems bridges the gap between symbolic and stochastic paradigms. The new training data sets are introduced into the linguistic knowledge base for upgrading the rule systems. The linguistic filters employed for reduction of the noise rules generated in the process of learning are based on the *cognitive transfer spaces* which comprise major groups of cross-lingual functional synonyms.

CONCLUSION

The urgency of the new hybrid methods of language objects presentation is caused by the demand for the optimal combination of advantages of the two research paradigms: logical linguistic modelling employing the designed rules and stochastic approach based on machine learning. This development is of special importance for the tasks of structural analysis and computer modelling of the full text scientific and patent documents. One of the latest developments is connected with implementing the natural language web service for the multilingual search and analysis of financial information.

The Cognitive Transfer approach provides a sound and extensible platform for simulation of cross-lingual syntactic-semantic transfer and can be applied to a greater number of languages (especially with similar categorial feature-value structures). However, the problems of discontinuity, reference resolution and ambiguity, though partially treated, still remain. Further research is connected with introducing special feature-value augmentations to the existing presentations for tracing the discontinuous structures, specifying the semantic values of particular head features and verbal subcategorization frames, and numerous phrasal units adjustment.

Our focus on configurations provides high *portability* to the language processing software designed under these principles: we can operate with a lexicon which has only standard linguistic information including morphological characteristics, part of speech information and the indication of transitivity for verbs.

We have evidence that by focusing on the

cognitive transfer principles we will be able to build natural language translation systems which are more accurate, efficient, and scalable than those which currently exist. It is the goal of the current development to advance this method by means of the language engineering environment developed in the course of the current project.

The approach taken would be important in further development of educational programs for computer science and computational linguistics courses. Educational relevance of the methods proposed lies in deeper understanding of uniform cognitive mechanisms employed in particular language embodiments of semantic structures.

The work was supported by the Russian Foundation for Basic Research, Grant 11-06-00476-a “Cognitive Linguistic Representations and Disambiguation of Language Structures for Intelligent Knowledge Management and Machine Translation Systems”.

BIBLIOGRAPHY LIST

- [Nirenburg, et al.,1992] Nirenburg, S., Carbonell, J., Tomita, M., and Goodman, K. Machine Translation: A Knowledge-based Approach. Morgan Kaufmann. 1992.
- [Visson, 1989] Visson, L. Syntactical Problems for the Russian-English Interpreter. No Uncertain Terms, FBIS, vol. 4, N 2, 1989, 2-8.
- [Visson, 1991] Visson, L. From Russian Into English: An Introduction to Simultaneous Interpretation. Ann Arbor, Michigan: Ardis, 1991.
- [Grover et al., 1993] Grover, C., Carroll, J. and Briscoe, T. The Alvey Natural Language Tools Grammar (4-th Release). Technical Report, 1993, Computer Laboratory, University of Cambridge, 1993.
- [Shaumyan, 1987] Shaumyan, S. A Semiotic Theory of Language. Indiana University Press, 1987.
- [Bondarko, 2001] Bondarko A.V. Printsipy Funktsional'noi Grammatiki I Voprosy Aspektologhii. Moskwa, URSS, 2001 /Functional Grammar Principles and Aspectology Questions. Moscow, URSS, 2001 (In Russian).
- [Kibrik, 2001] Kibrik A.E. Ocherki po Obstchim I Prikladnym Voprosam Yazykoznaniiya. Moskwa, URSS, 2002. /Studies in General and Applied Linguistics Issues. Second Edition. Moscow, URSS, 2001 (In Russian).
- [Zolotova, 2001] Zolotova G.A. Kommunikativnye Aspekty Russkogo Sintaksisa. Moskwa, URSS, 2001/ Communicative Principles of the Russian Syntax. Moscow, URSS, 2001 (In Russian).
- [Gale, Church, 1993] Gale W. A., Church K. W. A program for aligning sentences in bilingual corpora // Computational Linguistics, 1993. Vol. 19. P. 75–102.
- [Chen, 1993] Chen S. F. Aligning sentences in bilingual corpora using lexical information // Proceedings of the 31st Annual Conference of the Association for Computational Linguistics, 1993. P. 9–16.
- [Masahiko, Yamazaki, 1996] Masahiko H., Yamazaki T. High-performance bilingual text alignment using statistical and dictionary information // ACL 34, 1996. P. 131–138.
- [Och, Ney, 2000] Och F. J., Ney H. A comparison of alignment models for statistical machine translation // COLING'00: The 18th International Conference on Computational Linguistics. Saarbrucken, Germany, 2000. P. 1086–1090.
- [Och, Ney, 2004] Och F. J., Ney H. The alignment template approach to statistical machine translation // Computational Linguistics, 2004. Vol. 30. P. 417–449.

[Kozerenko, 2003] Kozerenko E. B. Cognitive Approach to Language Structure Segmentation for Machine Translation Algorithms // Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications, Las Vegas, USA, 2003. – CSREA Press, 2003. P. 49–55.

[Kozerenko, 2008] Kozerenko E. Features and Categories Design for the English-Russian Transfer Model // Advances in Natural Language Processing and Applications Research in Computing Science, 2008. Vol. 33. P. 123–138.

КОГНИТИВНО-ЛИНГВИСТИЧЕСКИЕ ПРЕДСТАВЛЕНИЯ ЯЗЫКОВЫХ СТРУКТУР В СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И МАШИННОГО ПЕРЕВОДА

Козеренко Е.Б.

*Учреждение Российской академии наук
Институт проблем информатики РАН,
Москва, Российская Федерация*

kozerenko@mail.ru

В работе рассматриваются вопросы создания представлений семантики синтаксических структур для русского и английского языков в системах машинного перевода и обработки знаний. Рассматривается проблема сегментации предложений для установления фразовых структур, переводимых методом трансфера. Применяется подход на основе обобщенных когнитивных сущностей, которые проявляются, как в системах грамматических категорий ряда европейских языков, так и в функциональных ролях языковых единиц в предложении.

Разработан и реализован декларативный модуль синтаксического анализа и синтеза системы машинного перевода “Cognitive Translator”; данный подход также использовался при создании лингвистических процессоров интеллектуальных систем обработки знаний.

Keywords: машинный перевод, фразовые структуры, синтаксис, семантика, трансфер.

ВВЕДЕНИЕ

Данная работа посвящена актуальным проблемам создания семантико-синтаксических представлений для систем машинного перевода и извлечения знаний из естественно-языковых текстов. Целью наших исследований является построение целостной лингвистической модели на основе синергетического подхода, использующего лингвистические знания, статистические методы и механизмы машинного обучения для извлечения новых грамматических правил из текстовых корпусов и разрешения неоднозначности. Для формализации лингвистических знаний используется когнитивная трансферная грамматика (КТГ), являющаяся семантически мотивированным вариантом унификационно-порождающей грамматики. Для подготовки обучающих компонентов систем и получения статистических

данных о языковых структурах создается многоязычный лингвистический ресурс, представляющий собой банк синтаксических деревьев (Treebank) и корпус семантически выровненных параллельных текстов на русском, английском и ряде других европейских языков.

ОСНОВНОЕ СОДЕРЖАНИЕ

Современный период развития исследований и разработок в области машинного перевода и систем извлечения знаний из текстов характеризуется интенсивным процессом «гибридизации» подходов и моделей. Потребность в этом носит объективный характер. Значительные вычислительные ресурсы современных систем позволяют накапливать и использовать ранее переведенные текстовые фрагменты, обеспечивать машинный перевод, основанный на прецедентах эффективно поддерживать компоненту «переводческой памяти».

Для машинного перевода наиболее сложной проблемой является реализация языковых трансформаций, которые необходимо производить при переводе с одного языка на другой. Текущий этап развития систем машинного перевода характеризуется исследованиями в области когнитивной семантики, вероятностных языковых моделей и разработкой семантико-синтаксических представлений, учитывающих многозначность и неоднозначность синтаксических структур.

Предлагаемый нами подход на основе когнитивной трансферной грамматики (КТГ) дает возможность компактного представления структуры составляющих предложения (грамматика фразовых структур), с одной стороны, а, с другой стороны, учитывает механизмы зависимости между узлами дерева предложения. Ядро КТГ составляют прототипические структуры исследуемых языков (в исходной модели – русского и английского), их наиболее вероятные позиции в предложении, а также статистические данные о дистрибутивных характеристиках структур (т.е. информация о контекстных условиях употребления исследуемых объектов - о структурных контекстах), схемы полного разбора предложений.

ЗАКЛЮЧЕНИЕ

Система когнитивной трансферной грамматики, дает возможность строить такие алгоритмические представления, которые не ведут к экспоненциальному росту правил и вычислительных затрат.

Дальнейшие исследования связаны с расширением числа типов трансформаций в англо-русском и русско-английском переводе и построением лингвистических представлений для многоязычной ситуации.

Работа выполнена при поддержке гранта РФФИ 11-06-00476-а «Когнитивно-лингвистические представления и разрешение неоднозначности языковых структур в системах интеллектуальной обработки знаний и машинного перевода».



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 681.3.01

INTELLIGENT SYSTEMS FOR ENTITIES EXTRACTION BASED ON EXTENDED SEMANTIC NETWORKS

Kuznetsov I.P., Charnine M.M., Kozerenko E.B., Matskevich A.G., Nikolayev V.G., Somin N.V.

Institute of Informatics Problems of the Russian Academy of Sciences, Moscow, Russian Federation

igor-kuz@mtu-net.ru

Intelligent systems with linguistic processors extracting the entities and their links from natural language texts are considered. The conceptual model underlying the algorithmic developments is the extended semantic networks (ESN). The paper analyzes the use of the processors for text formalization in various subject fields: economy monitoring, criminal actions, mass media, terrorist activities (in Russian and English). Specific features of the texts are taken into account by linguistic knowledge of the processor: the system can be tuned to various subject areas.

Keywords: entities extraction, intelligent systems, knowledge mining, natural language, semantics.

INTRODUCTION

This work is dedicated to the questions of creating the engineering linguistic models of natural language for construction of linguistic processors for different classes of information systems and to description of the experience of the creation of linguistic ideas in the systems, which relate to the artificial intelligence research field. In the center of our attention are located the intellectual systems, developed on the basis of the apparatus of the extended semantic networks (ESN) [Kuznetsov, 1986], [Kuznetsov et al., 2009a,b]. We call them ESN-systems. These systems were created by the association of developers, including the authors of this article at the Institute of Informatics problems of the Russian Academy of Sciences during the period of two decades within the framework of research projects and applied systems, oriented at the concrete subject areas and customers.

Intellectual ESN- systems contain the developed bases of knowledge, in this case the knowledge is represented in the form of the records in the language of the extended semantic networks, called ESN - structures. Linguistic knowledge is, thus, a special case "of knowledge" and it is also represented in the form of the records in the language of the extended semantic networks. Basic structural element of the ESN is the named N-ary predicate, called "fragment". The whole set of language objects are given in the form of predicate-argument structures, in this case the mechanisms for presentation of embedded structures are supported, which gives very powerful presentation mechanisms for describing the objects of different language levels.

In the process of analysis and synthesis of natural language sentences the formal grammatical apparatus,

similar to the dependency grammar, is used. With this approach the words and the constructions, which perform the role of predicates in the sentence, are the "support" elements, and the result of the analysis of a sentence must become one predicate, which corresponds to the predicate of the sentence (i.e. to basic verb in the tensed form or to another basic predicate expression) in question. Thus, in the process of analysis, in the first place, the processing is performed of the "action words" and the "relation words", i.e., of the verbs and other words, which have syntactic-semantic valences. An example of a "relation word" the word "father", "friend", and the like, i.e., in this case a "relation" is a word which assigns strong clearly expressed syntactical-semantic expectations. Semantic analysis in the engineering linguistic understanding is the process of translation of natural language expressions into "internal" structures of the knowledge base (KB) in our case these "internal" structures are the records in the ESN language. Thus, a KB structure is the code of sense in the intellectual information systems.

1. Basic aspects of semantic modelling

The procedure of conceptual-linguistic simulation on the basis of the ESN apparatus is based on the following principles: • a model must be "open" , i.e., support the effective mechanism of expansion and information update; • the model of the "sense" presentation should consider the facts of extralinguistic reality, which in the form of rules and relations compose a certain basic "world model" and the concrete models of subject areas; • the model should be practical, i.e., not overloaded by the detailed descriptions of connections and relations between the concepts in order to ensure the possibility of its

realization, but at the same time, it should reflect the relevant information for specific objectives.

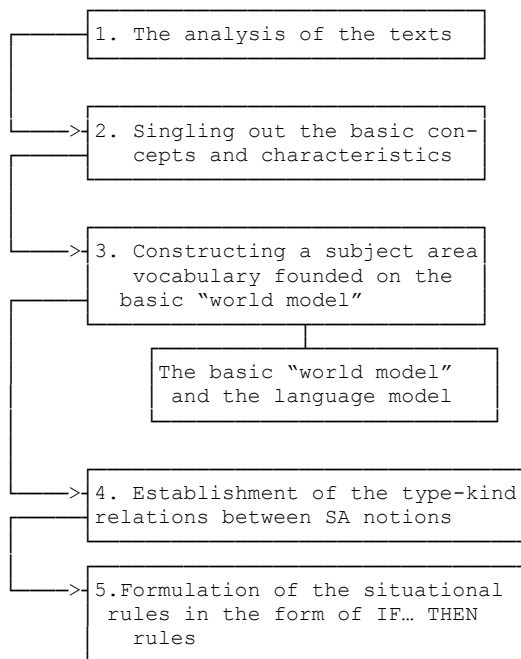


Figure 1 – The flowchart of conceptual linguistic modeling.

Realistic approach to the formulation of the problem dictates the need of limitation to a domain-oriented subset of a natural language. The essence of limitations consists in the following: - first, analyzed text materials contain expert knowledge from particular subject areas (we developed the systems for the subject areas for the diagnostics of the microcircuits production failures, forecast in the social sphere, criminology, and others); - in the second place, for the purposes of the maximally possible elimination of ambiguity, dictionary is built according to the modular principle: there is a certain most general common part (1-2 levels) completed by special dictionaries for each particular subject area. The proposed model of lexical semantics is based on the principle of the "nuclear" value realized in the context of this subject area with the subsequent inductive supplementation of other meanings (if they are actualized in the contexts in question). The taxonomy is also used which is realized in the form of the hierarchical trees of the word classes. The general "world model" of the system serves as the basis for the subject area models. The classes of words, are subdivided into concept/names, relations, actions, properties, characteristics of actions, time and place locatives. The most general notion is "concept", or universal class, which is subdivided into object, the situation, process and others. The words which relate to the classes of actions and relations, are represented as the semantic-syntactic frames, which determine the predicate-argument structures (government model). However, in the described approach (let us name it the ESN-approach) the range of argument values is substantially extended. This extension consists in the fact that in the role of arguments there can appear simple objects corresponding to the individual words, structural objects which present word combinations,

phrases and clauses, and concept of "case" includes not only semantic, but also syntactic aspects. Approach, based on ESN allows to reflect the arbitrary level of the structures embedding it makes it possible to reflect the structural nature of lexical semantics, which in this model has a hierarchical network structure. Linguistic knowledge is represented in the system dictionary and the declarative modules of linguistic processor. In the ESN systems the function of dynamically formed semantic dictionary which is expanded automatically by the system in the course of concrete texts processing is also realized on the basis of initial linguistic information. In Fig. 2 the "internal" description of the verb in the semantic dictionary is represented. This dictionary is automatically generated by the ESN-systems DIES2, LOGOS-D, IKS in the course of natural language texts processing.

```

{(ВЫРАБАТЫВА895_)(DICSEM)
COORD(PROGNOZ1,RUS,ВЫРАБАТЫВА895_,S5
0_31_51_20,%) SUB(UNIV,0+) SUB(UNIV,1+)
SUB(UNIV,2+)
ВЫРАБАТЫВ(0-,1-,2-/3+) INFI(3-)
ПРИДЕТСЯ(3-) ПРИДЕТСЯ(3-/4+) FUT1(4-)
SUB(СРЕД,5+)
  
```

Figure 2 - An example of the presentation of the verb *vyrabatyvat'* - "to manufacture" in the semantic dictionary.

2. The linguistic processor for entities extraction

The linguistic processor (LP) is realized by means of the language DEKL and is controlled by the linguistic knowledge (LK) in the form of object dictionaries, means of parametric tuning, and also the rules of extracting objects and connections [Kuznetsov, Kozerenko 2003]. With the aid of LK the tuning of LP to the appropriate categories of users and text corpora is accomplished. Concrete realization appears as a result. Thus, the paper deals with the means of constructing a class of processors with powerful mechanisms for their tuning and updating. Further development of such processors (LP) is connected with the development of implicit information, which we will consider in a narrow plan, i.e. as the addition of the structures of knowledge by the new information, which is absent or assigned implicitly. In this article the procedure of this development is proposed, which consists in the use of LP for mapping NL texts onto the structures of knowledge (ESN) and the use of the means of logical-analytical processing (productions of the language DEKL) for the creation of new information.

Advantages and deficiencies of the proposed procedure will be examined on specific objectives from the area of "criminology", that is the role functions establishment for the persons (participants) on the basis of the acts performed by them or due to the participation in some specific events. We consider the problem of assignment of properties to the persons (basing on their participation in the acts of different kinds) - "the suffered", "the suspect" and others, if an explicit description of such properties is absent from the

text. For example, if it is said in the text “suffered Ivanov I.I.”, then another task appears, i.e. extraction of some property in the process of linguistic analysis and forming of the corresponding fragments in the knowledge structure. In this article the discussion will deal with LP, customized for the Russian language texts (NL), although the possibilities of LP are wider. There is a sufficient test of tuning LP to the English language texts.

3. The design method

The task of the role functions establishment for the information objects is a special case of the more general task, connected with the estimation of objects according to their descriptions in the NL texts, for example, with the estimation of the stability of enterprise (according to the information from the Internet), by featuring political figures (positive or negative depending on the statements in the press), by the estimation of the role functions quality of product (basing on the statements of users) and so forth. Quite frequently, it is not said directly whether something is bad, or good. As a rule, in NL texts the events are described, the situations, in which one or other information object participated. On the basis of them the estimation is done, which is often represented in the form of a new (generated) property of object.

For the solution of this problem different methods are used [Banko et al., 2007], [Clark et al., 2007], [Gildea, Palmer, 2002], [Pasca, Van Durme, 2007], [Punyakanok et al., 2008], [Cunningham, 2005], [Han, 2004], [Han, Kamber, 2006]. The most common one is the method of the new properties of objects development by using the syntactical-semantic forms. For example:

<what-medicine> caused allergy in <who-human organism> ...,
< what-medicine > has side effects ...
<who-person> made scandal... and so forth.

The application of such forms to the NL texts consists in the search for “estimating” or “characterizing” words (of type “scandal”) or for word combinations of the type “caused allergy” (“it can cause allergy”), it “has side effects” (“side-line actions”), “to make scandal” (“to brawl”)... And then the environment is analyzed, i.e., the words, which stand to the left and to the right, their semantic classes (objects are recognized by them) and case forms. Estimations of information objects as a result are given. By the first two forms the “quality of medicines” is estimated, while by the latter it is recognized that a man performed “hooligan actions” or that he is “suspected”. It is known that in NL many versions are possible for expressing the same idea - with the aid of different syntactic constructions, verbal groups, forms and so forth.

Therefore the number of estimating word combinations will be sufficiently large. Moreover, the application of such forms requires different forms of analysis - morphological (in order to reduce different

word forms to one form), syntactic (the trees are built of the selection of sentences in order to isolate the connected components and to find place for the estimated words) and semantic (in order to extract the objects, which are evaluated). The use of syntactical-semantic forms is connected with certain difficulties caused by special NL features: by the presence in texts of participial, verbal-adverbial constructions, different explanations, facultative components (time, place, purpose), anaphoric references and other language structures. As a result, information objects are frequently disconnected from the estimated words. Hence - the significant losses, which influence the quality of estimation.

Example 1 (*the text is taken from the summaries of incidents of the City Office of Home Affairs, Moscow*):

... Gorelov Peter Sergeevich, 01.03.76 yr/bir, liv: c. Moscow, st. Young Leninists, h.71-6-12, does not work, 01.02.1998 yr. at 4.30 in his house out of hooligan motives in the state of alcoholic intoxication made scandal and broke window glass in the apartment of Litvinova Galina Ivanovna, 20.07.1961 yr/bir, ...

In this example the estimating (characteristic) words are “made scandal” and “broke the window glass”, they are located at a significant distance from the estimated person - “Gorelov Peter Sergeevich”. This limits the possibilities of applying the forms. It is required that the initial extraction of components, which must not be considered in the forms: the years of birth, addresses, specific properties (“he does not work”, “in the state of alcoholic intoxication”), time, place and others, which requires sufficiently deep text analysis with the extraction of objects, their properties and attributes. In connection with the aforesaid, another more promising method is represented - when evaluation is accomplished at the level of knowledge structures. For their construction the objective-oriented LP is used producing the structures of knowledge in which the objects are directly connected with the events and the actions and excluding the above mentioned losses. For the development of implicit information (role functions of objects) the rules of the DEKL language are used which analyze the structures of knowledge (ESN) and form new properties of objects. In this case the structure of knowledge does not change, but it is only supplemented by new (useful) fragments.

4. Presentations of the meaningful portraits of documents

Within the framework of the proposed procedure the development of the role functions of objects (implicit information) is achieved at the level of the structures of knowledge, called the meaningful portraits of documents (SS-documents). Let us examine how such structures appear in the ESN formalism .

Example 2 (translation of the Russian text given below). *Text N22 is taken from the summaries of incidents of the City Office of Home Affairs, Moscow*:

01.02.98 yr. 16-30 to the Home Office applied citizen Mitrofanov Victor Mikhaylovich, 1955 yr. bir.,

liv.: Bohr Highway 38-211, n/w. he stated that 01.02.98 yr. at 10-00 in house 3 at St. Fedosino the unknowns being found in the drunk state made scandal, they expressed themselves by unquotable swearing, they set dog. As a result of what Mitrofanov applied to trauma care center, where the diagnosis was set: the bite of foot.

The objective-oriented LP performs the deep analysis of the text and automatically builds its meaningful portrait (SS- document, transliterated):

DOC_(22, "1-02-98", "SUMMARY; " /0+) 0 (RUS)
 OVD_(OVD/1+)
 FIO(MITROFANOV), VICTOR,
 MIKHAYLOVICH, 1955/2+) UNEMPLOYED (2-/3+) 3- (22, PROP)
 ADR_(Borovskiy, Sh., 38,211/4+)
 PROZH. (it is 2nd, 4)
 ADR_(UL, FEDOSINO, HOUSE, 3/5+)
 FIO (" " , " " , " " , NESKOLKO/6+)
 UNKNOWN (6)
 DRUNK (6-/7+) 7 (2, PROP_)
 SCANDAL (6, PYANYY/8+)
 IS EIGHTH (22, ACT_)
 TO REPORT (IT IS 2ND, 8-/9+) 9 (22, ACT_)
 DATA_(1998,02, ~01, " 10-00" /10+)
 When (9, 10)
 TO TURN (1, GR- N, 2-/11+) 11- (22, ACT_)
 DATA_(1998,02, ~01, " 16-30" /12+)
 When (11-, 12-)
 EXPRESS (6, UNQUOTABLE, [BRAN]/13+) 13- (22, ACT_)
 TO SET (6, [SOBAKA]/14+) 14 (0, ACT_)
 TO TURN (IT IS 2ND, IN, [TRAVMPUNKT]/14+) 14 (0, ACT_)
 TO PLACE (DIAGNOSIS, BITE, [NOGA]/16+) 16 (0, ACT_)
 PREDL_(22,11-, 4, 3-, 9, 13-, 14-/17+) 17- (2,15,341)
 PREDL_(22,15-, 16-/18+) 18- (6,342,448)

A meaningful portrait consists of the elementary fragments, arguments of which are words in the normal form (necessarily for the search and processing). Each elementary fragment has its unique code, which is written in the form of the number with the sign + and is separated by a slash line. For example, in the fragment OVD_(OVD/1+) the sign 1+ is its code (but 1 is the reference to it). Fragments DOK_(22, "1-02-98.TXT", "SUMMARY; " /0+) 0 (RUS) indicate that the meaningful portrait is built on the basis of the Russian-language text of document with number 22 of the file of 1-02-98.TXT", which was processed as the summary of the incidents (linguistic knowledge depend on this). The following fragments present police department OVD_(... /1+), person's surname, name and patronymic FIO (... /2+), person's specific property UNEMPLOYED (2-/3+), address ADR_ (... /4+) and so forth; the signs 2+, it is 2nd, 3+, 3-,... are the codes of the fragments, with the aid of which their connections and relations are assigned. For example, the fragment PROZH (live) (it is 2nd, 4) represents the relation that the person (represented as FIO with code

2+) lives at the address (fragment [ADR]_ with code 4+). Actions are represented in the form of fragments of the type SCANDAL (6, PYANYY/8+) it is 8 (22, ACT_), where it is represented that "person (FIO with code 6+), being drunk, made scandal". With the aid of it is the fragment 8_(22, ACT_) indicates that the first fragment is SCANDAL (.../8+) presents the action and relates to the document with the number 22. A similar role is played by the fragments of the type 3-(22, PROP_), by which the properties are noted. The codes of fragments also serve for the idea of time, scene of action and cases, when one action is included in the composition of another. For example, the fragment TO REPORT (it is 2nd, 8-/9+) represents that the person (code 2+) "reported" (code 9+) about the action (code 8+), i.e., about "made scandal". The following fragments DATA_(... /10+) when (9, 10) represent the time (DATA_), which relates (when) to the action "to report". Special role is played by the fragments PREDL_(...), which correspond to the sentences. They are filled up with the words, which did not enter the information objects (in this example they are absent), or with the codes of objects themselves.

To these fragments the indicators of their position in the text are added. For example, the fragment PREDL_(22,11-, 3-, 9, 13-, 14-/17+) 17- (2,15,341) represents the fact that the objects with codes 11- (corresponding to the action "to turn"), 3- (corresponding to the property "unemployed") and others are located in the sentence, which begins from the 2nd line of the text of the document and they occupy the place from the 15-th to the 310-th byte. These means of positioning are necessary for the work of the reverse linguistic processor (LP).

Analyzing this example, it is possible to make the following conclusions: 1) In SS- document the estimating (characterizing) words occur either in one fragment with the object - SCANDAL (...), or the next one, i.e., the codes of the actions, in which the object participates, are nearby in PREDL_(... 9, 13-, 14...). In this case the possibility of composite actions is considered. 2) On the actions, represented as SCANDAL (...), it is possible to draw the conclusion that the discussion deals with "that suspected", and TO REPORT (,) - that the person is "suffered" or "the applicant". Such conclusions are easily arrived at with the aid of the rules IF... THEN (productions) of the language DEKL, which are the basis for the extraction of role functions. 3) The particular difficulties of dividing the text into the sentences occur (in the old version).

The reduction "of n/r" (with the point at the end) was not understood as the end of a sentence. 4) The linguistic processor (LP) correctly identified the pronoun "he", and also it knew how to reveal the participation of the subject ("unknowns") by the actions "to be evinced by unquotable swearing" and "to set dog", which also characterize subject. At the same time the LP could not connect the action "diagnosis was set" with the person - "Mitrofanov..." (the code is 2-nd). In this case an example proved to be successful. Also the processor LP (with its linguistic knowledge - LK) was

developed for the tasks of the criminal police, connected with different forms of the objective searches: the search for similar participants (addresses, and so forth), search according to the connections, precise search for objects, for the search by signs and other identifiers. In this case the analysis of some complex NL forms was not required, i.e. the cases of the enumeration of the objects participating in the uniform actions (they are described by one verb), the enumeration of the actions of one object and others in contrast to the aforesaid, with the extraction of role functions for each object the indication of its participation in each action is required. Hence it follows that with the use of the proposed procedure the more qualitative extraction of role functions is directly connected with the works on improvement of LP in the aspect of the development of objects and their actions. In many instances the numerous errors caused the inaccuracies in SS-document, e.g.: the absence of punctuation marks or their presence (where it was not required), the inappropriate reductions, gaps in the words and many others. The fact is that the documents, entering the summaries of incidents, are composed on the spot by people (militiamen) of different degree of literacy. Hence – the additional noise and loss. Thus, meaningful portraits are the collections of fragments of ESN which represent the sufficiently high level of formalization of NL texts and are convenient for the working - with the aid of the instrument means - DEKL. Besides LP which analyzes texts and builds SS-documents, there is a reverse linguistic processor (LP) which on the basis of the fragments of the SS-document generates the NL texts presented to the user.

5. The means for the establishment of the role functions

Within the framework of the proposed procedure (instead of the application of syntactical-semantic forms to the documents) the rules are used for logical conclusion and transformation of the knowledge structures - the SS- documents, in which there are no morphological features (of type who, whom,...), and the subjects and the objects are distinguished by their arrangement in the fragments of ESN, which present actions. Syntactical-semantic forms are transformed into the fragments of ESN which determine conversions and logical conclusion achieved by productions of language DEKL. Such fragments play the role of the logical-semantic shell, which determines conversions and logical conclusion on the basis of SS-documents. After filling of the shell by ontological-fragmental knowledge (OFK) which consist of the mentioned fragments (ESN), the program is formed, which accomplishes the development of role functions and completion of the SS-document by the appropriate fragments. With this approach it is possible to avoid many difficulties, connected with the design features of NL and the specific character of the use of syntactical-semantic forms. There are many versions of construction of the shells and representation of the corresponding knowledge which are distinguished

according to the degree of their generality. Let us examine the version which is at present realized and verified.

Case 1. The role functions are determined by the names of actions. In this case for the extraction of objects (participants) which should be assigned properties (role functions), the fragments of the following form are used :

```
INTERPRET (MAN_2, FIO, " suffered")
FORMA_CC (MAN_2, CLASS_D4, " ") CLASS_D4
(TO TURN, TO STATE, TO REPORT, TO PASS
AWAY,...)
```

The first fragment INTERPRET (...) means that from the SS- document it is necessary to extract the fragments of the form FIO (...), that correspond to participants, and to analyze the possibility of assigning them the property "suffered". Such participants are conditionally designated as MAN_2. The second fragment FORMA_CC (...) specifies the conditions for assigning this property to MAN_2, determined by the constant CLASS_D4. In the third fragment CLASS_D3 (...) the words are given which present actions. It is represented that the words belong to the class CLASS_D3. If the participant occurs in one of the enumerated actions, then to this participant the property "suffered" is assigned. This participation is revealed via the analysis of the SS-document. If there is a fragment TO TURN (... , it is n-th,...) in it, the argument of which is the code FIO (... /N+), then the fragment N- ("suffered") is added that represents the role function of the corresponding participant. Conformably for the SS-document represented in example 2 the analysis will occur as follows. Consecutively the extraction of fragments FIO (...) corresponding to the participants is performed. First FIO (MITROFANOV,... /2+) will be extracted . Its code is 2- is the argument of the fragment TO APPLY (1, GR- N, 2-/11+), that presents the action. In connection with this to SS- document the fragment 11- ("suffered") will be added, which via the reverse LP will be transformed into the statement that "Mitrofanov Victor Mikhaylovich is a suffered person". These actions are realized within the framework of the logical-linguistic shell.

Case 2. Role functions are determined by the actions and elucidating words. For this the same fragments are used, as in the first case, but during the enumeration of the names of actions the additional fragments which present actions with the possible elucidating words, are introduced:

```
INTERPRET (MAN_1, FIO, "suspect")
FORMA_CC (MAN_1, CLASS_D3, " ")
FRAUD (USER, POKUPATEL/15+)
TO SET (DOG/16+)
TO BE EXPRESSED (UNQUOTABLE,
SWEARING, OBSCENE,... /17+)
CLASS_D3 (IS DELAYED, TO BE SOUGHT,...,
15,16-, 17-)
```

The given fragments determine actions of the extraction of persons (MAN_1), by which the property of "suspect" is assigned. For this at the level of the knowledge structures their participation is analyzed in the actions "is delayed", "to be sought", and also in the composite actions: "to set dog", "to be expressed

unquotable...”, “*to be expressed by swearing...*” and others. In example 2 the code of fragment FIO (“ ”, “ ”, “ ”, FEW]6+), that represents the unknown persons is the argument of the fragment TO SET (6, DOG/14+), representing action “*to set*” with the elucidating word “*dog*” – “*sobaka*”. Therefore the fragment 6 is added (“*suspect*”), that represents that “*the unknown persons are suspected*”, and through the reverse LP the explanation to this conclusion is offered, see below. A similar conclusion will be made on the basis of the fragment TO BE EXPRESSED (6, UNQUOTABLE, SWEAR/13+), but with other explanations.

Case 3. The actions determine the role functions of several persons. For this (additionally to the fragments INTERPRET) the fragments are added: CLASS_D1 (TO STRIKE, TO BEAT UP,...) FORMA_CC (MAN_1, CLASS_D1, MAN_2), where FORMA_CC (...) indicates the need of the search of two persons - “*suspect*” and “*suffered*” (MAN_1 and MAN_2), that participate in one action, which are mentioned in the fragment CLASS_D1 (...). For example, “*certain person struck another...*”. In the appropriate fragment TO STRIKE (...) the code FIO (...) that corresponds to the first person will stand in front of the second. The given fragments of ESN compose the knowledge OFK which are constantly supplemented - due to the filling of classes by the new words-actions and with the elucidating words. The process of filling is sufficiently simple. If role function is not revealed, then it is necessary to look in the SS- document in which the action of one or another participant (by the text its role is easily determined) occurs. Further, the corresponding constants are located, by which are supplemented the classes of knowledge OFK. Subsequently it is intended to automate the process of completing the knowledge OFK as follows. In the text the words, which determine role functions, are noted. Further, in the formed SS-document the corresponding constants which supplement knowledge OFK are located.

6. Factors of processor quality

The quality of a linguistic processor is determined by a number of factors. First, the possibility for isolation of objects and connections. These are the types of objects being isolated, their quantity. The Semantix processor identifies up to 40 types of objects, including very complex ones, which correspond to actions and events. With an increase in the quantity appear the additional difficulties, connected with collisions of the extraction rules of: some rules can seize the words, which relate to other objects and those extracted by other rules. It becomes important to consider the order of the application of rules, including of the rules of identification. In the second place, an important factor is the selectivity of rules and procedures of the identification: the factor of the noise and losses. By noise we mean the presence of excessive words in the objects. Losses are the situations when an object is not revealed or revealed partially: in the text there are the words, which did not enter into the object. In the Semantix processor the rules are arranged in such a way

that they ensure the high degree of selectivity and the minimization of noise and losses with the large number of the objects being selected.

The third factor is the possibility and the labor expense for tuning to a corpus of texts (for increasing the selectivity of rules for extraction of objects), and also tuning to the new objects. Due to the complexity of analysis this tuning should be achieved through the linguistic knowledge (LK). The latter should have all means for increasing the selectivity of rules and necessary conveniences in the plan of their creation and correction. Ideally, with the aid of LK the tuning to the special features of language as well as to the standard language forms should be ensured. The Semantix linguistic processor ensures the analysis of the Russian and English language forms with the aid of the uniform language model.

The fourth factor is the speed of linguistic processor operation, i.e., the time of text analysis. The speed is determined by the design features of a processor (by means of search time decrease), and also by the number of objects being extracted. The application of rules of extraction is connected with the search for the necessary words, where sortings are required. The greater the number of objects and rules, the greater the time of analysis. In the Semantix processor there are different means of sorting time decrease. Besides the program there are also means of control by linguistic knowledge. It is indicated for each rule, what words should be searched for the initiation of the process of its application. The permissible contexts (to the left and to the right of revealed words) are assigned. These features ensure sufficiently high speed (fractions of a second for 1 KB of text) with a sufficiently large number of objects extracted. More than 40 different types of objects are supported by the Semantix processor. The subject areas represented in the text documents are as follows.

- Documents about terrorism in the Russian language. The analysis of the documents, in which the discussion deals with the terrorist acts and the groups. This feature supports the extraction of 40 types of objects, their connections and the degree of participation in the criminal actions.
- Documents about terrorists in the English language. The objects and links include persons (their family name, name, patronymic – FNP), posts, organizations, terrorist groups, instruments of crime, time and place of events and so forth, and also connection with and participation in the actions.
- Summaries of incidents. Is ensured the extraction of figurants, their connections, organizations, dates, documents, numbers of bank accounts, details of weapons, etc. with the indication of their participation in particular criminal actions.
- Accusatory conclusions, information about the criminal cases. Objects are identified along the entire field of text. Their connections and criminal actions are revealed.
- Government communications, media issues. Persons, dates, organizations, positions and other

significant information and also connections and participation in the actions are selected.

- Autobiographies in the Russian language. From the Russian language resumes all attributes of people, periods of time and place of their work, studies, language proficiency and so forth are extracted.

- Autobiographies in the English. From the English language resumes are all attributes of people, periods of time and place of their work, studies, language proficiency and so forth are extracted.

- Documents of media issues in English. From the English language texts the persons mentioned in media issues, positions, organizations, dates, terrorist and anti-terrorist groups, weapons, events, their time and place, different connections and other features are extracted.

In the processors of the Semantix, Lingua-Master, "Criminal" systems up to 40 types of objects are extracted with high accuracy and minimum noise. For example, the system "Criminal" was verified on about 500 thousand incidents from the summaries of Moscow Criminal Police Department, and on the basic objects showed the unique results: the coefficient of noise, i.e. excessive words in the objects) is not more than 1-2% and losses are not more than 1%. The Semantix Processor was fixed on a smaller quantity of documents dealing with the terrorist activity, and therefore there can be more noise and losses in it. But this can be quickly fixed. The fact is that to consider everything which can be encountered in the NL texts is impossible. Therefore, in the first place, the representative collections of test documents are extremely important, and in the second place, the means of fixing or tuning of linguistic processors are as follows: the employment of hybrid approaches comprising hand-made rules and statistical means for rapid correction and fine adjustment of linguistic knowledge.

In our systems there is an entire complex of such means which ensure rapid tuning to the applications (including the introduction of new objects and connections) taking into account the demands of customers. A sufficiently in-depth analysis of sentences is conducted with the development of verbal forms, and also with the identification of objects of the entire text. The analysis of the complex language structures is ensured: forms with verbal nouns, participial and adverbial constructions, coordinated terms, etc. is supported by the expert component. The Semantix processor can be used as a stand-alone (independent) module. At present the first release of the English language version of the object - oriented linguistic processor Semantix has been developed.

CONCLUSION

The objective-oriented linguistic processor for entities extraction can be used in different areas of application. It has a number of essential advantages, the main is that it outperforms the recently appeared systems by the number of the supported object types, there is an entire complex of logical and statistical means which ensure rapid tuning to the applications

(including the introduction of new objects and connections) taking into account the demands of customers. Our further efforts are connected with the developing a simultaneous bilingual search and extraction features.

The procedure of the role functions extraction centered at the analysis of knowledge structures is sufficiently promising from the point of view of the knowledge bases technology development. The current task is to improve its performance.

The methods were tested on the basis of the summaries of incidents which contain about three thousand documents (each document consists of 10 - 80 lines). In the case of the summaries processing the documents with the mentioned enumerations (there were about 10% of them) were withdrawn and in the remained texts the gaps in the words were removed. At the current moment the program which realizes the proposed procedure gave about 80% of correct recognition of role functions, and about 65% of complete explanations with the indication of all acts. But these numbers rapidly change for the better due to the means (the LK and OFK knowledge) of tuning the LP to special features of the subject area texts. For this not much time is required. Let us note that tuning itself to the extraction of the role functions of persons from the mentioned summaries (with reaching the indicated percentages), required about two weeks of the work of one person. The development and fixing of the shell itself took about four days. The subsequent development is connected with the improvement and the tuning of LP to the work with complex NL forms. At present the extraction of actions is interfered with causal word combinations of the type "out of the hooligan motives", "owing to the hostile relations" and so forth, which at present are introduced into the system. Difficulties appear with the transfer of the subject of action to other actions to which the subject is not assigned explicitly, but its presence is implied.

The second direction of research and development is connected with the extension of the shell features to the solution of other problems connected with the estimation of objects depending on the nature of statements about them in the texts of description. Within the framework of the studies conducted it is also intended to tune the shell to the work with the English language texts. Since the meaningful portraits of the English language and Russian language texts have the identical structure (SS-documents), this tuning cannot be labor-consuming.

Bibliography List

- [Kuznetsov, 1986] Kuznetsov, I.P. Semanticheskie Predstavleniia. Moscow: Nauka, 1986, 290 p.
- [Kuznetsov et al., 2009] Kuznetsov I.P., Efimov D.A., Kozerenko E.B. Tools for Tuning the Semantix Processor to Application Areas // Proceedings of ICAI'09, Vol. I. WORLDCOMP'09, July 13-16, 2009, Las Vegas, Nevada, USA. - CRSEA Press, USA, 2009. P. 467-472.
- [Kuznetsov et al., 2009] Kuznetsov I.P., Kozerenko E.B., Kuznetsov K.I., Timonina N.O. Intelligent System for Entities Extraction (ISEE) from Natural Language Texts // Proceedings of the International Workshop on Conceptual Structures for Extracting Natural Language Semantics - Sense'09, Uta Priss, Galia Angelova

(Eds.), at the 17 International Conference on Conceptual Structures (ICCS'09), University Higher School of Economics, Moscow, Russia, 2009. P. 17-25.

[Kuznetsov, Kozerenko, 2003] Kuznetsov, I.P., Kozerenko T.B. The system for extracting semantic information from natural language texts // Proceeding of International Conference on Machine Learning. MLMTA-03, Las Vegas US, 23-26 June 2003 г., p. 75-80.

[Banko et al., 2007] Banko M., M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open Information Extraction from the Web // Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07), 2007. P. 2670–2676.

[Clark et al., 2007] Clark P., P. Harrison, and J. Thompson. A Knowledge-Driven Approach to Text Meaning Processing // Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning, 2007. P. 1–6.

[Gildea, Palmer, 2002] Gildea D. and M. Palmer. The necessity of syntactic parsing for predicate argument recognition. In Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02), Philadelphia, PA, 2002. P. 239–246.

[Pasca, Van Durme, 2007] Pasca M. and B. Van Durme. What You Seek is What You Get: Extraction of Class Attributes from Query Logs // Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07), 2007. P. 2832–2837.

[Punyakank et al., 2008] Punyakank V., D. Roth, and W. tau Yih. The Importance of Syntactic Parsing and Inference in Semantic Role Labeling // Computational Linguistics 34(2), 2008. P. 257–287.

[Cunningham, 2005] Cunningham H. Automatic Information Extraction // Encyclopedia of Language and Linguistics, 2nd ed. Elsevier, 2005.

[Han, 2004] Han J., Pei Y. Yin, and Mao R. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach, // Data Mining and Knowledge Discovery, 8(1), 2004. P. 53–87.

[Han, Kamber, 2006] Han J. and Kamber, M. Data Mining: Concepts and Techniques // Morgan

этой основе структуры знаний. Одно из направлений развития таких процессоров связано с выявлением имплицитной информации, которая рассматривается в узком плане - как выявление новых свойств объектов, заданных в неявном виде. Предлагается методика такого выявления, основанная на анализе структур знаний. В качестве примера рассматривается выявление ролевых функций фигурантов на базе их описаний в сводках происшествий.

ОСНОВНАЯ ЧАСТЬ

Научное направление, связанное с обработкой произвольных текстов ЕЯ (в заданной предметной области) развивается с учетом задач определенной категории пользователей: большинство пользователей – специалистов в своей области - интересуются лишь конкретными вещами для своих задач. Например, следователям важны фигуранты, их места жительства, телефоны, приметы, действия лиц (с указанием места, времени), связи и др. Такая информация является основой для оперативно-розыскных действий, поиска по связям, различных видов анализа. Специалистов по кадрам интересуют организации, где человек работал, кем и когда это было. Подобную информацию будем называть *сущностями, информационными объектами*, которые различаются по типам. Например, лица и фигуранты – это объекты одного типа, адреса – другого и т.п.

ЗАКЛЮЧЕНИЕ

Анализ методики проводился с использованием сводки происшествий, содержащей около трех тысяч документов (каждый документ содержит от 10 до 80 строк). Предлагаемая методика, основанная на анализе структур знаний, представляется перспективной для развития технологии баз знаний.

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ИЗВЛЕЧЕНИЯ СУЩНОСТЕЙ, ОСНОВАННЫЕ НА РАСШИРЕННЫХ СЕМАНТИЧЕСКИХ СЕТЯХ

Кузнецов И.П., Шарнин М.М., Козеренко
Е.Б., Мацкевич А.Г., Николаев В.Г.,
Сомин Н.В.

*Учреждение Российской академии наук
Институт проблем информатики РАН,
Москва, Российская Федерация*

igor-kuz@mtu-net.ru

В данной работе приведены модель, механизмы и реализации интеллектуальных систем извлечения знаний - сущностей и их связей - из текстов на естественном языке. В основе проектных решений лежит аппарат расширенных семантических сетей. Инструментом создания систем служит оригинальный язык логического программирования – ДЕKL.

ВВЕДЕНИЕ

Дается описание семантико-ориентированного лингвистического процессора, извлекающего из текстов естественного языка информационные объекты, их свойства и связи и формирующего на



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОНТОЛОГИЯ АВТОРИТЕТА: МЕХАНИЗМЫ АПЕЛЛЯЦИИ К ЧУЖИМ МНЕНИЯМ В ТЕКСТАХ РАЗЛИЧНЫХ ЖАНРОВ

Ефименко И.В.

*Национальный исследовательский университет
«Высшая школа экономики»,
факультет филологии
г. Москва, Россия
iefimenko@hse.ru*

В статье обсуждается онтологический подход к исследованию механизмов апелляции к экспертным мнениям, транслируемым в документах различных жанров. Рассматриваются лингвистические и экстралингвистические аспекты, подлежащие учету при проектировании онтологий в соответствующей предметной области.

Ключевые слова: экспертное мнение, экспертное знание, онтологический инжиниринг, анализ дискурса, лингвистический контекст, экстралингвистический контекст.

ВВЕДЕНИЕ

Настоящая работа посвящена онтологическому подходу к исследованию механизмов апелляции к авторитетному – т.е. экспертному – мнению в документах различных жанров.

Проблематике анализа экспертных мнений посвящен ряд работ в рамках различных дисциплин. Несмотря на множество существенных результатов, имеющиеся работы и проекты обладают рядом ограничений, главным образом, в силу междисциплинарного характера исследуемой проблемы, о чем будет сказано ниже. В частности, недостаточное внимание уделяется лингвистическим и онтологическим аспектам трансляции и использования экспертного знания.

При этом актуальность тематики не вызывает сомнения: обращение к экспертным знаниям, экспертиза проектов, заявок и т.п., обоснование некоторой позиции через апелляцию к экспертному мнению (например, в СМИ) являются в настоящее время важнейшими элементами многих процедур и сценариев ситуаций, лежащих в общественной, научно-технической, финансовой и других плоскостях.

Базисом для настоящей работы послужили исследования Е. Н. Пенской ([Пенская, 2010], Пенская, 2011)), где, в частности, рассматриваются экспертные ситуации в науке и повседневности, определяется понятие экспертной деятельности,

специфицируются ее отличия от исследовательской деятельности, изучаются особенности самоидентификации экспертов и разрабатывается тезаурус экспертности.

1. Исследование механизмов трансляция экспертного мнения: онтологический подход

1.1. Междисциплинарный характер исследуемой проблемы

При исследовании механизмов апелляции к авторитетным мнениям речь идет, с одной стороны, о формализации таких комплексных явлений, как эксперт, экспертиза, экспертные знания, с другой стороны, об анализе СМИ и других типов информационных источников. Поэтому одной из важнейших задач является выбор подходов, методов и средств для моделирования сложных предметных областей и анализа контента.

Исследования, связанные с анализом различных видов контента (СМИ и других информационных источников, в т.ч., для новых видов СМИ и Интернет-ресурсов – блоги, форумы и т.п.), ведутся в настоящий момент в рамках целого ряда дисциплин и научных направлений, таких как филология, искусственный интеллект, журналистика, социология. В качестве более частных направлений и основных сфер прикладных разработок можно назвать различные области математической лингвистики, анализ дискурса,

контент-анализ.

В сфере экспертного знания следует отметить ряд работ и проектов, появившихся в последние годы как в России, так и за рубежом, и посвященных, в частности, анализу возникновения и развития «фабрик мысли», think tanks ([Иванченко и др., 2006], [Куренной, 2006], [Беляева и др., 2008], [Ашкерова, 2009]).

В Европе проекты, связанные с контент-анализом и проблематикой формализации экспертного знания, ведутся в ряде университетов, а также в рамках Седьмой рамочной программы ЕС (7th Framework Program, FP7) и некоторых других европейских программ.

Существенным ограничением представленных выше работ и проектов является следующее. Результаты, полученные в части анализа экспертного сообщества, фабрик мысли, методов работы с экспертами и экспертными знаниями, лежат, в основном, в области социологии или политологии и часто представлены в слабо формализованном виде. Математические работы, посвященные, например, проблематике согласования мнений, не исследуют саму природу экспертного знания и механизмы его трансляции как в профессиональных, так и в непрофильных источниках. В свою очередь, в области анализа контента можно выделить два основных, почти не интегрированных с друг другом, направления: «гуманитарное», в рамках которого используются методики и инструменты, характерные для филологической науки, литературоведения, журналистики, и «точное», где контент рассматривается с точки зрения методов работы с информацией, математики и искусственного интеллекта. Недостатком первого направления является слабая формализация, второго – отсутствие учета, в ряде случаев, всех аспектов (социальных, языковых и др.) таких сложных явлений, как общественное сознание, отражение в информационных источниках социальных ситуаций и событий, имеющих комплексный характер, формирование авторами публикаций информационных потоков, в т.ч., направленное, т.е. в интересах различных групп, сочетание субъективного и объективного в прессе, взаимодействие экспертного знания и общественного мнения и т.п.

Представляется, что использование онтологического подхода, позволяющего, с одной стороны, учитывать как лингвистические, так и экстралингвистические (предметные) особенности моделируемых областей, в другой стороны, достичь высокого уровня формализации, поможет преодолеть представленные выше ограничения.

1.2. Подходы, методы и средства

В качестве одного из базисных подходов в работе используется методология онтологического инжиниринга, впервые предложенная в рамках

проекта TOVE (TOronto Virtual Enterprise) и известная также как методология Грюнингера и Фокса (Grüninger-Fox's methodology) [Grüninger, Fox, 1994], [Гурьянова М. А. и др., 2011]. Основополагающим понятием данной методологии является понятие вопросов оценки компетенции (или компетентности), применяемое, прежде всего, к самой онтологии. Вопросы оценки компетентности создаваемой онтологии основаны на мотивационных сценариях (определяемых целями и задачами создания онтологии) и рассматриваются как требования к изобразительной мощности онтологии и возможности с ее помощью решать задачи, обозначенные в мотивационных сценариях. По сути, вопросы компетенции являются тем множеством критериев (benchmark), которому должна удовлетворять модель, чтобы быть пригодной для решения поставленных задач. При этом должны быть найдены все правильные решения, и только они. Такой подход позволяет управлять разработкой новых моделей и проверять функциональные возможности существующих моделей. С учетом вышесказанного, очевидно, что понятие вопросов компетенции может применяться не только к модели предметной области в целом, но и к ее элементам. Так, например, в качестве самодостаточной модели, взаимосвязанной при этом с другими онтологиями в рамках системы онтологий, может рассматриваться модель компетенции отдельного эксперта, как независимо от ситуации трансляции экспертного знания, так и в ее контексте.

В качестве основной среды разработки и онтологического редактора выбрана система Protégé. Такой выбор обусловлен, в частности, необходимостью создания системы взаимосвязанных онтологий, а также использования аксиом и правил вывода.

Фрагмент предметной области (главным образом, в части типов объектов), визуализированный в системе Protégé, представлен на рисунке 1. Проблемам проектирования онтологии в части лингвистических аспектов, цель создания которой – учесть языковые особенности апелляции к экспертным мнениям в текстах различных жанров, посвящен следующий раздел.

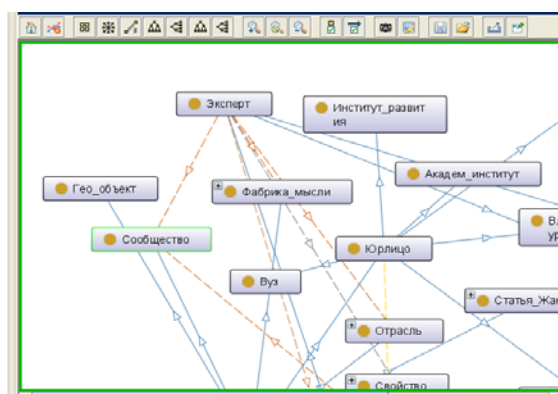


Рисунок 1 – Фрагмент предметной области «Экспертные знания»

1.3. Область применения результатов

Система предметных и лингвистических онтологий в области механизмов апелляции к экспертному знанию является основой для проведения аналитических исследований, результаты которых применимы в сфере гуманитарных технологий, экспертной журналистики, анализа дискурса, прогнозирования.

Онтологии в предметной области экспертного знания могут также стать основой для разработки систем, автоматизирующих деятельность в области экспертизы. При этом одним из важнейших аспектов является возможность внедрения автоматизированных механизмов проверки consistency экспертных оценок (как для нескольких экспертов, так и для одного специалиста с учетом динамики возможного изменения экспертного мнения во времени).

2. Лингвистические аспекты апелляции к экспертным мнениям в текстах различных жанров

2.1. Типы информационных источников и жанры текстов

Источники информации, в рамках которых наиболее частотна апелляция к чужому мнению, т.е. традиционно используемые как площадки выступления экспертов и/или инструменты трансляции экспертных знаний, целесообразно разделить на два основных типа:

- Профильные источники, т.е. по самой своей природе предназначенные для передачи экспертного и/или научного и научно-технического знания (родственного по своей природе экспертному), в т.ч. научные статьи, патенты, профессиональные СМИ, профессиональные блоги и форумы и т.п.;
- Непрофильные источники, т.е. источники, основной целью функционирования которых не является передача экспертных и научных знаний, но в которых на периодической основе или спорадически появляются материалы, содержащие прямые или косвенные отсылки к экспертным мнениям. Такого рода источники играют значительную роль в области воздействия на общественное мнение, поскольку именно они предназначены для широкого круга читателей. Прежде всего, к таким источникам следует отнести СМИ, основным профилем которых не является научно-исследовательская, разработческая или экспертная деятельность.

Промежуточное положение занимают многие источники среды Интернет.

Используемое выше понятие экспертного знания применимо, в большей степени, для профильных источников информации. В случае непрофильных источников речь идет, скорее, об экспертном мнении (или о трансляции экспертного знания журналистом-непрофессионалом в предметной

области эксперта или третьими лицами).

Указанное разделение актуально для задач организации автоматизированного мониторинга различных типов источников и проведения исследований в области «глубокой аналитики». С точки зрения непосредственно лингвистического анализа большой интерес представляет понятие жанра документа, а также, в ряде случаев, предметная область экспертного знания.

2.2. Ключевые противопоставления, подлежащие учету в лингвистической онтологии

Анализ корпусов текстов различных жанров, где имеет место прямая или опосредованная апелляция к чужому мнению, которое может рассматриваться как экспертное, показывает, что при проектировании лингвистических аспектов предметной области трансляции экспертного знания должны быть учтены основные противопоставления, представлены ниже. Такого рода противопоставления определяют особенности лингвистических механизмов трансляции экспертного знания, сведения о которых, в свою очередь, могут служить дополнительным источником информации при исследовании вопросов точности передачи экспертного мнения, объективности vs. ангажированности автора, транслирующего экспертное знание, и т.п. Аспекты, подлежащие учету, могут быть объединены в группы, которые должны стать основой концептов онтологии, примеры представлены ниже:

- Лингвистические особенности, связанные с характеристиками источника:
 - экспертное мнение в российском vs. зарубежном источнике (наиболее релевантно для СМИ);
 - жанр текста;
 - предметная область документа;
 - предметная область экспертного мнения/знания (а также, более широко, принадлежность предметной области к естественнонаучному, точному, гуманитарному знанию), тематика, научная дисциплина (для профильных источников). Может совпадать или не совпадать с предметной областью документа;
- Лингвистические особенности, связанные с характеристиками субъекта, мнение которого транслируется:
 - трансляция мнения иностранного vs. отечественного эксперта;
 - мнение с указанием авторства vs. анонимное;
 - для мнений без указания авторства – с указанием принадлежности к юридическому лицу или определенному сообществу vs. без такового;
 - наличие эксплицитного указания на анонимность эксперта;
 - частное vs. коллективное мнение;

- Лингвистические особенности контекста (лингвистического и экстралингвистического):
 - позитивный vs. негативный контекст, в т.ч., подтверждение или отрицание некоторой точки зрения;
 - наличие vs. отсутствие указания лингвистического контекста, в рамках которого было высказано то или иное экспертное мнение;
 - наличие vs. отсутствие указания экстралингвистического контекста, в рамках которого было высказано то или иное экспертное мнение;
 - экспертное мнение об экспертном мнении: согласие или не согласие одного эксперта с другим, цитирование мнения одного эксперта в рамках мнения другого эксперта (в свою очередь, цитируемого автором документа);
 - «история вопроса»: сравнение мнений одного и того же или различных экспертов, высказанных в различное время;
 - наличие апелляции к экспертному мнению, высказанному во внешних по отношению к исследуемому источниках информации, в т.ч., в средствах Web 2.0 (с учетом лингвистических особенностей средств Web 2.0);
 - наличие эксплицитного указания целей и задач обращения к экспертному мнению;
 - апелляция к мнению одного vs. нескольких экспертов (с указанием мнения каждого – для обоснования одной и той же или различных позиций);
 - наличие в рамках документа цитирования противоборствующих мнений;
 - сопоставление мнения эксперта и общественного мнения;
- Лингвистические особенности передачи экспертного мнения как такового:
 - прямая vs. не прямая речь (или пересказ);
 - используемые при трансляции экспертного знания глаголы речи и другие типы речевых оборотов;
 - использование лексики и других средств, вводящих имплицитный компонент, в рамках не прямой речи;
 - использование лексики и других средств, вводящих имплицитный компонент, в рамках прямой речи;
 - соотношение позитивного, негативного и нейтрального в высказываниях экспертов;
 - введение фрагмента прямой речи (точной цитаты) во фрагменты, являющиеся в целом не прямой речью;
- Целевая установка:
 - наличие рекомендации в высказывании эксперта;
 - наличие оценочного компонента в высказывании эксперта;
 - наличие количественных данных в высказывании эксперта;

- наличие прогнозной составляющей в высказывании эксперта.

ЗАКЛЮЧЕНИЕ

В настоящей работе представлен поход к исследованию экспертных мнений, транслируемых в текстах различных жанров, в рамках парадигмы онтологического инжиниринга. Представлены основные лингвистические аспекты, подлежащие учету при проектировании онтологий в предметной области экспертных знаний и мнений. Областью применения указанного подхода и полученных в рамках такого подхода результатов являются гуманитарные технологии, экспертная журналистика, анализ дискурса, прогнозирование, экспертная деятельность.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Ашкерев, 2009] Ашкерев А. Тезисы об экспертократии, М., 2009
- [Беляева и др., 2008] Беляева Н. А., Зайцев Д. Т. Фабрики мысли и центры публичной политики. Политика, 2008
- [Гурьянова М. А. и др., 2011] Гурьянова М. А., Ефименко И. В., Хорошевский В. Ф. Онтологическое моделирование экономики предприятий и отраслей современной России. Часть 2. Мировые исследования и разработки: аналитический обзор. Препринт. М., НИУ ВШЭ, 2011
- [Иванченко и др., 2006] Экспертиза в современном мире: от знания к деятельности/ Под ред. Г.В. Иванченко, Д.А. Леонтьева. – М.: Смысл, 2006
- [Куренной, 2006] Куренной В. Мыслящая Россия, М. 2006
- [Пенская, 2008] Эксперт и/или интеллектуал. Русский журнал, рабочие тетради, 2008.
- [Пенская, 2010] Эксперт, экспертиза экспертность, Материалы 11 Международной конференции НИУ ВШЭ, 2010
- [Пенская, 2011] Новые медиа и их экспертная составляющая. Материалы 11 Международной конференции НИУ ВШЭ, 2011
- [Gruninger, Fox, 1994] Gruninger, M., Fox, M.S., "The role of competency questions in enterprise engineering", Proceedings of the IFIP WG5.7 workshop on benchmarking - theory and practice, Trondheim, Norway, Jun, 1994

ONTOLOGY OF AUTHORITY: APPEALING TO EXTERNAL OPINIONS IN TEXTS OF DIFFERENT GENRES

Efimenko I.V.

*National Research University
Higher School of Economics,
Moscow, Russia
iefimenko@hse.ru*

Ontological approach to analyzing mechanisms of appeal to expert opinions which are transmitted through the sources of different genres is being presented in the article. Linguistic and extralinguistic aspects relevant to modeling an ontology of the expert knowledge domain are discussed.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

АВТОМАТИЧЕСКОЕ ФОРМИРОВАНИЕ НЕОДНОРОДНОЙ СЕМАНТИЧЕСКОЙ СЕТИ НА ОСНОВЕ ВЫЯВЛЕНИЯ КЛЮЧЕВЫХ ПРЕДИКАТНЫХ СТРУКТУР ПРЕДЛОЖЕНИЙ ТЕКСТА

Харламов А.А. *, Ермоленко Т.В. **

* *Институт высшей нервной деятельности и нейрофизиологии РАН,
г. Москва, Россия*

kharlamov@analyst.ru

** *Институт проблем искусственного интеллекта, г. Донецк, Украина*

etv@iai.dn.ua

В докладе представлен новый подход к автоматическому выявлению семантических отношений в тексте. Подход использует ту же методику формирования семантической сети, которая используется в ранее разработанной одним из авторов технологии автоматического смыслового анализа текстов TextAnalyst, только в качестве исходного материала к этому анализу вместо обычного текста используется текст с выявленными предикатными структурами предложений. Разметка осуществляется автоматически на основе правил согласования слов в синтаксических группах. В процессе анализа, также как и в случае однородной (ассоциативной) семантической сети, формируется семантическая сеть, ключевые понятия которой ранжированы по их смысловой значимости в тексте, но наряду с ассоциативными отношениями между понятиями сети используются некоторые общепринятые семантические отношения, в результате чего вместо однородной (ассоциативной) сети формируется неоднородная семантическая сеть. Эта неоднородная семантическая сеть является объединением точно выявленной (в виде предикатных структур) семантики отдельных предложений с обобщенной семантикой целого текста, представленной семантической сетью текста. **Ключевые слова:** автоматическое выявление предикатных структур, автоматический смысловый анализ текста, автоматическое формирование неоднородной семантической сети, синтаксические правила.

ВВЕДЕНИЕ

Основное содержание текста может быть выражено разными способами: перечислением ключевых предикатных структур, содержащихся в предложениях текста, выявлением ключевых понятий текста в их взаимосвязях с последующим построением семантической сети. В первом случае предикатные структуры раскрывают содержание отдельных предложений текста. Причем сумма их смыслов не исчерпывает содержания текста [Ягунова, 2008]. Во втором случае семантическая сеть текста теряет подробности, содержащиеся в смысле отдельных предложений. Объединение двух подходов, заключающееся во включении подробной семантической структуры предложений в общую семантическую сеть текста позволит сохранить детальность описания семантики отдельных предложений с обобщенными семантическими представлениями целого текста. Такое объединенное представление совмещает детальность описания семантики текста с помощью

неоднородной семантической сети с автоматизмом формирования этой сети, что существенно отличает это от других подобных представлений, в которых неоднородная семантическая сеть строится исключительно вручную.

Для выявления предикатных структур необходимо сначала определить, какие понятия текста (слова и устойчивые словосочетания) являются ключевыми понятиями в тексте, а затем определить, какие предикатные структуры, их содержащие, являются наиболее важными в этом тексте.

Выявление ключевых понятий текста не является на настоящий момент очень сложной задачей. Существует множество подходов к их определению. Есть подходы, основанные на заранее сформированных семантических сетях предметных областей как в Excalibur [convera.com, 2011]. В других случаях в процессе поступления текстов формируется семантическая сеть конкретного текста как в TextAnalyst [Kharlamov, 2004; Харламов, 2001]. Однако выявление ключевых

понятий, даже в их взаимосвязях в тексте, оказывается недостаточным для описания смысла текста. Чаще всего, этого описания хватает лишь для классификации текста (отнесения текста к одной из предметных областей).

Поскольку предикатная структура характеризует отношения понятий в предложении, последовательность ключевых предикатных структур предложений текста исчерпывает содержание текста. Таким образом, выявление ключевых предикатных структур позволяет эксплицировать основное содержание текста. Причем ранжирование предикатных структур в тексте по их смысловому весу позволяет формировать более или менее компактные выжимки смысла текста.

Выявление ключевых предикатных структур в тексте возможно с использованием того же подхода, который был использован для выявления ключевых понятий в тексте в технологии TextAnalyst и зарекомендовал себя достаточно хорошо [Gartner Symposium, 2001]. Отличие в использовании технологии в случае выявления ключевых понятий, и в случае выявления ключевых предикатных структур заключается в первичной обработке текста. Во втором случае специфическая первичная обработка позволяет выявлять ключевые предикатные структуры вместо ключевых слов. Первичная обработка для выявления предикатных структур в русскоязычных текстах может использовать, например, согласование в синтаксических группах и формальные признаки для выявления типов отношений [Сокирко, 2001], например как в словаре валентностей глаголов.

После выявления предикатных структур и построения их иерархий в предложениях становится возможным построить первичную семантическую (уже неоднородную, размеченную типами связей) сеть также как ассоциативную сеть в технологии TextAnalyst. В этом случае звездочки ключевых понятий с их ближайшими ассоциантами как их (ключевых понятий) семантическими признаками заменяются на предикатные структуры. Таким образом, ассоциативные отношения заменяются на неоднородные отношения актантов предикатива. Перенормировка весов ключевых понятий (как вершин) уже этой сети затем позволит получить семантическую сеть текста, и, тем самым, ранжировать предикатные структуры по их смысловому весу в тексте.

1. Отображение в многомерное пространство как способ формирования поуровневых словарей событий разной частоты встречаемости

Обработка внутренне структурированной информации различных модальностей (в том числе текстовой), имеющей многоуровневую структуру, с помощью нейронных сетей на основе нейроподобных элементов с временной суммацией

сигналов [Харламов, 2001] позволяет автоматически сформировать словари событий разной частоты встречаемости (словари разных уровней иерархии). Такая обработка сводится к отображению F , подвергнутой предварительно первичной обработке информационной последовательности A , в многомерное пространство R^n , в результате которого информационная последовательность преобразуется в связанную последовательность точек многомерного пространства – траекторию \hat{A} :

$$\hat{A} = F(A). \quad (1)$$

Это отображение обладает свойством ассоциативности обращения к траектории: как только в информационной последовательности появляется повторяющийся фрагмент, траектория возвращается к ранее пройденному участку. Запоминание числа прохождений траекторией точек многомерного пространства, с последующим применением порогового преобразования, позволяет выявлять фрагменты траектории заданной частоты появления, которые составляют словари событий входной информации заданной частоты встречаемости $\{\hat{B}_i\}$. Для лингвистической информации это, например, словари флективных морфем, корневых основ, синтаксических групп. Выявленные таким образом 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 [Харламов, 2008], словарь синтаксических групп – для формирования неоднородной семантической сети.

Ранее одним из авторов была реализована технология обработки текстовой информации TextAnalyst [Sullivan, 2001], позволяющая автоматически выявлять ключевые понятия в тексте на основе только информации о структуре самого текста (независимо от предметной области). Для этого формировался 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 (предложениях). Частотный портрет, таким образом, содержал информацию о частоте встречаемости понятий и их попарной (в терминах их ассоциативной связи) встречаемости в тексте. Использование хопфилдоподобного алгоритма [Hopfield, 1982] позволяло перейти от частоты встречаемости к смысловому весу (вес связей при этом оставался неизменным).

Этот способ включал несколько этапов. На этапе первичной обработки из текста удалялась нетекстовая информация, текст сегментировался на

слова и предложения, из текста удалялись стоп-слова, рабочие и общеупотребимые слова, а оставшиеся слова подвергались морфологической обработке. Для простоты анализа морфологическая обработка производилась с использованием традиционного морфологического словаря – словаря первого уровня – $\{\hat{B}_i\}_1$. Далее формировался словарь второго уровня – $\{\hat{B}_i\}_2$ – словарь корневых основ (и устойчивых словосочетаний). На следующем этапе строился частотный портрет текста, то есть выявлялись 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 формировался словарь третьего уровня $\{\hat{B}_i\}_3$). И, наконец, на третьем этапе, частоты встречаемости перенормировывались в смысловые веса с использованием итеративной процедуры, похожей на алгоритм искусственной нейронной сети, предложенной Хопфилдом:

$$w_i(t+1) = \left(\sum_{i \neq j} w_i(t) w_{ij} \right) \sigma(\bar{E}), \quad (2)$$

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 получали 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):

$$N \cong \{ \langle c_i c_j \rangle \}, \quad (3)$$

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 переопределена как множество так называемых звездочек $z_i = \langle c_i \langle c_j \rangle \rangle$:

$$N \cong \{ z_i \} = \{ \langle c_i \langle c_j \rangle \rangle \}, \quad (4)$$

Под звездочкой $z_i = \langle c_i \langle c_j \rangle \rangle$ 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.

2. Синтаксический анализ как средство первичной обработки для последующего семантического анализа

В процессе синтаксического анализа из предложений текста выделяются содержащиеся в них предикатные структуры, а также отношения сочинения, и атрибутивные отношения. 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.

- Поиск шаблона, соответствующего одной из минимальных структурных схем предложений (МСС), описывающих предикативный минимум предложения [Белошарова и др., 1997].

Множество простых предложений русского языка задается перечнем МСС, описывающих предикативный минимум предложения. МСС — это модель, отвлеченный образец, отражающий способ выражения предикативности.

Использование МСС в качестве формального образца позволяет получить предикативную основу (структурную схему) простого предложения, и в дальнейшем — его предикатную структуру [Харламов и др., 2011]. Это первый и обязательный шаг для проведения первичного семантического анализа в формировании информационного

портрета текста, поскольку 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.

Семантико-синтаксический анализ предложения предусматривает создание электронного словаря валентности глаголов. При этом для каждого из глаголов (около 20 тысяч в русском языке) необходимо указать, какими падежами и с какими предлогами он может управлять, а также — в каких семантических ролях (семантических падежах) выступают актанты глагола.

Предикатная структура предложения по сути дела является звездочкой. Эти звездочки объединяются в сеть. Объединение иерархично: к главному понятию первой звездочки (первого предложения) присоединяются звездочки других предложений, в которых главным понятием является субъект предикатной структуры, они объединяются с главным понятием первой звездочки ассоциативными связями. К их ассоциантам как главным понятиям уже присоединяемых звездочек присоединяются с помощью соответствующих связей (предикатных, атрибутивных, ассоциативных) второстепенные понятия этих звездочек, и так до включения в сеть всех понятий. Затем сеть перенормируется как и в разделе 2. Используя (4), предикатную структуру, описанную с помощью (1), можно представить в виде звездочки

$$\langle Pred \langle Obj_i \rangle \rangle, i = 0 \dots n, \quad (5)$$

где *Pred* является главным понятием, *Obj_i* — понятиями-ассоциантами (*Obj₀* — активный субъект *Subj*). Поскольку глагольная конструкция считается несущей семантической нагрузки в ассоциативной семантической сети, место главного понятия в звездочке занимает субъект.

Затем, эти звездочки объединяются в сеть. Объединение иерархично: к ассоциантам субъекта предикатной структуры (главного понятия) первой звездочки (первого предложения) присоединяются звездочки других предложений, в которых главные понятия (субъекты их предикатных структуры) совпадают с ассоциантами первой звездочки. К ассоциантам главного понятия первой звездочки также присоединяются с помощью соответствующих связей (атрибутивных,

ассоциативных) второстепенные понятия первой звездочки (соответствующие актанты ее предикатной структуры). Этот процесс продолжается до включения в сеть всех понятий. Затем сеть перенормируется как в разделе 1.

3. Семантический анализ текста

Предикатные структуры предложений анализируемого текста являются входными единицами для последующей обработки информации, почти повторяющей обработку, описанную в разделе 1, для выявления ключевых понятий текста. Синтаксические группы канонического представления предложения являются элементами звездочек в терминах такой обработки [Харламов, 2008]. Главное слово синтаксической группы соответствует главному понятию звездочки c_i , оно связано с второстепенными словами, которые являются его ближайшими ассоциантами — $\langle c_j \rangle$. В отличие от подобного представления в разделе 1, звездочки являются не парами, а тройками. В этом случае под размеченной (неоднородной) семантической сетью N понимается совокупность несимметричных троек понятий $\langle c_i, r_o, c_j \rangle$, где c_i и c_j — понятия, связанные между собой отношением r_o , принадлежащим множеству R отношений, принятых в используемой модели мира:

$$N \cong \{ \langle c_i, r_o, c_j \rangle \}. \quad (6)$$

Несимметричность семантической сети предполагает неравенство двух троек: $\langle c_i, r_o, c_j \rangle \neq \langle c_j, r_o, c_i \rangle$. Семантическая сеть, описанная таким образом, может быть переописана как множество звездочек: $z_i = \langle c_i \langle r_o, c_j \rangle \rangle$:

$$N \cong \{ z_i \} = \{ \langle c_i \langle r_o, c_j \rangle \rangle \}. \quad (7)$$

Под звездочкой $z_i = \langle c_i \langle r_o, c_j \rangle \rangle$ в этом случае понимается конструкция, включающая главное понятие c_i связанное с множеством понятий-ассоциантов c_j , которые являются семантическими признаками главного понятия, отстоящими от главного понятия на одну связь. Связи направлены от главного понятия к понятиям-ассоциантам, и размечены отношениями r_o .

Семантическая сеть строится из таких звездочек. Второстепенное слово первой звездочки z_1 становится главным во второй звездочке z_2 . И так далее. В некоторый момент какое-нибудь второстепенное слово одной из предыдущих звездочек z_i становится главным в следующей звездочке (уже встречавшейся ранее) z_1 . Все

вместе они формируют сеть – циклический граф. Причем частоты встречаемости характеризуют первичные весовые характеристики понятий звездочки, а связь получает метку отношения, которое ею реализуется. Таким образом, сеть становится неоднородной (размеченной), в отличие от однородной (ассоциативной) сети в разделе 1. Сеть строится с учетом анафор и сочинений в рамках абзаца таким образом, что главное слово первого предиката первого анализируемого предложения данного абзаца становится главным словом первой звездочки. Главные слова вершинных предикатов всех остальных предложений одного абзаца становятся второстепенными понятиями в этой первой звездочке и присоединяются к главному слову сочинительными (ассоциативными) связями r_s . Подчиненные предикаты присоединяются к второстепенным словам первой звездочки своими главными словами, каждый – на свое место. Связи в предикатных звездочках размечаются типами связей r_p , принятых в заданной модели предметной области $R \in \{r_s, \{r_p\}, r_a\}$. На свои места присоединяются атрибутивными связями r_a определяющие понятия.

Далее, также как в разделе 1, первичная сеть перенормируется. Частоты встречаемости p_i и попарной встречаемости p_{ij} перевычисляются в смысловые веса w_i . Получаем окончательную неоднородную (размеченную) семантическую сеть.

Необходимо напомнить, что связи, полученные на этапе синтаксического анализа предложений, дополняются сочинительными связями, полученными в результате раскрытия анафорических ссылок и объединения предложений абзаца как смысловой единицы.

Итак, на этом этапе обработки сначала, на основе предложений, содержащих ключевые понятия, выделенные на этапе лексической обработки, из предикатных пар формируется частотный портрет текста. Его единицами являются пары понятий, выделенные на этапе анализа предложений, а связями – попарные включения этих понятий, с учетом их связей, анафор и сочинений в рамках абзаца. После формирования частотного портрета частоты встречаемости единиц перенормируются в смысловые веса. Взвешенные по смыслу предикатные структуры ранжируются и составляют иерархию предикатных структур, описывающих предметную область.

4. Прагматический анализ текста

Описанным в разделе 3 способом строятся семантические сети целых предметных областей. В этом случае сети формируются на основе корпусов текстов, описывающих предметные области.

Последующий прагматический анализ входного текста заключается в выделении цепочки наиболее весомых ключевых предикатных структур, характерных для предметной области, к которой относится текст. Объем этой цепочки зависит от порога, примененного к смысловому весу предикатных структур (выделение главных предикатных структур), а порядок этих предикатных структур – от порядка их следования в предложениях текста. Такая цепочка полностью характеризует смысловое содержание текста в терминах выбранной предметной области.

Отнесение текста к одной из предметных областей осуществляется сравнением ассоциативной сети текста и ассоциативных сетей предметных областей (классификация) [Харламов, 2008]. Степень пересечения ассоциативных сетей текста и предметной области характеризует смысловое пересечение текста и предметной области.

Последовательность главных предикатных структур, выбранных из текста, с использованием порогового преобразования, описывает основное

ЗАКЛЮЧЕНИЕ

Автоматический смысловый 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. Замена первичной обработки текста, заключающейся в морфологическом анализе слов, на первичную обработку, связанную с выявлением предикатных структур в предложениях текста, позволяет перейти от автоматического формирования ассоциативной сети текста к автоматическому формированию неоднородной семантической сети. Проекция исходного текста на сформированную семантическую сеть, с учетом смысловых весов ключевых понятий, позволяет выявить прагматическую структуру текста как последовательность наиболее существенных предикатных структур в данном тексте.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Ягунова Е.В., 2008] Ягунова Е.В. Вариативность стратегий восприятия звучащего текста (экспериментальное исследование на материале русскоязычных текстов разных функциональных стилей) / – Пермь: Пермский университет, 2008 - 395с.
- [convera.com, 2011] URL: <http://www.convera.com>.
- [Kharlamov, 2004] Kharlamov A.A. Networks constructed of neuroid elements capable of temporal summation of signals / Alexander A. Kharlamov, Vladimir V. Raevsky // “Neural Information Processing: Research and Development”, Jagath C. Rajapakse and Lipo Wang, Editors, “Springer-Verlag”, May, 2004 - 478 p., ISBN 3-540-21123-3.
- [Харламов, 2001] Харламов А.А. Нейросетевая технология представления и обработки информации (естественное представление знаний). / А.А. Харламов // - М.: «Радиотехника», 2006. - 89 с.
- [Gartner Symposium, 2001] Data Mining, Gartner Symposium ITXPO 2001.

[Сокирко А.В., 2001] Сокирко А.В. Семантические словари в автоматической обработке текста (по материалам системы ДИАЛИНГ) / А.В. Сокирко // Диссертация на соискание ученой степени кандидата технических наук – М.: 2001.

[Харламов, 2008] Харламов А.А. Перестройка модели мира, формируемой на материале анализа текстовой информации с использованием искусственных нейронных сетей, в условиях динамики внешней среды. / А.А. Харламов, В.В. Раевский // Речевые технологии, N 3, 2008. Стр. 27-35.

[Sullivan, 2001] Sullivan D. Document Warehousing and Textmining./ Dan Sullivan // - N.Y.: "Wiley publishing house", 2001.

[Hopfield, 1982] Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. / J.J. Hopfield // Proc. Natl. Acad. Sci. 79, 1982. Pp. 2554 – 2558.

[Белошапкова и др., 1997] Белошапкова В.А. Современный русский язык: Учебник для филологических специальностей высших учебных заведений / В.А. Белошапкова [и др.] // Под ред. Белошапковой – 3-е изде, испр. и доп. - М.: Азбуковник, 1997 - 928 с.

[Харламов и др., 2011] Харламов А.А. Метод выделения главных членов предложения в виде предикатных структур, использующий минимальные структурные схемы. / А.А. Харламов, Т.В. Ермоленко, Г.В. Дорохина, Д.С. Гнигтько // Речевые технологии, N 2, 2011 (В печати).

AUTOMATICAL MAKING OF HETEROGENEOUS SEMANTIC NETWORK ON THE BASE OF EXTRACTION OF KEY PREDICATE STRUCTURES OF TEXT SENTENCES

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

In the work a new approach to automatical extraction of semantic relationships of texts keywords is presented. The TextAnalyst technology of making semantic network is used but instead of key words the approach uses key predicate structures. This approach combines two other ones: the extraction of predicate structure of sentence as a sense of sentence; and extraction of key words of text with their relationships as a semantic network of the whole text. Their combination is a detailed sense of text which can be extracted automatically.

The main text content can be shown as a list of its key words for example. Extraction of key words from text is not a problem now [Kharlamov, 2004; Харламов, 2001]. But there are no relationships their as a very important information. Extraction of the information about relationships of key words in text can be fulfilled by using its predicate structures. Automatic extraction of predicate structures from sentences of the text is the main goal of the work.

Automatic extraction of key word from text can be

made by TextAnalyst technology of automatic semantic text analysis. The main key words with their relationships are extracted from the text by using artificial neuron network on the basis of neuroliked elements with time summation of input signals [Kharlamov, 2004]. The semantic network elements (nodes and edges) have their weights. The network is homogenous (associative) in this case.

For doing the analysis more exact we can use the information about relationships between the key words. In such a way we are making the heterogeneous semantic network. For this we can use predicate structures of the text sentences which we can extract by using special syntactical rules [Харламов, 2011].

Extracted from the text sentences predicate structures include relationships of their actants. Those relationships allow paint the edges of previously prepared semantic network.

The sequence of the predicate structures which include the main key words of the text characterize the text pragmatic.

The TextAnalyst technology allows extract automatically the main text key words in their associative relationships. By changing the text preprocessing we can automatically extract from the text not only associative but other types of relationships and do the extracted semantic network the heterogeneous one. By projecting the text onto the semantic network we can find in the text a sequence of the main predicate structures which include the text pragmatic.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

СЕМАНТИЧЕСКИЕ УНИВЕРСАЛИИ В СИСТЕМЕ ОПИСАНИЯ ЗНАЧЕНИЙ МОРФЕМ

Сулейманов Д.Ш., Гатиатуллин А.Р.

*НИИ «Прикладная семиотика»
Академии наук Республики Татарстан,
Казанский (Приволжский) Федеральный университет
г. Казань, Россия
dvdt.slt@gmail.com
agat1972@mail.ru*

В работе описываются семантические универсалии, представляющие собой семантические сценарии и в совокупности составляющие реляционно-ситуационную систему. Реляционно-ситуационная система предназначена для описания значений аффиксальных морфем, представленных в структурно-функциональной модели морфем. Семантические универсалии рассматриваются на примере описания значений падежной морфемы –*ГА* (директив).

Ключевые слова: семантические универсалии, модель морфем, реляционно-ситуационная система, семантический аспект

ВВЕДЕНИЕ

Такие извечные и, одновременно, не имеющие до сих пор эффективных решений проблемы, как понимание текстов на естественном языке, автоматизированный поиск многоязычной информации, машинный перевод с языка на язык, и ряд других, настраивают исследователей на поиск «прорывных» идей и подходов в этой области. Практически является общепризнанным, что возможное решение лежит на плоскости создания семантических технологий, позволяющих проникновение на глубинный, «доязыковой», уровень значения (смысла) лингвистического объекта (морфемы, слова, предложения, фразы и т.д.). Вместе с тем, далее исследователи расходятся как в интерпретации самого понятия семантика, так и в методах и принципах ее описания и извлечения. Одни пытаются построить универсальный язык, систему кодирования смысла [Мартынов, 1994; Лезин и др., 1996], другие – контекстные семантические формулы, грамматики, семантические роли и падежи [Шенк, 1980; Филлмор, 1981; Апресян и др., 1982; Бухараев и др., 1990], третьи – семантические универсалии, как основы универсальной интерпретации, претендующие на роль инструментария, в совокупности наиболее полно описывающего смысловые конструкции ЕЯ-текстов независимо от

языка и предметной области [Поспелов, 1981; Осипов, 2006; Закиев, 1992].

В данной работе авторы описывают семантические универсалии в системе представления значений морфем, которые при их дальнейшем развитии могут быть использованы для семантического описания и более сложных лингвистических конструкций. Как известно, морфемы являются наименьшими значащими (семантическими) единицами языка, из которых составляется словоформа, а далее, соответственно, и лексема. А из лексем составляются предложения и фразы. Значение морфологического уровня при обработке ЕЯ-текстов, особенно для языков агглютинативного типа, общепризнанно и создание открытой концептуально-функциональной модели морфем, отражающей свойства морфем во всех проявлениях его в структуре текста, включая фонетический, морфологический, синтаксический и семантический уровни, является несомненно актуальной и важной задачей. Ценность и прагматическая ориентированность концептуально-функциональных моделей, отнесенных нами к лингвистическим моделям концептуально-функционального класса [Сулейманов, 1998], главным образом, заключается в том, что они являются универсальными, наиболее полными описаниями значимых единиц языковых уровней и их грамматик, и являются той базой

«лингвистического строительного материала», из которого на основе прагматически-ориентированной технологии строятся лингвистические модели как основа систем обработки ЕЯ-текстов определенного типа. К тому же, имея максимально полные концептуально-функциональные модели для составляющих элементов на разных языковых уровнях мы можем на их основе организовать тот самый эффективный асинхронный децентрализованный анализ продукта языковой деятельности, о которой говорится в работах Нариньяни А.С. [Труды, 1996]. Несмотря на методологическую, теоретическую и практическую ценность, такого рода модели крайне слабо исследованы и отражены в литературе. Также косвенным подтверждением важности и необходимости построения такого рода моделей являются аналогичные исследования, проводимые за рубежом [Casey и др., 1992]. В работе [Pulman, 1995] утверждается близкая мысль о том, что интеллектуальная обработка естественного языка в реальных приложениях требует наличия лексикона, который обеспечивает разработчика богатой информацией о морфологических, синтаксических и семантических характеристиках слов, хорошо структурированных, и которые могут быть эффективно применены. Эти цели могут быть достигнуты разработкой инструментария, который облегчает приобретение лексической информации с машиночитаемых словарей и корпусов текстов, также из баз данных и теоретических знаний о слове, предлагаемых в кодированном виде, необходимом для целей NLP (*Natural Language Processing*). В этих работах, как правило, исследуются языковые единицы и их характеристики на уровне слов (полисемия, описание значений, допустимые употребления слова). Учитывая особую важность морфологического уровня для татарского языка как языка агглютинативного типа, а также отсутствие такого рода исследований и технологических решений, авторами данной статьи разработана и описана компьютерная структурно-функциональная модель аффиксальных морфем, которая вполне может быть использована также для описания морфем других языков [Сулейманов и др., 2003].

Наибольшую сложность при создании структурно-функциональной модели морфем вызвала разработка формальной структуры для отражения значений (семантики) морфем. Практически, впервые возникла необходимость описания значения татарских морфем на некотором формальном языке «глубинных структур», т.е. на языке семантического уровня. В данной работе дается краткое описание структурно-функциональной модели морфем и описание реляционно-ситуационной системы, состоящей из совокупности семантических сценариев, называемых нами семантическими универсалиями, служащими для обозначения позиции той или иной морфемы в соответствующем контексте. С целью обеспечения компактности изложения и раскрытия

сути идеи семантических универсалий для описания значений морфем, здесь приводится не полное описание всей совокупности универсалий, а лишь той ее части, которая используется для контекстного описания значений морфемы –ГА, маркируемой в татарской грамматике как аффиксальная морфема направительного падежа (или директив).

1. Структурно-функциональная модель аффиксальных морфем

Структурно-функциональная модель аффиксальных морфем представляет собой концептуально-функциональное описание элементов морфологии и используется для построения прагматически-ориентированных морфологических модулей как основы лингвопроцессоров. Особенностью данной модели является то, что она позволяет осуществить полную «инвентаризацию» татарских аффиксальных морфем с описанием функций и ситуаций проявления их на всех языковых уровнях (формирования звука, словоформы, предложения, значения). Важными при построении компьютерной структурно-функциональной модели являются концептуальные (полнота описания характеристик), технологические (удобство обработки, прагматичность) и инструментальные аспекты разработки и реализации (открытость, наличие функций и средств для доступа и применения).

Из названия самой модели - структурно-функциональная, следует, во-первых, что она структурно, т.е. на разных языковых уровнях (фонологическом, морфологическом, синтаксическом, семантическом) отображает проявление свойств морфемы. Модель является открытой, что позволяет вносить в нее даже единичные, незначимые на первый взгляд, характеристики или проявления аффикса в словоформе, при необходимости модифицируя и саму структуру. Во-вторых, модель функциональная, т.е. после наполнения ее лингвистами-экспертами является информационно-справочной и учебно-консультационной базой при изучении и преподавании татарского языка, а также мета-моделью, используемой для разработки прагматически-ориентированных лингвистических моделей в качестве формальной базы для лингвопроцессоров. Модель имеет структуру, в которой каждая морфема характеризуется по пяти основным параметрам, называемым аспектами:

- 1) Функциональный;
- 2) Морфологический;
- 3) Синтаксический;
- 4) Семантический;
- 5) Морфонологический.

Каждый основной параметр подразделяется на подпараметры, а те, в свою очередь, также могут подразделяться на подпараметры и т.д. [Сулейманов

и др., 2003]. При описании семантики морфем мы исходим из утверждения, что каждая морфема используется для кодирования того или иного значения в некотором контексте, отражающем некую локальную "модель мира" или здравый смысл "говорящего", отображаемого в "слушающем". Использование аффиксальных морфем, внося определенные семантические оттенки, ограничения в значение корневой морфемы, позволяет существенно сократить количество лексем для передачи (кодирования) некоего смысла, то есть служит как элемент, редуцирующий лексическое пространство, необходимое для формирования контекста. Локальная модель мира представляет собой формализованное описание некоторого контекста, отражающего объекты и их отношения. Разделение лексем или групп лексем на объекты и отношения является достаточно условной процедурой и зависит от семантических ролей, исполняемых лексемами или группами лексем, отражающими некие значения в определенном контексте.

Примечание 1. Наша задача заключается не в том, чтобы построить семантическую модель, которая позволяет "понять", определить "смысл" некоторого текста, а в том, чтобы построить локальную концептуальную модель мира, которая может быть соотнесена со значениями конкретных морфем и, соответственно, размечена соответствующими им поверхностными выражениями аффиксальных морфем.

Как известно, в лингвистике разделяются такие понятия, как значение высказывания (или сущности) и его пресуппозиция [Моделирование, 1997]. Пресуппозиция определяется как предшествующий контекст, предшествующее знание, или как контекст, в котором происходит определение значения сущности. Очевидно, что понимание сущности, прежде всего, обеспечивается именно пресуппозицией. Следовательно, при определении значений аффиксальной морфемы важно рассматривать ее не только как часть словоформы, а также как составляющую контекста, образующего пресуппозицию.

Попытка описать значения каждой морфемы в подразделе «Семантический аспект» компьютерной модели татарских морфем в некотором унифицированном виде, с целью их дальнейшего использования в лингвопроцессорах, привела к необходимости систематизации отношений, выражаемых аффиксальными морфемами, и создания целостной картины этих отношений. Как известно, значение морфемы, зачастую, может быть установлено только исходя из контекста, в котором она встречается и является трудноформализуемым параметром. Тем не менее, создание некоего формального аппарата, возможно, нечеткого, позволяющего максимально полно отразить те отношения, в лексикализации которых участвуют рассматриваемые морфемы, повышает

репрезентативность, прозрачность и прикладную ценность модели. 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 – комбинация нескольких взаимосвязанных семантических сценариев, называемые ситуациями. Соответственно, семантические сценарии являются теми самыми семантическими универсалиями, которые используются для описания глубинного смысла определенного контекста и включаются в модель описания значений морфем. В дальнейшем происходит локализация конкретной морфемы в соответствующей позиции семантического универсалия.

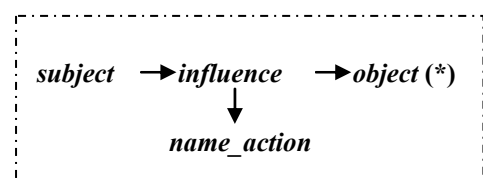
В этом разделе приводится описание семантических универсалий на примере аффиксальной морфемы *-ГА*, являющихся частью реляционно-ситуационной системы и используемых в структурно-функциональной модели для формального представления значений морфем.

2. Описание семантических универсалий для формального представления значений аффиксальной морфемы *-ГА*

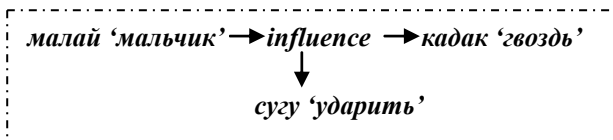
Аффиксальные морфемы по определению, как минимальные значащие единицы языка, имеют хотя бы одно значение, проявляющееся при использовании его в словоформе. В татарском языке, как правило, в зависимости от окружения, аффиксальные морфемы имеют различные интерпретации, точнее, используются для поверхностного оформления различного рода контекстов. В результате исследований нами выявлено 19 контекстных значений аффиксальной морфемы *-ГА*. Это означает, что данный аффикс в составе словоформы участвует в формировании (или поверхностном описании, лексическом проявлении) 19 контекстных значений.

Далее рассмотрим 10 семантических универсалий, в заполнении которых лексическими единицами по заданному тексту участвует морфема *-ГА*. В рассматриваемых универсалиях позиция, маркируемая (заполняемая) данной морфемой, обозначена символом звездочка (*).

Значение 1. Зависимое имя с аффиксом *-ГА* выражает объект физического воздействия.

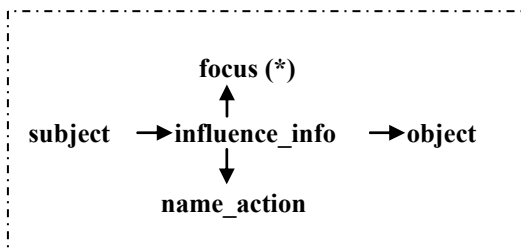


малай кадакка сукты 'мальчик ударил по гвоздю'

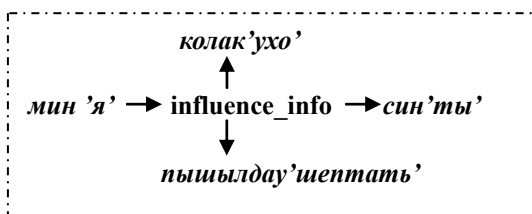


В реляционно-ситуационной системе рассматриваемая словоформа заполняет следующую ролевую ситуацию: **Отношение Действие: Действие-воздействие: Объект, испытывающий физическое воздействие.**

Значение 2. Зависимое имя с аффиксом **-ГА** выражает объект информационного воздействия

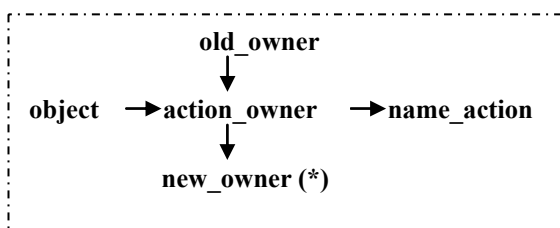


мин синең колагыңа пышылдадым 'я прошептал тебе на ухо'

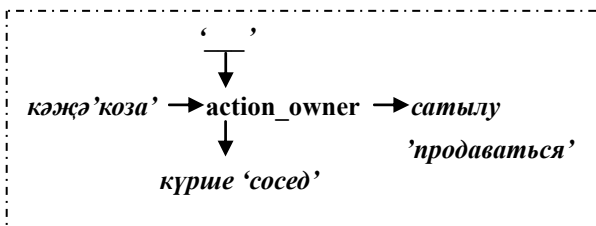


В реляционно-ситуационной системе рассматриваемая словоформа заполняет следующую ролевую ситуацию: **Отношение Действие: Отношение воздействия: Объект, испытывающий информационное воздействие.**

Значение 3. Зависимое имя с аффиксом **-ГА** выражает нового владельца объекта:



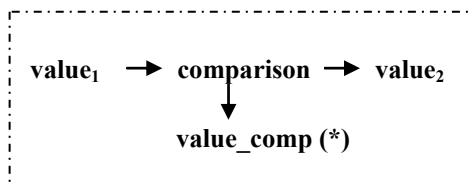
кәжә күршегә сатылды 'коза продана соседу'



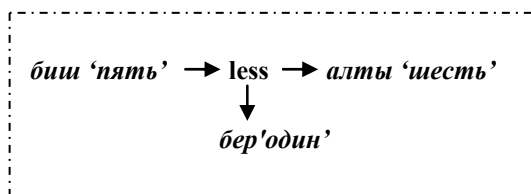
В реляционно-ситуационной системе рассматриваемая словоформа заполняет следующую

ролевую ситуацию: **Отношение Действие: Действие по изменению отношения принадлежности: Новый владелец.**

Значение 4. Зависимое имя с аффиксом **-ГА** выражает разницу (степень различия) между значениями.

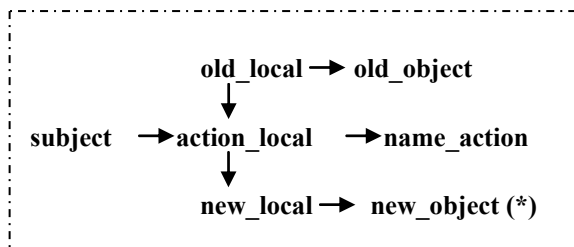


биш алтыдан бергә азрак 'пять на единицу меньше шести'

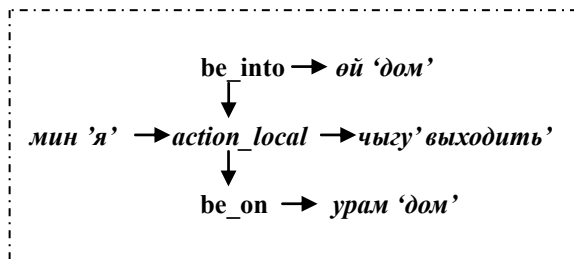


В реляционно-ситуационной системе рассматриваемая словоформа заполняет следующую ролевую ситуацию: **Отношение Сравнения: Разница между значениями.**

Значение 5. Зависимое слово с аффиксом **-ГА** выражает конечный пункт направления процесса, обозначенного глаголом движения.



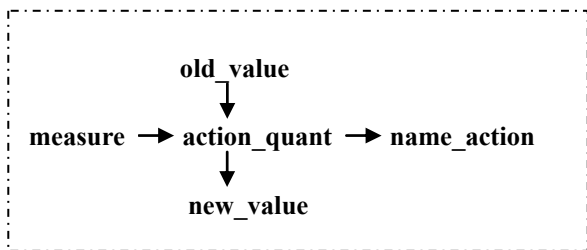
мин өйдән урамга чыктым 'я вышел из дома на улицу'



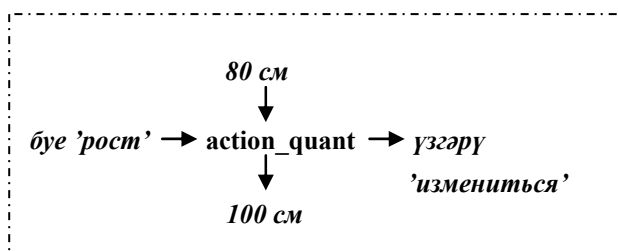
В реляционно-ситуационной системе рассматриваемая словоформа заполняет следующую ролевую ситуацию: **Отношение Действие: Действие по изменению пространственного отношения: Новое местоположение субъекта.**

Значение 6. Зависимое имя с морфемой **-ГА**, которое имеет при себе конкретизирующее

числительное, подчиняясь глаголам действия, выражает меру и степень действия в пространстве и во времени.

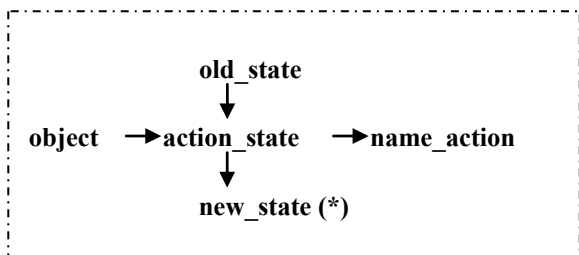


буе 80 сантиметрдан 100 сантиметрга кадәр үзгәрде 'рост изменился с 80 до 100 сантиметров'

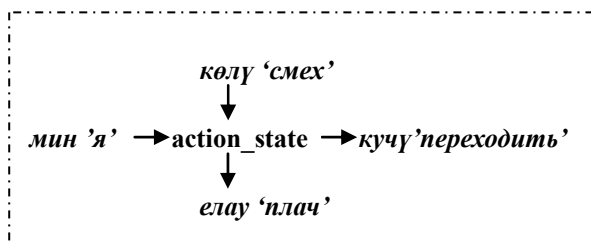


В реляционно-ситуационной системе рассматриваемая словоформа заполняет следующую ролевую ситуацию: **Отношение Действие: Действие по изменению количественного отношения: Разница между значениями.**

Значение 7. Зависимое слово с аффиксом -ГА выражает новое состояние объекта.

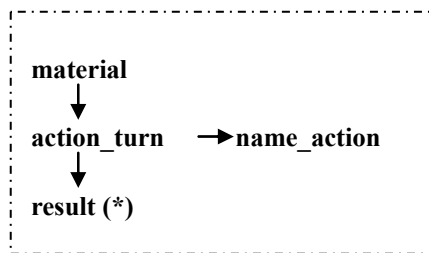


мин көлдән елауга күчтем 'я перешел от смеха к плачу'

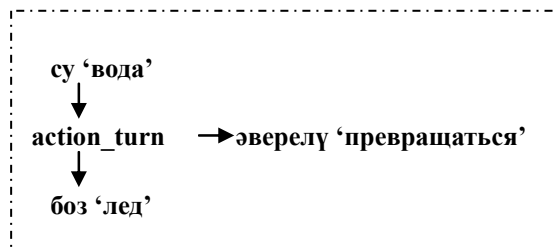


В реляционно-ситуационной системе рассматриваемая словоформа заполняет следующую ролевую ситуацию: **Отношение Действие: Действие по изменению состояния: Новое состояние объекта.**

Значение 8. Зависимое слово с аффиксом -ГА выражает новый объект.

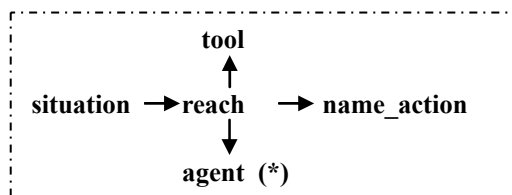


су бозга эверелде 'вода превратилась в лед'

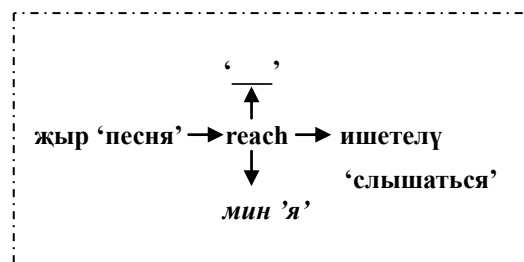


В реляционно-ситуационной системе рассматриваемая словоформа заполняет следующую ролевую ситуацию: **Отношение Действие: Действие по изменению состояния: Новый объект.**

Значение 9. Зависимое слово с аффиксом -ГА выражает субъекта восприятия ситуации.

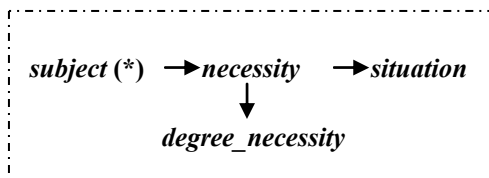


миңа жыр ишетелде 'мне послышалась песня'

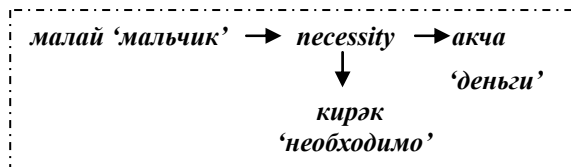


В реляционно-ситуационной системе рассматриваемая словоформа заполняет следующую ролевую ситуацию: **Отношение Действие: Физическое восприятие: Объект восприятия некоторой ситуации.**

Значение 10. Зависимое слово с аффиксом -ГА является субъектом, к которому обращена модальность.



малайга акча кирәк 'мальчику нужны деньги'



В реляционно-ситуационной системе рассматриваемая словоформа заполняет следующую ролевую ситуацию: **Отношение Модальность: Отношение необходимости.**

ЗАКЛЮЧЕНИЕ

Семантические универсалии, описанные в данной статье для представления значений морфем, при их дальнейшем развитии могут быть использованы для семантического описания и более сложных лингвистических конструкций.

Главная идея заключается в том, чтобы создать целостную реляционно-ситуационную систему, представляющую совокупность семантических универсалий, которые могут быть использованы как инвентарная база для формального кодирования семантических ситуаций в картине мира и возможного их дальнейшего описания на некотором естественном языке. Очевидно, семантические сценарии, составленные из таких языко- и предметно-независимых универсалий могут стать хорошей базой для создания интеллектуальных программ, таких как программы машинного перевода, извлечения знаний и машинного поиска.

Поддержан грантом РФФИ № 12-07-00615-а «Модели и методы реляционно-ситуационного анализа текстов тюркской группы языков для задач семантического поиска».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Мартынов, 1984] Мартынов В.В. Универсальный семантический код: УСК-3. – Минск: «Наука и техника». – 1984. – 131с.
- [Лезин и др., 1996] Лезин Г.В., Боярский К.К., Попов А.И. Систематизация информации: семантическое кодирование текстов // Труды межд. семинара ДИАЛОГ-96: компьютерная лингвистика и ее приложения (Пушино, 4-9 мая, 1996г.). –М., 1996. – С.131-136.
- [Шенк, 1980] Шенк Р. Обработка концептуальной информации. – М.: Энергия, 1980. – 361 с.
- [Филлмор. 1981] Филлмор Ч. Дело о падеже // В кн.: Новое в зарубежной лингвистике. Вып. X. Лингвистическая семантика. М.: Прогресс. 1981.
- [Апресян и др., 1992] Апресян Ю.Д., Богуславский И.М., Иомдин Л.Л. и др. Лингвистический процессор для сложных информационных систем. – М.: Наука, 1992. – 256 с.
- [Бухараев и др., 1990] Бухараев Р.Г., Сулейманов Д.Ш.

Семантический анализ в вопросно-ответных системах Казань: Изд-во Казан. ун-та. – 1990. – 124 с.

[Поспелов, 1981] Поспелов Д.А. Логико-лингвистические модели в системах управления. – М.: Энергоиздат, 1981. – 232 с.

[Осипов и др., 2006] Осипов Г.С., Тихомиров И.А., Смирнов И.В. Eхactus – 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.

[Закнев, 1992] Закнев М.З. Татарская грамматика. Т3. Синтаксис. – Казань: Таткнигоиздат. 1992. – 488 с.

[Сулейманов, 1998] Сулейманов Обработка ЕЯ-текстов на основе прагматически-ориентированных лингвистических моделей Сб.под ред. Соловьева В.Д.: Обработка текста и когнитивные технологии. -Вып.3. Труды научного семинара “Когнитивное моделирование” (Пушино, октябрь 1998 г.). – С.205-212.

[Труды , 1996] Труды Международного семинара Диалог-96: компьютерная лингвистика и ее приложения (Пушино, 4-9 мая, 1996г.). Под ред. А.С.Нариньяни. – М.: 1996. – 305 с.

[Casey и др., 1992] Casey, Nagy G. 1992. At the frontiers of OCR. Proceedings of the IEEE, (80(7):1093-1100.). – 1992.

[Pulman, 1995] Pulman Stephan G. Semantics // In Cole Ronald A., Mariani Joseph, Uszkoreit Hans, et al (editors). Survey of the State of the Art Human Language Technology. – 1995/. – P.122-129 / ftp://speech.cse.ogi.edu/pub/docs/HLT/.

[Сулейманов и др., 2003] Сулейманов Д.Ш., Гатиатуллин А.Р. Структурно-функциональная компьютерная модель татарских морфем. – Казань: Фэн, 2003. – 220с.

[Моделирование, 1987] Моделирование языковой деятельности в интеллектуальных системах. -М., Наука. Главная ред. физ.-матлит., 1987. –280 с. /Под ред. А.Е. Кибрика А.С. Нариньяни; С предисловием А.П. Ершова.

SEMANTIC UNIVERSALS IN THE SYSTEM OF DESCRIPTION OF MORPHEME MEANINGS

Suleymanov D.Sh., Gatiatullin A.R.

«Scientific Research Institute of Applied Semiotics» of Academy of Sciences of the Republic of Tatarstan,
Kazan (Volga region) Federal University,
Kazan, Russia

dvdt.slt@gmail.com

agat1972@mail.ru

The article offers a description of semantic universals which represent semantic scenarios and form in total an object-predicate system. The object-predicate system is aimed at describing the meanings of affixal morphemes presented in the structural-functional morphemic model. Semantic universals are studied upon the example of description of the meanings of a case morpheme *-GA* (directive).

Key words: semantic universals, morphemic model, relation-situation system, semantic aspect



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.432.4

ИНТЕЛЛЕКТУАЛЬНЫЙ МОРФОЛОГИЧЕСКИЙ АНАЛИЗАТОР, ОСНОВАННЫЙ НА СЕМАНТИЧЕСКИХ СЕТЯХ

Шарипбаев А.А., Бекманова Г.Т., Ергеш Б.Ж., Бурибаева А. К., Карабалаева М.Х.

*Евразийский национальный университет имени Л. Н. Гумилева,
г. Астана, Казахстан*

sharalt@mail.ru
gulmira-r@yandex.ru
banu_kazakh@yahoo.com
buribayeva@mail.ru
mkarabal@mail.ru

Разработан интеллектуальный морфологический анализатор казахского языка на основе формализации морфологических правил с помощью семантических сетей.

Ключевые слова: морфологические правила, морфологический анализатор, семантические признаки, семантическая сеть, синхронизированное линейное дерево.

ВВЕДЕНИЕ

Очевидная сложность обработки естественно-языковых процессов вызвана трудностью их формализации. Сложность заключается в невозможности словоизменения слов для какой либо части речи по заданной траектории без предварительной обработки словаря начальных форм, поскольку существует зависимость словоизменения слова от его смысла, то есть от его семантического содержания.

1. Построение семантической базы начальных форм слов

В казахском языке словоформы образуются путем конкатенации корня аффиксов (суффиксов и окончаний). При этом каждый аффикс связан с набором семантических признаков и порядок добавления аффиксов строго определен. Например, для имен существительных к основе слова вначале добавляется суффикс и далее окончание множественного числа, затем притяжательное окончание, далее следует падежное окончание и последним окончание формы спряжения (добавляется только кодушевленным существительным) [Казахская грамматика, 2002].

Для формализации правил добавления суффиксов и окончаний используются семантические сети, в которых вершины

представляют морфологические единицы, а дуги задают отношения между ними. Для разработки морфологического анализатора используется база данных начальных форм слов, на которой будет формироваться словарь словоформ казахского языка со всеми семантическими признаками.

Новые словоформы образуются с учетом морфологических и семантических признаков начальных форм следующим образом: сначала к начальной форме слова добавляются суффиксы. Затем, двигаясь слева направо, определяется категория (глухие, звонкие и т.п.) последней буквы (последнего звука) начальной формы слова для добавления того или иного окончания [Бекманова, 2009].

В качестве семантических признаков начальных форм слов выступают такие категории как часть речи, одушевленность и неодушевленность для имен существительных, образование сравнительных и превосходных степеней прилагательных, образование собирательных и порядковых числительных, для глагола сочетание в сложных формах с вспомогательными глаголами как «отыр», «тұр», «жатыр», «жүр» и др. Всего в базе знаний, по которой осуществляется словоизменение, более 100 семантических признаков. База данных начальных слов с семантическими признаками представлена на рисунке 1.

word	имя существительное	прилагательное	числительное	глагол	местоимение
анализ	1	0	0	0	0
аналитик	1	0	0	0	0
аналитикалық	0	1	0	0	0
аналық	0	1	0	0	0
анар	1	0	0	0	0
анархизм	1	0	0	0	0
анархист	1	0	0	0	0
анау	0	0	0	0	1
анықтама	1	0	0	0	0
аңғал	0	1	0	0	0
анғар	1	0	0	0	0
аңғар	1	0	0	0	0
аңғарғыш	0	1	0	0	0
аңғарлы	0	1	0	0	0
аңғару	0	0	0	1	0
аңғарымпаздық	1	0	0	0	0

Рисунок 1 – Вид база данных начальных форм слов казахского языка

2. Формализация морфологических правил казахского языка.

Для формализации правил добавления суффиксов и окончаний предлагается использовать семантическую нейронную сеть, предложенную в [Шуклин, 2001]. С помощью такой сети генерируются словоформы казахского языка и порождается структура словаря начальных форм в виде синхронизированного линейного дерева.

Для представления словоформы и ее признаков будут использоваться следующие метасимволы:

- # - разделитель между словами,
- (- начало слова,
-) - конец слова,
- ! - начало признака словоформы (падеж и т. д.),
- * -конец признака словоформы.

Рассмотрим пример для слова «бала - ребенок» (основа слова) и двух его словоформ «балам - мой ребенок», «балаң - твой ребенок» (в казахском языке одушевленные существительные изменяются по лицам с помощью личных окончаний). Рецептор возбуждается на символ начала слова «(». Далее переходит в состояние «б», при подаче символа «б», далее последовательно «(ба», «(бал», «(бала)», и затем одновременно два субсостояния «(балам)» и «(балаң)» (рисунок 2).

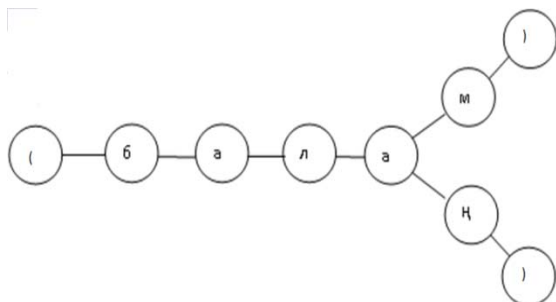


Рисунок 2– Синхронизированное линейное дерево для словоформ

На рисунке 3 приведен пример структуры связей леммы, определяющей следующие признаки: имя существительное(зат есім) –«!зе*», одушевленное – «!жа*», притяжательное окончание (тәуелдікжалғау)первого лица – «!11*» (бірінші жак),притяжательное окончание (тәуелдікжалғау)второго лица – «!22*»(екінші жак). При подаче на лемму слова «(балам)» она переходит в возбужденные субсостояния: «(балам)», «!зе*», «!жа*», «!11*»а при подаче слова «балаң» в возбужденные субсостояния: «(балаң)», «!зе*», «!жа*», «!22*».

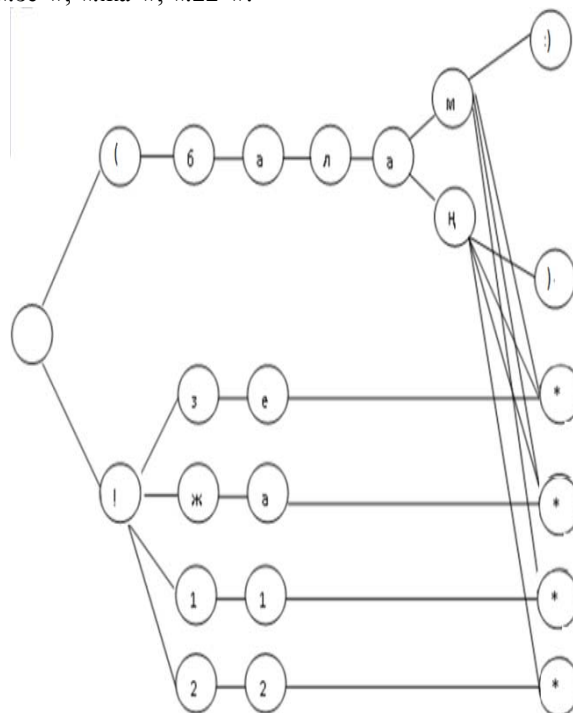


Рисунок 3 – Синхронизированное линейное дерево для словоформ и их морфологической информации

Нейроны рецепторы распознают отдельные символы входной символьной последовательности. На выходе рецептор генерирует сигнал, означающий наличие или отсутствие соответствующего символа в анализируемом тексте. Нейроны-эффекторы выдают результат распознавания отдельных фрагментов входной символьной последовательности. Заменив в синхронизированном линейном дереве сигнал от рецептора сигналом от эффектора того же дерева, получим возможность использовать в качестве входных символов фрагменты символьных последовательностей.

Для обозначения таких фрагментов во входной символьной последовательности будем применять метасимволы скобок: "(" и ")". Тогда приведенный пример переписывается в виде: ((бала)м), ((бала)ң), (((бала)м)ның), (((бала)ң)да) [Шарипбаев, 2009].

Таким образом формализованы все части речи казахского языка, общее количество формальных правил составило около 20000 записей, например

только для глагола это число 13000 формальных правил.

Ниже приводятся фрагмент формальных правил словоизменения на примере существительного с учетом закона сингармонизма, который обуславливает добавления мягких или твердых окончаний в зависимости от мягкости или твердости основы. Приведенный пример показывает фрагмент правил, где «зе» – затесім (имя существительное), «жа» - жанды (одушевленность), «01» заканчивается на твердые гласные а, о, ұ, «)» между закрывающими скобками помещены окончания существительных, после «!» морфологическая информация.

((жежа01)мын)!жі11
((жежа01)мыз)!жі11
((жежа01)сың)!жі22
((жежа01)сыңдар)!жі22
((жежа01))!жі33
((жежа01)м)!тә11
((жежа01)мыз)!тә110
((жежа01)ң)!тә22
((жежа01)ңыз)!тә22
((жежа01)сы)!тә33
((жежа01)лар)!кт
(((жежа01)лар)мыз)!ктжі11
(((жежа01)лар)сыңдар)!ктжі22
(((жежа01)лар))!ктжі33
(((жежа01)лар)ым)!кттә11
(((жежа01)лар)ымыз)!кттә110
(((жежа01)лар)ың)!кттә22
(((жежа01)лар)ыңыз)!кттә22
(((жежа01)лар)ы)!кттә33
((жежа01))!ат0
((жежа01)ның)!іл
((жежа01)ға)!ба
((жежа01)ны)!та
((жежа01)да)!жс
((жежа01)дан)!шы
((жежа01)мен)!кө
((жежа01)менен)!кө
(((жежа01)м)ның)!тә11іл
(((жежа01)м)а)!тә11ба
(((жежа01)м)ды)!тә11та
(((жежа01)м)да)!тә11жс
(((жежа01)м)нан)!тә11шы
(((жежа01)м)мен)!тә11кө
(((жежа01)м)менен)!тә11кө
(((жежа01)мыз)дың)!тә110іл
(((жежа01)мыз)ға)!тә110ба
(((жежа01)мыз)ды)!тә110та
(((жежа01)мыз)да)!тә110жс
(((жежа01)мыз)дан)!тә110шы
(((жежа01)мыз)бен)!тә110кө
(((жежа01)мыз)бенен)!тә110кө
(((жежа01)ң)ның)!тә22іл
(((жежа01)ң)а)!тә22ба
(((жежа01)ң)ды)!тә22та
(((жежа01)ң)да)!тә22жс
(((жежа01)ң)нан)!тә22шы
(((жежа01)ң)мен)!тә22кө
(((жежа01)ң)менен)!тә22кө

(((жежа01)ңыз)дың)!тә22іл
(((жежа01)ңыз)ға)!тә22ба
(((жежа01)ңыз)ды)!тә22та
(((жежа01)ңыз)да)!тә22жс
(((жежа01)ңыз)дан)!тә22шы
(((жежа01)ңыз)бен)!тә22кө
(((жежа01)ңыз)бенен)!тә22кө
(((жежа01)лар)дың)!ктіл
(((жежа01)лар)ға)!ктба
(((жежа01)лар)ды)!ктта
(((жежа01)лар)да)!ктжс
(((жежа01)лар)дан)!ктшы
(((жежа01)лар)мен)!кткө
(((жежа01)лар)менен)!кткө
(((жежа01)лар)ым)ның)!кттә11іл
(((жежа01)лар)ым)а)!кттә11ба
(((жежа01)лар)ым)ды)!кттә11та
(((жежа01)лар)ым)да)!кттә11жс
(((жежа01)лар)ым)нан)!кттә11шы
(((жежа01)лар)ым)мен)!кттә11кө
(((жежа01)лар)ым)менен)!кттә11кө
(((жежа01)лар)ымыз)дың)!кттә11іл
(((жежа01)лар)ымыз)ға)!кттә11ба
(((жежа01)лар)ымыз)ды)!кттә11та
(((жежа01)лар)ымыз)да)!кттә11жс
(((жежа01)лар)ымыз)дан)!кттә11шы
(((жежа01)лар)ымыз)бен)!кттә11кө
(((жежа01)лар)ымыз)бенен)!кттә11кө
(((жежа01)лар)ың)ның)!кттә22іл
(((жежа01)лар)ың)а)!кттә22ба
(((жежа01)лар)ың)ды)!кттә22та
(((жежа01)лар)ың)да)!кттә22жс
(((жежа01)лар)ың)нан)!кттә22шы
(((жежа01)лар)ың)мен)!кттә22шы
(((жежа01)лар)ың)менен)!кттә22кө
(((жежа01)лар)ың)бен)!кттә22кө
(((жежа01)лар)ың)бенен)!кттә22кө
(((жежа01)лар)ым)сыңдар)!кттә11жі22
(((жежа01)лар)ың)быз)!кттә22жі11
(((жежа01)лар)ы)сыңдар)!кттә33жі2

3 Алгоритм морфологического анализа

Предложен следующий алгоритм морфологического анализа слов казахского языка:

1. Слово считывается;
2. Открывается словарь начальных форм и в нем выполняется поиск считанного слова;
3. Если слово найдено, то перейти к шагу 12, иначе шаг 4 ;
4. Слово в цикле посимвольно считывается, начиная с последнего символа, то, что получается, ищем в словаре окончаний;
5. Если окончание найдено, то остаток ищем в словаре начальных форм;
6. Запоминаем морфологическую информацию окончания;

7. Если остаток слова найден в словаре начальных форм, то

8. Если это глагол, то считываем слово слева перед ним и определяем время глагола

9. Если это деепричастие, то считываем слово справа после него и определяем время глагола

10. Если это прилагательное, то считываем слово слева перед ним и проверяем его входит ли оно в список вспомогательных слов, использующихся для образования превосходных степеней, если входит, то это превосходная степень прилагательного.

11. Если такое слово не найдено, то переходим к шагу 4, иначе к шагу 12;

12. Конец.

Это алгоритм реализован на языке программирования BorlandDelphi.

С помощью построения морфологических правил и генерации слов казахского языка были получены следующие результаты:

- создана база данных начальных форм слов объемом 45 000 слов с разметкой частей речи и других признаков, необходимых для генерации словаря словоформ; получена формальная модель словоизменения и словообразования казахского языка с учетом семантики на основе семантической нейронной сети;
- автоматически сгенерирована база данных казахских словоформ объемом более 2 800 000 словарных статей с полной морфологической информацией;
- разработаны алгоритмы и программы морфологического анализа естественно-языковых текстов с учетом семантики на основе семантической нейронной сети и клеточных автоматов;

ЗАКЛЮЧЕНИЕ

Наибольшее количество словоформ генерируется из начальной формы существительного, прилагательного и начальной формы глагола. Полученные словари могут быть изданы в качестве орфографических словарей. Полученные формализации, методы и алгоритмы могут использоваться в системах обработки естественно-языковых текстов (орфографических корректорах, переводчиках, обучающих системах), системах распознавания и синтеза казахской речи, а также в семантических поисковых системах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[**Казахская грамматика, 2002**] Казахская грамматика. Фонетика, словообразование, морфология, синтаксис. –Астана, 2002.

[**Бекманова, 2009**] Бекманова Г. Т. Некоторые подходы к проблемам автоматического словоизменения и морфологического анализа в казахском языке // Вестник Восточно-Казахстанского государственного технического университета им. Д. Серикбаева. – Усть-Каменогорск: 2009г. – № 4. – с. 192-197.

[**Шуклин, 2001**] Шуклин Д. Е. Структура семантической нейронной сети, реализующей морфологический и синтаксический разбор текста// Кибернетика и системный анализ. Киев. Изд-во Института кибернетики НАН Украины, 2001. - № 5. с. 172-179.

[**Шарипбаев, 2009**] Шарипбаев А. А., Бекманова Г. Т. Построение логической семантики слов казахского языка // Материалы Всероссийской конференции с международным участием «Знания-Онтологии-Теории (ЗОНТ-09)». – Новосибирск: 2009. – Том 2. – с. 246-249.

THE INTELLECTUAL MORPHOLOGICAL ANALYZER BASED ON SEMANTIC NETWORKS

Sharipbaev A.A., Bekmanova G.T., Ergesh B.J.,
Buribaeva A.K., Karabalaeva M.H.

*Eurasian National University named after
L.N. Gumilev, Astana, Kazakhstan*

sharalt@mail.ru
gulmira-r@yandex.ru
banu_kazakh@yahoo.com
buribayeva@mail.ru
mkarabal@mail.ru

In this article the order of construction of semantic base of initial words is described, morphological rules of the Kazakh language are formalized. In the Kazakh language of a word form are formed by coupling of a root and affixes (suffixes and the terminations). Thus each affix is connected with sets of semantic signs and the order of addition of affixes is strictly defined. For example, for nouns to a word basis the suffix and further plural termination, then the possessive termination in the beginning is added, the case inflection further follows and the last the termination of the form of conjugation On the basis of formal rules develops algorithm of the morphological analyzer.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ БЕЛОРУССКО- И РУССКОЯЗЫЧНЫХ ЕЯ-ИНТЕРФЕЙСОВ ВОПРОСНО-ОТВЕТНЫХ СИСТЕМ

Гецевич Ю.С.*, Гецевич С.А. *, Елисеева О.Е.***, Житко В.А.***, Кузьмин А.А.**

** Объединенный институт проблем информатики
Национальной академии наук Беларуси,
г. Минск, Республика Беларусь*

Yury.Hetsevich@gmail.com

Novaeimya@gmail.com

***Белорусский государственный университет,
г. Минск, Республика Беларусь*

olae@open.by

kuzAleksAleks@gmail.com

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

zhitko.vladimir@gmail.com

В работе рассматривается семантическая технология проектирования естественно-языковых и речевых интерфейсов для интеллектуальных вопросно-ответных систем. Данная технология развивается в рамках открытого проекта OSTIS [Ostis, 2012]. Рассматривается также библиотека компонентов проектирования естественно-языкового интерфейса, возможности ее пополнения сторонними компонентами и создания новых компонентов.

Ключевые слова: анализ естественно-языковых текстов, анализ речевого ввода, естественно-языковой интерфейс, речевой синтез, технология проектирования.

ВВЕДЕНИЕ

В связи с динамичным развитием и распространением компьютерных систем в различных сферах деятельности человека актуальным является снижение затрат на подготовку новых пользователей. Перспективным в этом случае представляется использование привычного для пользователя естественного языка для организации диалога с компьютерной системой. Такая возможность реализуется средствами естественно-языкового пользовательского интерфейса, обладающего рядом преимуществ: минимальной подготовкой пользователя для работы с системой, простотой и высокой скоростью задания произвольных запросов к пользовательскому интерфейсу и высоким уровнем модели предметной области. Использование речевого синтеза при этом позволяет снизить нагрузку на пользователя по восприятию результатов обработки не через графический интерфейс (зрительную систему), а

посредством речи. Кроме того, языковой ввод команд и вопросов позволяет пользователю использовать эту же систему в качестве справочной и получать от неё помощь в исполнении любой деятельности (получение справки при выполнении технических работ и т.д.).

Следует отметить, что для естественно-языкового пользовательского интерфейса вопросно-ответных систем возможно использование ограниченного набора лексики и грамматики языка без серьезного 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, т.к. обладает сокращенным набором лексики и грамматики. Это позволяет снизить время обработки естественно-языковых конструкций в вопросно-ответной

системе, а также частично избежать лингвистических неоднозначностей.

Целью данной работы является создание семантической технологии компонентного проектирования естественно-языковых интерфейсов интеллектуальных вопросно-ответных систем. Такая технология основывается на семантической технологии компонентного проектирования пользовательских интерфейсов [Ostis, 2012], разрабатываемой в рамках открытого проекта OSTIS. В соответствии с указанной технологией пользовательский естественно-языковой интерфейс представляет собой набор взаимодействующих между собой модулей. Такие модули будем называть ip-компонентами, продуктами интеллектуальной собственности. Все модули разрабатываемого естественно-языкового интерфейса располагаются в библиотеке совместимых ip-компонентов. Такая библиотека осуществляет хранение ip-компонентов, предоставляет разработчику доступ к информации о хранимых компонентах, а также средства поиска нужных компонентов.

К ip-компонентам могут относиться также сторонние разработки и системы, например, системы распознавания и синтеза речи. В соответствии с этим в число задач данной работы включена также разработка методики интеграции сторонних продуктов в качестве независимых ip-компонентов. Данная задача подразумевает разработку универсальных и специализированных языков общения для всех ip-компонентов. Одним из таких базовых языков является семантический язык вопросов.

1. Исследование естественно-языковых интерфейсов

Использование естественно языка для организации диалога пользователя с компьютерной системой в общем случае сопряжено с рядом проблем: неоднозначность естественно языка, несоответствие возможностей реализации естественно-языкового интерфейса ожиданиям пользователя и др.

Как указывалось выше, для естественно-языкового пользовательского интерфейса вопросно-ответных систем возможно использование ограниченного естественно языка без серьёзного ущерба их функциональности. Благодаря использованию продуманных лексических и грамматических ограничений, принятых в рамках ограниченного естественно языка, появляется возможность избежать многих проблем, связанных с анализом неограниченного естественно языка. При этом язык общения пользователя с компьютерной системой по-прежнему остаётся вполне естественным, а процесс реализации этого общения становится более управляемым и в значительной степени упрощается.

Пользователь, впервые столкнувшись с естественно-языковым интерфейсом, может иметь

завышенные или заниженные ожидания от такой формы взаимодействия с компьютерной системой. Сравнительный анализ типов пользовательских интерфейсов (основанных на формах, с формальным языком запросов, графические пользовательские интерфейсы и пр.) показывает, что при построении пользовательских интерфейсов с использованием естественно языка у разработчиков преобладает желание максимально приблизить интерфейс к потребностям неподготовленного пользователя. Это несколько поднимает планку требований к степени дружелюбности и надёжности (безотказности) естественно-языковых пользовательских интерфейсов, поскольку пользователь, впервые столкнувшись с системой, "понимающей" естественный язык, слабо представляет реальные возможности системы. При этом его ожидания к степени понимания естественно языка могут отличаться от реальных способностей системы в обе стороны. Пользователь может спрашивать систему о том, чего она "не знает", а может "по привычке" использовать простейшие формулировки запросов.

Для решения обозначенной выше проблемы на начальных этапах развития естественно-языкового интерфейса может быть использована специально предусмотренная обратная связь с пользователем. В рамках такой обратной связи пользователю, задающему вопрос системе, предоставляется возможность видеть результат разбора этого запроса и его представление на формальном языке. Таким образом, пользователь, методом «проб и ошибок», во время работы с вопросно-ответной системой, может приспособиться (научиться) более эффективно использовать все возможности естественно-языкового интерфейса. В качестве целевого формального семантического языка используется специализированный предметно независимый язык вопросов. Указанный язык вопросов обладает большей семантической мощностью по сравнению с языками запросов к базам данных и знаний, что упрощает анализ естественно-языковых пользовательских запросов.

Семантическая (смысловая) неоднозначность естественно-языкового диалога может разрешаться за счет того, что предметная и лингвистическая базы знаний создаются в рамках единого информационного пространства, единой предметной области. У каждого понятия и отношения в этом случае в предметной базе знаний имеются соответствующие естественно-языковые идентификаторы (слова естественно языка), и это составляет первичную связь предметных и лингвистических знаний. Построение более развернутых связей является задачей разработчика лингвистической базы знаний конкретной вопросно-ответной системы для конкретной предметной области.

2. Моделирование естественно-языкового интерфейса

Естественно-языковой пользовательский интерфейс в рамках разрабатываемой технологии компонентного проектирования рассматривается нами как специализированная интеллектуальная система, обеспечивающая диалог между прикладной вопросно-ответной системой и пользователем в заданной предметной области. Таким образом, будучи интеллектуальной системой, естественно-языковой интерфейс, как и любая другая подобная система, в качестве основных своих компонентов имеет базу знаний, машину обработки знаний и пользовательский интерфейс.

Определим здесь следующие основные функциональные возможности естественно-языкового интерфейса:

- возможность ввода сообщения в естественно-языковой форме посредством текста или речи;
- трансляция сообщения пользователя на внутренний язык интеллектуальной системы;
- трансляция ответа системы в текст естественного языка в форме текста или речи.

Указанный функционал определяет структуру интеллектуальной системы естественно-языкового интерфейса, которая включает следующие составляющие:

- пользовательский интерфейс, посредством которого происходит ввод сообщений пользователя и вывод ответа системы пользователю;
- трансляторы естественно-языковых запросов на внутренний sc-язык вопросов;
- трансляторы конструкций sc-языка (sc-конструкций) на естественный язык.

Общая структура интеллектуальной системы (ИС) естественно-языкового пользовательского интерфейса вопросно-ответных систем представлена на рисунке 1.



Рисунок 1 - Структура ИС естественно-языкового пользовательского интерфейса

2.1. База знаний естественно-языкового интерфейса

База знаний ИС естественно-языкового пользовательского интерфейса подразделяется на лингвистическую и предметную базы знаний. Предметная база знаний используется вопросно-ответной системой для поиска и генерации ответов по запросам пользователя. Кроме того, естественно-языковой интерфейс использует эту базу знаний для решения ряда задач, связанных с разрешением неоднозначности естественного языка, т.к. предметная база знаний однозначно задает контекст диалога пользователя с вопросно-ответной системой [Сулейманов, 2011].

Структура базы знаний естественно-языкового интерфейса представлена на рисунке 2.



Рисунок 2. Структура базы знаний естественно-языкового интерфейса

Лингвистическая база знаний содержит формальное описание используемого естественного языка, привязку лексики к предметной базе знаний, спецификации семантических языков.

Для описания знаний о каждом из уровней естественного языка (см. ниже описание структуры базы знаний) используется специализированный семантический язык представления лингвистических знаний. Спецификация данного языка также входит в состав лингвистической базы знаний. На практике лингвистическая база знаний может не включать все уровни описания естественного языка, т.к. использование ограниченного языка для общения пользователя с системой может дать более качественный результат, нежели использование неограниченного естественного языка [Byron Long, 1994].

2.1.1. Лингвистическая база знаний естественного языка

Лингвистическая база знаний естественного

языка имеет следующую логическую структуру, соответствующую уровням его формального описания:

- лексика и фразеология – включает описание лексического и фразеологического состава русского языка, а также некоторые закономерности словообразования;

- фонетика – включает описание звукового состава современного литературного русского языка и основные звуковые процессы, протекающие в языке;

- орфоэпия – включает описание норм современного русского литературного произношения;

- графика – описание русского алфавита, а также соответствий букв звукам;

- орфография – включает описание основных принципов русского написания – морфологические, фонетические, традиционные;

- морфемика и словообразование – содержит описание морфемного состава отдельных слов и основных способов образования новых слов, в том числе:

- морфологический;
- морфолого-синтаксический;
- лексико-семантический;
- лексико-синтаксический;
- грамматика – содержит описание правил морфологии и синтаксиса:
 - морфология – описание грамматических категорий и грамматических форм слов;
 - синтаксис – описание основных синтаксических единиц: словосочетания и предложения, виды синтаксических связей, типы предложений и их структуры;
 - пунктуация – включает описание совокупности правил расстановки знаков препинания;
 - текст, признаки, характеристики – описание совокупности правил построения текстов и их характеристики.

Как указывалось выше, для описания знаний каждого из перечисленных уровней используется специализированный формальный семантический язык представления лингвистических знаний, спецификация которого входит в состав лингвистической базы знаний.

2.1.2. Грамматический словарь русского языка

Грамматический словарь русского языка является важным компонентом базы знаний естественно-языкового интерфейса. Грамматические знания о естественном языке, записанные в лингвистической базе знаний, используются для грамматического анализа естественно-языкового текста вопроса, а также для синтеза ответов пользователю. Эта часть лингвистической базы знаний выделяется в отдельный *ip*-компонент и может использоваться в качестве предметной базы знаний по грамматике русского языка. Запись морфологической

информации о конкретном слове в лингвистической базе знаний осуществляется так, как показано на рисунке 3.

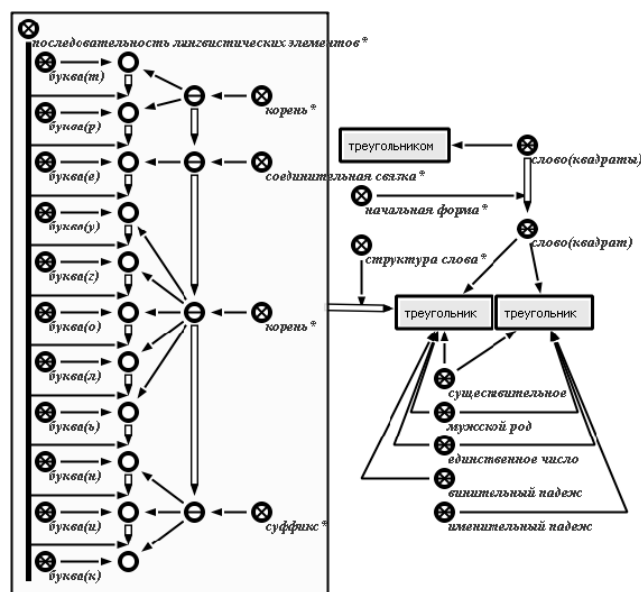


Рисунок 3. Запись морфологической информации в лингвистической базе знаний

При описании морфологической информации каждому слову ставятся в соответствие грамматические категории (число, род, склонение и т.д.). Для этого данное слово включается во множество конкретной грамматической категории (например, как показано на рис.3, единственное число, мужской род и т.д.). Каждое конкретное слово включено во множество его грамматических синонимов. Это позволяет быстро получать доступ ко всем наборам грамматических категорий при грамматическом анализе. Для каждого слова строится его разбиение на морфемы, а также эквивалентная запись в семантической сети.

На множестве слов задаются отношения синонимии, антонимии и пр. Это может использоваться при семантическом анализе естественно-языкового текста.

2.2. Машина обработки знаний

Машина обработки знаний естественно-языкового интерфейса включает в себя операции, обеспечивающие обработку различных лингвистических конструкций, перевод внешних языков на семантические языки интеллектуальной системы и обратно.

Все компоненты машины обработки знаний естественно-языкового интерфейса можно разделить на трансляторы и анализаторы. Задачей трансляторов является перевод знаний из одного языка представления знаний в другой, например, трансляция фактографических знаний по предметной области в текст на естественном языке. Задачей анализаторов является анализ фрагментов знаний и выявление ранее неизвестных фактов, например, анализ запроса пользователя,

направленный на поиск цели и задачи вопроса.

Трансляторы естественно-языкового интерфейса в минимальной конфигурации должны включать в себя транслятор ограниченного естественного языка на язык вопросов и транслятор фактографических знаний на ограниченный естественный язык.

Трансляторы естественно-языкового пользовательского интерфейса осуществляют обработку информационных структур, полагаясь исключительно на описание синтаксиса используемого языка, а также на описание семантики предметной области. Описание синтаксиса используемого языка хранится в лингвистической и предметной базах знаний. В лингвистической базе знаний должны быть также описаны специфические для данной области знания о языке. Такой фрагмент базы знаний может представлять собой отдельный ip-компонент, т.к. он представляет ценность только в совокупности с предметной и лингвистической базой знаний.

Цикл работы естественно-языкового интерфейса начинается с ввода пользователем сообщения на естественном языке путём ввода текста или голосового (речевого) сообщения. По введённому в систему тексту 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 разрешать спорные моменты, связанные с использованием одних и тех же терминов в разных предметных областях.

Первым этапом анализа запроса пользователя является морфологический анализ. На данном этапе для каждой словоформы в формальной записи предложения строятся отношения, задающие соответствие значениям её грамматических категорий (род, падеж, склонение и т.д.). Грамматические категории могут присутствовать в базе знаний естественно-языкового интерфейса явно (словоформа присутствует в словаре с описанием её грамматических категорий), либо грамматическая категория и её значение могут выявляться на основании морфологических правил используемого естественного языка. Такой подход позволяет разработчику при проектировании естественно-языкового интерфейса выбирать между быстродействием и объёмом проектируемой базы знаний.

На следующем этапе – этапе синтаксического анализа – строятся отношения синтаксических связей внутри предложения, выделяются главные и второстепенные члены предложения, выявляется тип предложения и т.д. Синтаксический анализ выполняется поэтапно, “наращивая” формальное описание структуры предложения с использованием информации, полученной на этапе морфологического анализа. На данном этапе используются лексические и синтаксические правила анализируемого языка. В общем случае, такие правила описывают критерии существования

синтаксических связей в предложении, при анализе предложения проверяется соответствие таким критериям и в зависимости от результата создаются либо удаляются соответствующие связи в предложении. Пример записи правил синтаксического разбора предложений приведен на рисунке 4.

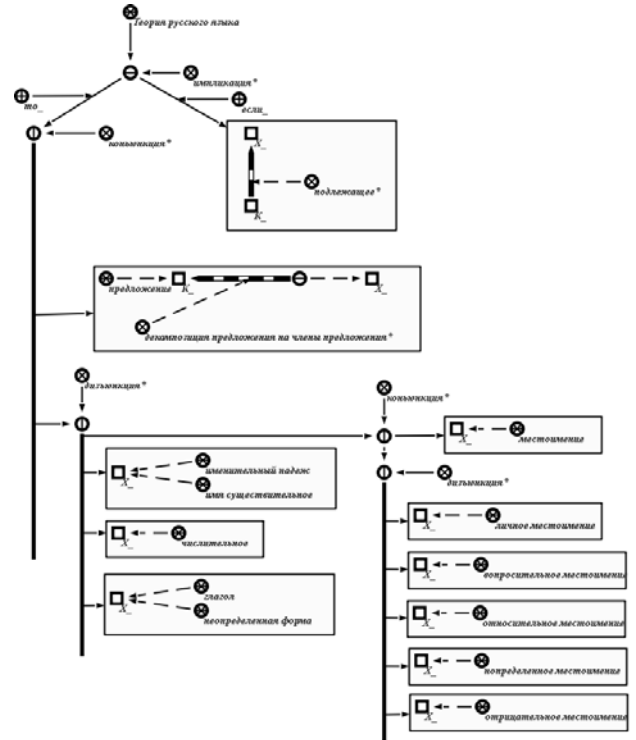


Рисунок 4. Запись синтаксического правила в лингвистической базе знаний

Результатом этапа синтаксического анализа является формальное описание синтаксического разбора предложения в базе знаний естественно-языкового интерфейса. На следующем этапе – семантического анализа – строятся отношения эквивалентности элементов предложения и узлов предметной области в памяти системы.

Этап семантического анализа является наиболее сложным этапом обработки естественно-языкового текста. В рамках используемой технологии семантический анализ представляет собой обработку семантической сети, отражающей результат анализа естественно-языкового текста на всех предыдущих этапах, а также присутствующих в системе знаний о языке и предметной области основной системы. На этом этапе происходит анализ соотношения лингвистической конструкции и конструкций, хранящихся в памяти системы для выявления отношений соответствия, эквивалентности и т.д.

Результатом семантического анализа является конструкция запроса к системе. Сложность получения такого результата связана с возможной семантической неполнотой исходного естественно-языкового вопроса, что тем не менее вполне естественно для предложений естественного языка. Например, вопрос: «Какими свойствами обладают

биссектрисы треугольников?» в семантически полном (развернутом) полужформальном виде будет выглядеть следующим образом: «Какие утверждения являются общими для понятия треугольник и понятия биссектриса?». Из примера видно, что в процессе развертки произошли следующие подстановки: высказывание «*обладать свойствами*» было развернуто в «*иметь общие утверждения*», «*свойства*» – в «*утверждения*», «*биссектрисы*» – в «*понятие биссектриса*», «*треугольники*» – в «*понятие треугольник*». Ответ на данный вопрос будет следующим: «У любого треугольника все биссектрисы пересекаются в одной точке»

Вопросы, сгенерированные естественно-языковым интерфейсом, обрабатывает универсальный решатель вопросно-ответных систем, являющийся частью вопросно-ответной системы по соответствующей предметной области. Стоит заметить, что универсальный решатель использует знания и лингвистической базы знаний. Таким образом система может отвечать на вопросы, связанные с используемым естественным языком.

Полученный от предметной справочной системы ответ транслируется в текст на ограниченном естественном языке, который в результате выводится пользователю [Карпилович, 1997] [Попов, 1982].

3. Библиотека совместимых ip-компонентов естественно-языкового пользовательского интерфейса

Главным элементом семантической технологии компонентного проектирования естественно-языковых пользовательских интерфейсов является библиотека совместимых ip-компонентов. Такая библиотека включает в себя лингвистические базы знаний по разным языкам, различные трансляторы и анализаторы естественных текстов, элементы пользовательского интерфейса. Это позволяет проектировать естественно-языковые интерфейсы, комбинируя уже существующие компоненты, выбирая нужные лингвистические базы знаний, анализаторы и синтезаторы. Задачей разработчика в данном случае является привязка естественно-языкового интерфейса к предметной области интеллектуальной системы [Мельчук, 1974]. Для этого необходимо добавить лингвистические знания о специфических для этой предметной области понятиях.

3.1. Схема взаимодействия компонентов ея-интерфейса

Если в системе используется только один естественный язык, то работу компонентов можно представить в виде последовательной обработки входящего сообщения. Если же в системе используются несколько языков или несколько компонентов, выполняющих одни и те же функции, процесс работы естественно-языкового интерфейса

начинает ветвиться.

В разрабатываемый прототип добавлена поддержка белорусского языка. По ряду технических причин существующий анализатор входящих сообщений не имел возможности обрабатывать тексты на белорусском языке. Поэтому в систему был добавлен простой, но универсальный компонент обработки входящих сообщений. Новый компонент может обрабатывать как белорусскоязычный, так и русскоязычный тексты на основе анализа состава и порядка слов в сообщении. В проведенных экспериментах было рассмотрено поведение системы в присутствии двух компонентов схожего функционала.

Цикл работы системы представлен на рисунке 4.



Рисунок 4. Схема работы компонентов ея-интерфейса с несколькими трансляторами

Из рисунка 4 видно, что пока один из трансляторов не создаст в системе рему и тему заданного пользователем вопроса, трансляция самого вопроса не будет произведена. Таким образом пользователь увидит ответ на свой вопрос только тогда, когда предметная интеллектуальная вопросно-ответная система не обработает входящий запрос, что произойдет только после того как один из анализаторов и трансляторов его не сгенерируют. Можно сказать, что в системе происходит свободная конкуренция функционально подобных компонентов. Те компоненты, которые «не успели» дать результат, при его появлении в памяти системы завершают свою текущую задачу.

3.2. Речевые компоненты

Сегодняшний информационный мир

предоставляет огромные возможности. Тем не менее, взаимодействие между человеком и компьютерным устройством по-прежнему базируется в основном на различных формах графического интерфейса. Но для повсеместной интеграции в нашу повседневную жизнь необходимы новые виды взаимодействия.

Если учесть, что после зрения вторым по объёму принимаемой информации органом чувств является слух, то вполне разумно выглядит использование речевой коммуникации в качестве дополнения к графическим интерфейсам. В сфере разработки альтернативных способов взаимодействия речевая коммуникация играла главенствующую роль на протяжении последних десятилетий. Идея дополнить традиционные устройства ввода голосовым контролем и устройства вывода – речевым синтезом, повысив тем самым комфорт и скорость ввода и вывода, кажется очень привлекательной. Однако до последнего времени самыми распространёнными примерами использования звука в интерфейсе можно считать озвучивание некоторых операций с использованием слов либо звуков из стандартного набора. В настоящее время наблюдается тенденция к ускорению прогресса в этой сфере по следующим причинам. Во-первых, речевые технологии достигли нового уровня развития, в основном за счёт использования метода стохастического моделирования. Во-вторых, по сравнению с другими альтернативными видами взаимодействия человека с компьютером, речь стала приоритетной.

Таким образом, очевидным выглядит факт, что после интерфейса командной строки (1960 – 1980-х гг.), и графического интерфейса (1980 – 2000-е гг.), будущее принадлежит комплексному пользовательскому интерфейсу, позволяющему задействовать, кроме зрения разные органы чувств человека, в первую очередь – его слух [Лобанов, 2006]. Многие факты говорят в пользу этого утверждения:

- речь – наиболее популярная форма коммуникации между людьми;
- нет необходимости в непосредственном контакте при взаимодействии, поскольку микрофон и динамики могут располагаться на расстоянии;
- руки и глаза остаются свободными, что делает голосовой интерфейс приоритетным в некоторых ситуациях, таких как, например, процесс вождения транспортного средства либо параллельного использования нескольких приложений одновременно;
- на данный момент в мире существует порядка 1,3 миллиарда мобильных телефонов, что в пять раз превышает количество компьютеров, подключённых к интернету. Это обеспечивает огромный рынок для будущих систем автоматического диалога.

В свете всех перечисленных тенденций вполне закономерным и актуальным выглядит разработка речевых способов управления и для поддержки

средств навигации и поиска в семантических сетях. Как правило, от таких приборов требуется, с одной стороны, обеспечивать обработку большого объёма запросов, а с другой – в процессе использования как можно меньше отвлекать пользователя от объекта поиска. При этом неизбежно растёт число элементов графического меню, что в свою очередь замедляет время поиска нужной опции и заставляет отвлечься от искомого объекта [Landauer et al., 1985].

Речевой способ взаимодействия – наиболее естественный интерфейс для общения человека с человеком. Это влечет за собой простоту изучения и использования речевого интерфейса при взаимодействии с базами знаний. Обсуждаемый вид взаимодействия позволяет освободить руки и глаза человека при подаче команд. Пользователь может осуществлять запросы (устно отдавая соответствующие команды) в процессе работы, передвижения или манипуляции другими объектами. Дополнительный комфорт интерфейса становится следствием отсутствия необходимости в специальных устройствах, таких как мышь, палочка или перчатки данных. Таким образом, широкому кругу пользователей, включая пожилых людей и инвалидов, удобно приспосабливаться к речевому интерфейсу.

3.3. Разработка компонента речевого ввода информации на основе пакета НТК

Как уже было сказано, небезосновательным выглядит факт, что после интерфейса командной строки и графического интерфейса будущее принадлежит комплексному пользовательскому интерфейсу, позволяющему задействовать, кроме зрения, разные органы чувств человека, в первую очередь – его слух.

В свете всех перечисленных тенденций, вполне закономерным и актуальным выглядит разработка речевых способов управления для поддержки средств навигации и поиска в семантических сетях. Как правило, от таких приборов требуется, с одной стороны, обеспечивать обработку большого объёма запросов, а с другой – в процессе использования как можно меньше отвлекать пользователя от объекта поиска. При этом неизбежно растёт число элементов графического меню, что в свою очередь замедляет время поиска нужной опции и заставляет отвлечься от искомого объекта [Landauer et al., 1985].

Одной из важнейших задач данной работы является выбор подходящей технологии и методики для создания модуля распознавания слитной русской и белорусской речи, призванного обеспечить использование голосового ввода для реализации запросов к базам знаний. При этом акцент ставится на разработку именно относительно автономного компонента, который мог бы быть использован разработчиком, не имеющим высокой квалификации в сфере обработки сигналов или теории Скрытых Марковских Моделей (СММ). Такая подсистема должна стать частью целой технологии проектирования интеллектуальных

систем, разрабатываемой в рамках открытого проекта OSTIS.

Второй отличительной чертой является возможность распознавания белорусской речи, что является весьма актуальной проблемой из-за малого количества аналогов. Причин того, что так немного внимания уделяется включению белорусского языка в интерфейсы различного рода цифровых устройств разные. Однако мы не сомневаемся в необходимости развития белорусскоязычных способов взаимодействия и делаем конкретные шаги в этом направлении.

Теория СММ была выбрана в качестве методологической основы для создания модуля распознавания речевых запросов. В качестве набора инструментов, реализующего все основные функции и алгоритмы, был использован пакет НТК (<http://htk.eng.cam.ac.uk/> - The Hidden Markov Model Toolkit)

3.3.1. Тестовые диалоги для интерфейса семантических баз знаний

При создании прототипа речевого интерфейса для осуществления различного рода запросов в семантических базах знаний были выбраны ряд тестовых диалогов пользователя с системой с целью выбора необходимых в общении слов и фраз для последующего создания СММ с соответствующими параметрами. Примером такого рода диалогов могут служить следующие ситуации:

Русский:

Вопрос: Что это такое (В окне геометрического редактора выделяется некоторая фигура)?

Ответ: Это треугольник со сторонами а, b и углом С, равным 45 градусам, между ними.

Вопрос: Как они связаны (В окне системы выделяются понятия треугольника и тригонометрии)?

Ответ: Эти два понятия связаны в теореме синусов и теореме косинусов.

Вопрос: Что из этого следует?

Ответ: Это дает возможность расчета численных характеристик конкретного треугольника.

Белорусский:

Вопрос: Як выглядае трохвугольнік?

Вопрос: Вывесці трансляцыю паняцця трохвугольнік.

Вопрос: Якое паняцце з'яўляецца надмноствам паняцця трохвугольнік?

И т. д.

Стоит отметить, что фразы выбирались по

следующему критерию: слова, входящие в запросы, должны состоять по возможности из большого количества звуков, чтобы повысить качество распознавания.

3.3.2. Выбор способа подготовки данных для создания моделей

В качестве вариантов, как для русскоязычного, так и для белорусскоязычного распознавателей рассматривались системы со следующими характеристиками: монофонные без разметки обучающих данных, монофонные с разметкой обучающих данных и системы на основе связанных трифонов, где обучение проводилось без разметки обучающих данных. Критерием для сравнения выступали среднее время распознавания на один фрейм, точность распознавания по фразам, а также точность распознавания по отдельным словам. Обучение осуществлялось на одинаковом ограниченном наборе из 50 фраз на русском языке и 73 фраз для белорусского аналога, содержащих в сумме 217 и 320 слов соответственно. Результаты представлены в сводной таблице 1 для обоих языков.

Таблица 1 - Оценка характеристик систем распознавания

Русский язык		
Среднее время распознавания на один фрейм, с/фрейм	Точность распознавания фраз, %	Точность распознавания слов, %
<i>Система на основе монофонов. Обучение без разметки данных</i>		
0.011396	74.00	94.97
<i>Система на основе связанных трифонов. Обучение без разметки данных</i>		
0.011802	72.00	93.53
<i>Система на основе монофонов. Обучение с разметкой данных</i>		
0.011626	98.00	99.57
Белорусский язык		
<i>Система на основе монофонов. Обучение без разметки данных</i>		
0.011567	77.00	95.60
<i>Система на основе связанных трифонов. Обучение без разметки данных</i>		
0.011912	74.00	94.40
<i>Система на основе монофонов. Обучение с разметкой данных</i>		
0.011803	98.00	99.65

На основе выше приведённой информации выбор был сделан в пользу системы на основе монофонов, для создания которой использовались данные, размеченные вручную. Решающим фактором стал высокий процент точности распознавания по фразам при приемлемом времени распознавания. Такие результаты можно объяснить следующим образом: исходя из условия ограниченности словаря и возможных последовательностей слов, целесообразным выглядит разметка ограниченного количества обучающих файлов вручную, с другой стороны. Это же ограничение препятствует созданию качественных трифонов, тем более связанных.

3.3.3. Алгоритм создания системы СММ для распознавания

Полный цикл создания хорошо обученных СММ включает два основных этапа: подготовка данных для обучения и непосредственно самообучение (Рисунок. 5).

Подготовка данных.

Шаг 1 -- Грамматика. Первым шагом явилось создание грамматики каждого из запросов. Грамматика в данном случае представляет собой сеть, которая включает строгие последовательности слов, допустимых для распознавания.

Шаг 2 -- Словарь. Спускаясь ниже по иерархии от фраз к отдельным словам, был создан упорядоченный словарь слов, входящих в запросы. Закрытый словарь предоставляет возможность создания уникальной фонетической транскрипции, адаптированной для конкретного диктора и учитывающей региональные особенности произношения некоторых слов русского и белорусского языков. Особенно это касается специфических звуков, характерных исключительно для белорусского языка: «дз», «дж», «ць» и т. д.

Шаг 3 -- запись данных. На данном этапе осуществляется создание набора файлов формата wav, содержащих несколько вариантов (не менее трёх) записей базовых слов, произнесённых диктором. В качестве инструмента использована функция HSLab из пакета НТК, которая позволяет не только записывать данные, но и размечать их по содержащимся фонемам. Всего для обучения использовалось 113 фраз, состоящих в сумме из 613 слов.

Шаг 4 -- кодирование данных. Финальным шагом в подготовке данных является обработка речевых сигналов и преобразование их в последовательности векторов признаков. В данной работе в качестве таких векторов были использованы кепстральные коэффициенты шкалы мел-частот [Oppenheim et al., 2004] [Опенгейм и др., 1979].

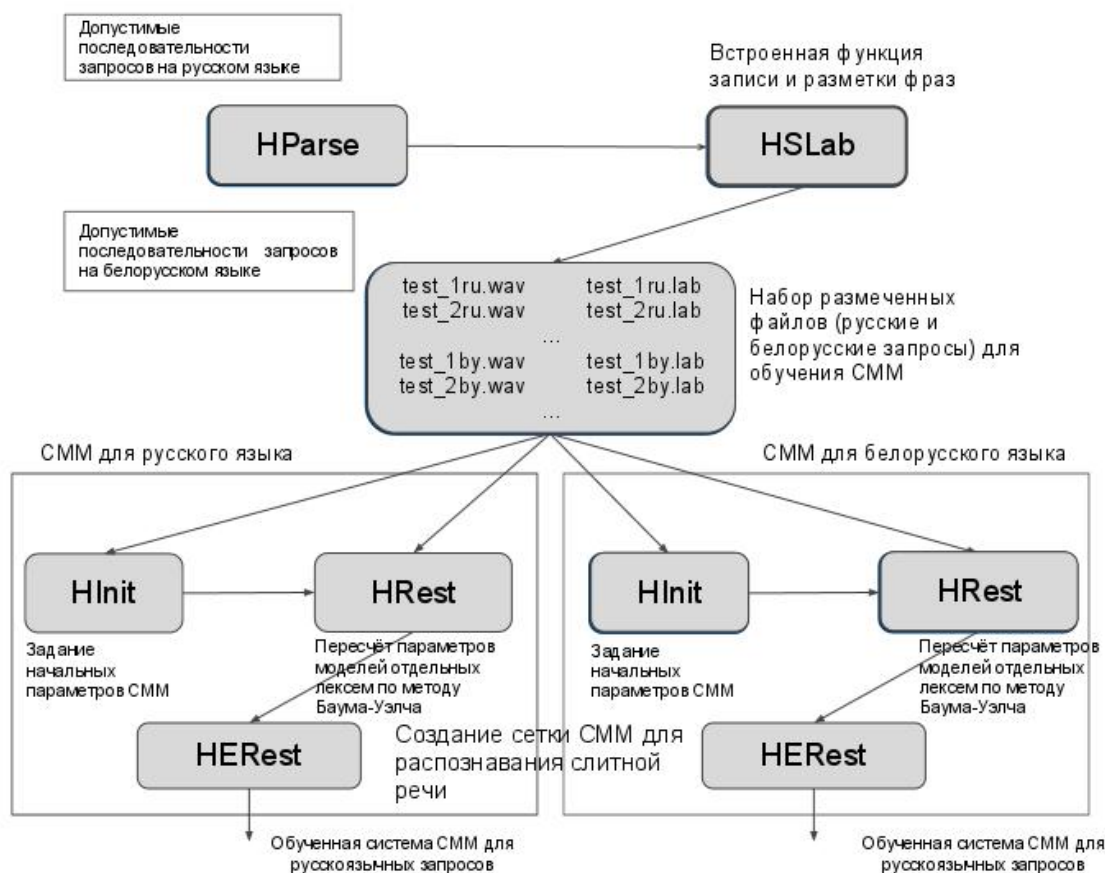


Рисунок 5. Алгоритм создания набора СММ для распознавания запросов на русском и на белорусском языках

Обучение

С этого момента начинается создание набора обученных СММ, эмиссионные вероятности в которых описываются смесью плотностей распределения Гауссовых случайных величин [Rabiner, 1988].

Шаг 5 -- Создание начальных монофонов. Первым шагом в создании системы СММ является определение модели прототипа. Для системы, основанной на фонемах, хорошей топологией является лево-правая схема с тремя состояниями [Бовбель и др., 2008]. Начальные значения для модели каждой фонемы рассчитывались на основании обучающих данных с помощью алгоритма Витерби [Бовбель и др., 1998]. Программным средством здесь выступила функция HInit.

Шаг 7 -- Пересчёт коэффициентов монофонов. Параметры модели каждой фонемы пересчитываются по методу Баума-Уэлча (алгоритм прямого-обратного хода или алгоритм максимизации правдоподобия) [Бовбель и др., 2008]. Пересчёт производится с помощью функции HRest.

Шаг 8 -- Создание системы СММ для распознавания фраз. Последним шагом является корректировка параметров моделей фонем, но уже не по отдельности, а в связке друг с другом в контексте как отдельных слов, так и целых предложений. Такая интегрированная система создаётся функцией HERest.

3.3.4. Ключевые особенности компонента речевого ввода

В заключение отметим, что результатом работы стало решение конкретной задачи, а именно создания модуля распознавания речи для осуществления голосовых запросов к интеллектуальной базе знаний. Цель была достигнута, что стало следствием решения следующих подзадач:

- выбран способ реализации системы с учетом требований к точности и времени распознавания;
 - создан прототип модуля, который способен с высокой точностью распознавать широкий спектр запросов к базе знаний по геометрии. При этом в модуле реализована двуязычность, позволяющая делать устные запросы как на русском, так и на белорусском языках;
 - выработана методика создания модуля распознавания как для русскоязычных, так и для белорусскоязычных фраз, которая может быть использована разработчиком, который не является специалистом в сфере речевых технологий, для создания модуля распознавания запросов к базам знаний в любых других сферах помимо геометрии.
- Кроме того, стоит отметить, что благодаря большому проценту протяжных гласных в белорусском языке, потенциально системы распознавания, созданные для фраз белорусского

языка, обладают большей точностью распознавания при прочих равных условиях.

3.4. Разработка компонента речевого вывода информации на основе синтезатора русского и белорусского языков

Для перевода текстовой информации в речевую из вопросно-ответной системы используется два ip-компонента, основанных на синтезаторах речи по тексту на белорусской [Гецевич и др., 2010] и русской речи [Лобанов и др., 2008]. Данные ip-компоненты реализованы в качестве сторонних подключаемых модулей. В результате такого подключения к системе синтезаторов речи по тексту пользователь получает озвученный ответ от системы. Таким образом, система естественно-языкового интерфейса становится еще более «естественной» для пользователя.

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 5. Синтезатор включает модули: текстовый процессор, просодический процессор текста и сигнала, и фонетический процессор. Каждый из этих модулей поддерживается наборами соответствующих баз данных и правил.



Рисунок 5. Структура системы синтеза речи по тексту

Входной орфографический текст подвергается ряду последовательных обработок в каждом из процессоров. Текстовый процессор обрабатывает входной орфографический текст в следующей последовательности: очистка текста, преобразование знаков (аббревиатур, сокращений, чисел и др.), расстановка словесных ударений и грамматических признаков словоформ.

Преобразованный текст поступает на входы просодического, а затем фонетического

процессоров. В результате работы просодического подпроцессора для текста, текст разделяется на синтагмы, акцентные единицы, далее он разделяется на элементы акцентных единиц: интонационное предъядро, ядро и заядро. И, наконец, просодический подпроцессор для языкового сигнала устанавливает в соответствии с базой данных просодических "портретов" синтагм значимости амплитуды, продолжительности фонем и частоту основного тона для каждого элемента акцентных единиц.

Акустический процессор на основе информации о том, какие необходимо синтезировать аллофоны, а также какие просодические характеристики должны быть приписаны каждому аллофону, генерирует речевой сигнал путем компиляции отрезков натуральных звуковых волн соответствующих аллофонов и мультифонов. Таким образом, входной текст преобразуется в речевой сигнал.

Главный модуль системы синтеза речи по тексту управляет работой всех остальных модулей, определением списка и загрузкой плагинов к остальным модулям, передачей данных между модулями.

Процессор нормализации текста перед лингвистической обработкой производит удаление из текста символов, ненужных для синтеза речи, удаляет случайное дублирование знаков препинания, заменяет похожие символы на один из них. Подключаемые плагины производят дополнительное преобразование поступающих данных перед подачей их на нормализацию, так может происходить преобразование PDF в простой текст.

Главный модуль лингвистического процессора 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). 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 лексико-грамматических характеристик этих слов, чтобы словосочетание не было правильно разбито на синтагмы. Процессор создания синтагм разделяет слова и 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.

ЗАКЛЮЧЕНИЕ

Семантическая технология компонентного проектирования естественно-языковых интерфейсов вопросно-ответных систем предоставляет единую основу для анализа естественно-языковых текстов, начиная с ввода текста, морфологического, синтаксического, семантического анализа и генерации текста. Кроме того, указанная технология позволяет устанавливать отношения между эквивалентными лингвистическими конструкциями и структурой внутренней памяти системы.

Библиотека совместимых ip-компонентов естественно-языкового пользовательского интерфейса позволяет разработчику проектировать естественно-языковой интерфейс посредством компоновки уже готовых компонентов, включая лингвистические базы знаний, трансляторы и элементы пользовательского интерфейса. Ip-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 естественно-языковой интерфейс еще более естественным для пользователя.

Возможность интеграции сторонних разработок и проектов в качестве внешних ip-компонентов позволяет производить интеграцию различных подходов и методов в рамках одного проекта, что

позволяет эффективно использовать их лучшие стороны.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Ostis, 2012] Open Semantic Technology for Intelligent Systems. [Электронный ресурс]. – 2012. – Режим доступа: <http://www.ostis.net/>. – Дата доступа: 01.02.2012
- [Сулейманов, 2011] Сулейманов Д.Ш. Двухуровневый лингвистический процессор ответных текстов на естественном языке / Сулейманов Д.Ш. // сборник трудов Международной научно-технической конференции OSTIS-2011, Минск, 2011 / БГУИР – Минск, 2011
- [Byron Long, 1994] Byron Long, Natural Language as an Interface Style / Byron Long // Dynamic Graphics Project Department of Computer Science University of Toronto, 1994.
- [Апресян, 1995] Апресян, Ю. Д. Избранные труды, том II. Интегральное описание языка и системная лексикография / Ю. Д. Апресян – Москва: Школа «Языки русской культуры», 1995.
- [Карпилович, 1997] Алгоритмы порождения предложений естественного языка (обзор и анализ) / Т.П. Карпилович. – Минск, 1977.
- [Попов, 1982] Попов, Э.В. Общение с ЭВМ на естественном языке / Э.В. Попов. – Москва: Наука, 1982.
- [Мельчук, 1974] Мельчук, И.А. Опыт теории лингвистических моделей «Смысл-Текст». Семантика, Синтаксис / И.А. Мельчук. – Москва, 1974.
- [Лобанов, 2006] Лобанов, Б. М. Речевой интерфейс интеллектуальных систем: учебное пособие / Б. М. Лобанов [и др.]. – Минск: БГУИР, 2006.
- [Landauer et al., 1985] Landauer T.K. Selection from alphabetic and numeric menu trees using a touch screen: Breadth, depth, and width / T.K. Landauer, D.W. Nachbar. New York : ACM, 1985.
- [Бовбель и др., 1998] Бовбель Е. И. Статистические методы распознавания речи: скрытые марковские модели / Е. И. Бовбель, И. Э. Хейдоров // Зарубежная радиоэлектроника. Успехи современной электроники. 1998. № 3. С. 36--54.
- [Rabiner, 1988] Rabiner L. A. Tutorial on Hidden Markov Models and Selected Applications in Speech / L. A. Rabiner // Recognition. IEEE Press, 1988. pp. 257—286.
- [Oppenheim et al., 2004] Oppenheim, A. V. From Frequency to Quefrancy: A History of the Cepstrum / A. V. Oppenheim, R. W. Schafer // IEEE Signal Processing Magazine. 2004. Vol. 21. P. 95—106.
- [Бовбель и др., 2008] Бовбель Е. И. Скрытые марковские модели и машины на опорных векторах от теории к практике / Е. И. Бовбель, И. Э. Хейдоров, Ю. В. Пачковский. Минск БГУ, 2008.
- [Опенгейм и др., 1979] Опенгейм, А. В. Цифровая обработка сигналов / А. В. Опенгейм, Р. В. Шафер. М., 1979.
- [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., Цирульник Л.И. Минск: Белорусская наука, 2008. – 344 с.: ил.

SEMANTIC TECHNOLOGY DESIGN NL INTERFACES FOR QUESTION ANSWERING SYSTEMS

Y.S. Hetsevich*, S.A. Hetsevich*, O.E. Yeliseyeva**,
V.A. Zhitko***, A.A. Kuzmin**,

* *United Institute of Informatics Problems of the
National Academy of Sciences of Belarus, Minsk,
Republic of Belarus*

Yury.Hetsevich@gmail.com, Novacimya@gmail.com

** *Belarusian State University, Minsk, Republic
of Belarus*

olae@open.by, kuzaleksaleks@gmail.com

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

zhitko.vladimir@gmail.com

INTRODUCTION

Description of technology to design a natural language interfaces for intelligent systems which is based on semantic networks (sc-technology) [Ostis, 2012], applying to a prototype of natural language interface for question answering intelligent system on the geometry is presented. Model includes speech-to-text and text-to-speech Belarusian and Russian subsystems, the generation of responses in the form of natural language and formal text. Also article include the description methods for linking various components of the system and description methods for development of new components.

MAIN PART

Given technology is based on semantic technology designing user interfaces for intelligent systems, which is part of OSTIS technology. It means that natural language user interface is also an intelligent system with its own knowledge base, knowledge production machine and user interface. But domain intelligent system and natural language interface aren't separated systems; they are integrated systems with shared memory and union knowledge bases.

Additional third-party component is natural Belarusian and Russian voice input. This component requires high performance, so it built in a traditional technology. But it doesn't mean that it not require semantic technology for improving functionality. Some data needed for recognizing human voice could be storage as knowledge in knowledge base or even generate from knowledge.

Another important third-party 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.

CONCLUSION

In the given paper short description of semantic technology to designing and prototyping natural language interfaces with voice input and output in Belarusian and Russian languages. Also described basics of designing linguistic knowledge bases for Russian language, it can be expand for various languages, which will be integrate in domain knowledge base in the future.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.82:004.55:007.52

ФАРМАЛІЗАЦЫЯ СЕМАНТЫКІ ПРАГМАТЫЧНЫХ ФРАЗЕМ БЕЛАРУСКАЙ МОВЫ

Кім Ю. А., Елісеева В. Я.

*Беларускі дзяржаўны ўніверсітэт,
г. Мінск, Рэспубліка Беларусь*

brikim@gmail.com

olac@open.by

У працы разглядаюцца прагматымы беларускай мовы і ажыццяўляецца апісанне семантыкі іх ужывання ў канкрэтных сітуацыях. Такія апісанні плануецца выкарыстоўваць у якасці асновы для стварэння базы ведаў прагматым беларускай мовы. База ведаў, у сваю чаргу, можа з'яўляцца ядром сістэм машыннага перакладу, семантычнага пошуку і вывучэння мовы, а таксама корпусаў тэкстаў.

Ключавыя словы: прагматыма, прагматычная фразема, сітуацыя ўжывання.

Уводзіны

У паўсядзённым жыцці, у працэсе камунікацыі ў межах канкрэтных сфер дзейнасці людзі выкарыстоўваюць дастаткова абмежаваны набор моўных сродкаў. Значную частку гэтага набору складаюць так званыя прагматымы – устойлівыя выразы ці часткі выказаў, якія ўжываюцца звычайна як адно цэлае ў пэўнай моўнай сітуацыі.

У працэсе камунікацыі чалавек пераўтварае наяўную інфармацыю ў моўныя выразы, якія выконваюць у дадзенай сітуацыі пэўную функцыю, дасягаюць пэўнай мэты. Дзякуючы такой уласцівасці, як сінанімічнасць моўных сродкаў, для адной і той жа функцыі можна пабудаваць больш за адзін эквівалентны сказ. У некаторых выпадках, хаця пабудаванне сінанімічнага сказа цалкам магчыма, выкарыстоўваецца амаль выключна адзін варыянт. Гэта можна назіраць, напрыклад, у вобласці фразеалогіі.

Вядома, што ў працэсе маўлення (сінтэза тэкстаў) чалавек мысліць не асобнымі словамі, а цэлымі выразамі. Менавіта гэтыя выразы з'яўляюцца асновай валодання мовай. У розных выпадках і сітуацыях чалавек выкарыстоўвае нейкую частку з агульнага мноства знаёмых і зразумелых для яго выказаў ці іх частак.

Мэтай дадзенай працы яўляецца вызначэнне, сістэматызацыя і семантычнае апісанне мноства пазначаных прагматым, уключаючы сітуацыі іх ужывання. На думку аўтараў гэта дазволіць

стварыць базу ведаў той часткі беларускай мовы, якая ўяўляе сабой аснову паўсядзённай камунікацыі ў той ці іншай сферы чалавечай дзейнасці.

1. Асноўныя значэнні і актуальнасць даследавання

Прагматымы – моўныя выказванні даўжынёй ад аднаго слова да некалькіх сказаў, кампазіцыянальныя на семантычным і сінтаксічным узроўні, узнаўляльныя як цэлае ў пэўнай аднаўляльнай сітуацыі. Гэта значэнне, як і вылучэнне саміх прагматым у асобны клас належыць тэорыі Сэнс-Тэкст (ТСТ) І. Мельчука [Мельчук, 2006, 2007].

Кампазіцыянальнасць адрознівае прагматымы ад іншых тыпаў фразем, якія з'яўляюцца некампазіцыянальнымі на выключна семантычным (калакацыі: *праводзіць аперацыю, прывесці ў адчай*; ідыёмы: *марскі вожык*) ці сінтаксічным і семантычным (*біць лынды*) узроўні. Узнаўляльнасць у выніку канвенцыяаналізацыі [Morgan, pp. 261 – 280] адрознівае прагматымы ад свабодных словазлучэнняў. Нарэшце, нельга блытаць прагматымы з клішэ, бо прагматымы ўжываюцца ў адной або некалькіх блізкіх, даволі дакладна вызначаных сітуацыях (*Асцярожна, дзверы зачыняюцца*), а сітуацыя ўжывання клішэ занадта генералізаваная (*Летш позна, чым ніколі*).

Прагматычныя фраземы, або прагматымы – устойлівыя і ўзнаўляльныя выразы, і гэта аб'ядноўвае іх з астатнімі фразеалагічнымі

адзінкамі. Адрознівае ж іх ад астатніх механізм фразеалагізацыі: яна адбываецца паміж інфармацыяй (сэнсам, глыбінным зместам, або канцэптuallyным прадстаўленнем у тэрмінах ТСТ) і яе семантычным прадстаўленнем пад уплывам сітуацыі маўлення (камунікатыўнай сітуацыі), з якой адпаведны выраз непарыўна звязаны.

Структурнай асаблівасцю прагматэм яўляецца тое, што яны маюць даўжыню ад аднаго слова да некалькіх сказаў. Гэта значна ўскладняе працэс выяўлення прагматэм у тэкстах. Таму, напрыклад, пры перакладзе тэкстаў так неабходна абавірацца непасрэдна на семантычны аналіз і ўлік сэнсу не толькі асобных слоў ці нават выказаў, а усяго пазатэкставага кантэксту апісанай у тэксце сітуацыі.

Графічна прагматэмы нярэдка маюць асаблівае афармленне (*Смачна есці* на сурвэтцы, *Паркоўка забаронена* з адпаведным сімвалам і г.д.). Таму ў базе ведаў, якую мы ствараем, плануецца змяшчаць таксама адпаведныя ілюстрацыйныя прыклады. Пры апісанні семантыкі сітуацыі такія выпадкі таксама павінны быць пазначаны.

Варта адзначыць, што прагматэмы як асобны тып фразем многімі фразеалагамі не вылучаюцца асобна. Адна з прычын гэтага – непрыняцце імі погляду на мову як на трансфармацыйны механізм, як гэта мае месца ў тэорыі Сэнс – Тэкст. Нават тыя навукоўцы, што ўлічваюць у сваіх працах па фразеалогіі прагматычны аспект, выдзяляюць толькі частку тых выказаў, якія належаць да прагматэм. Між тым, вылучэнне прагматычных фразем у максімальна шырокім сэнсе, іх збор, класіфікацыя і апісанне на розных мовах з’яўляюцца адной з надзённых задач у сферы лінгвадыдактыкі і перакладу (асабліва аўтаматызаванага і аўтаматычнага). У гэтай сувязі прадметам нашага даследавання з’яўляецца прадстаўленасць і спецыфіка прагматычных фразем у беларускай мове.

Прадстаўленае даследаванне з’яўляецца, на нашу думку, досыць актуальным, бо ва ўмовах, калі беларуская мова задзейнічана ў сістэмах аўтаматычнага перакладу, калі вядзецца падрыхтоўка да стварэння беларускага модуля пераўтварэння тэкст у голас, своечасовы ўлік і апісанне такіх адзінак як прагматэмы не саступае па надзённасці апрацоўцы лексічнага матэрыялу, бо пераклад дадзеных выказаў павінен ажыццяўляцца не праз дэ- і рэканструкцыю, а праз эквіваленты. Нельга забывацца і на тое, што многія прагматэмы ўваходзяць у склад самых частаўжывальных выказаў, якія трэба ведаць на памяць для паспяховай камунікацыі на адпаведнай мове. Размоўнікі і падручнікі для іншамовных навучэнцаў – першы тып выданняў, у якіх трэба ўдасканаліць разгляд прагматэм; другі тып – падручнікі па беларускай мове для рускамоўных навучэнцаў, якія, дзякуючы больш уважліваму разгляду прагматэм, павінны

даць чытачу магчымасць пазбавіцца ад празмернага ліку калек, якія засмечваюць яго маўленне.

Уключэнне беларускага слоўніка прагматэм у шматмоўную базу дазволіць пашырыць эмпірычную базу для далейшага ўдасканалення тэорыі прагматэм. Такая база з’явіцца каштоўным рэсурсам для параўнальнага мовазнаўства, лінгвакразнаўства і, магчыма, псіхалінгвістыкі.

Трэба таксама адзначыць, што ў параўнанні з іншымі мовамі (напрыклад, рускай ці французскай), склад беларускіх прагматэм значна бяднейшы. Гэта абумоўлена слабай прадстаўленасцю самой беларускай мовы ў многіх сферах грамадскага жыцця. Але ўлік гэтай акалічнасці пры стварэнні базы ведаў прагматэм, на наш погляд, павінен даць дадатковую глебу для далейшага развіцця беларускай мовы ва ўмовах яе сучаснага сутыкнення з іншымі, больш развітымі мовамі.

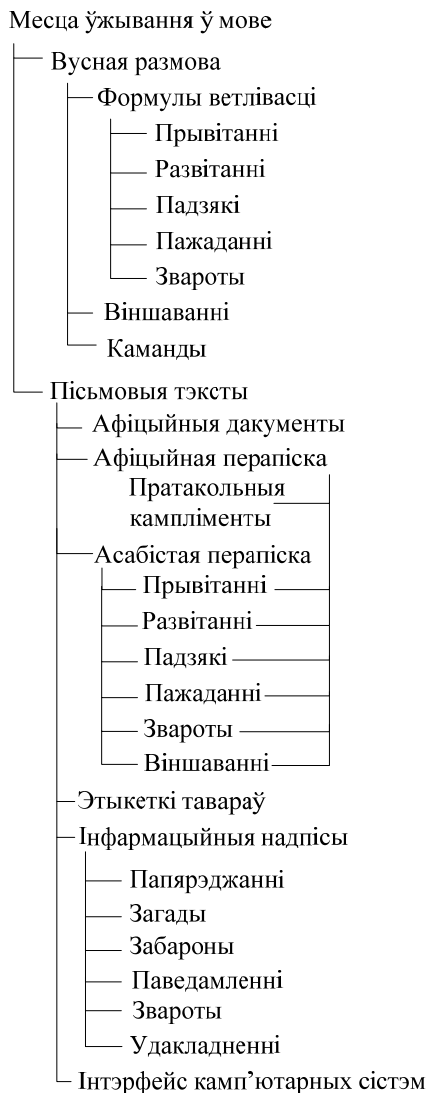
2. Класіфікацыя прагматычных сітуацый і падыходы да іх фармалізацыі

2.1. Класіфікацыя прагматэм і сітуацый іх ужывання

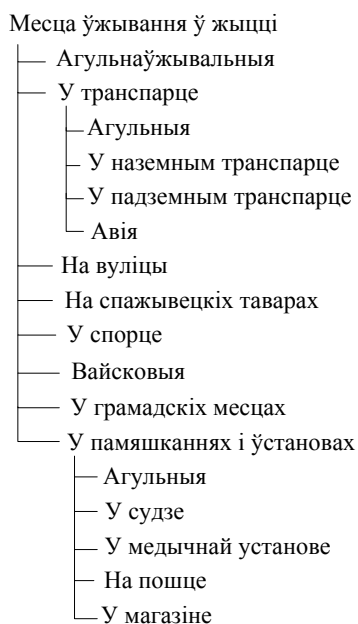
Намі былі вылучаны тэматычныя групы прагматэм паводле таго, да якой вобласці належыць сітуацыя камунікацыі той ці іншай прагматычнай фраземы. Прывядзем тут прыклады некаторых такіх сітуацый з іх нефармальным апісаннем.

На малюнках 1 і 2 прыведзена першапачатковая класіфікацыя сітуацый ужывання прагматэм. Усе сітуацыі ўжывання прагматэм падзелены на 2 вялікія класы: месца ўжывання ў мове і месца ўжывання ў жыцці. Мноствы прагматэм абодвух гэтых класаў цалкам супадаюць, бо дадзены падзел зроблены ў мэтах атрымання больш дасканалай інфармацыі адносна кожнай прагматэмы ў прыкладным аспекце, напрыклад, для выкарыстання пры перакладах ці ў вывучэнні мовы. Відавочна, што дадзеная класіфікацыя не поўная і будзе ўдакладняцца. Напрыклад, жыццёвых сфер ужывання прагматэм значна больш, чым адлюстравана на малюнку.

Варта адзначыць, што назвы некаторых выдзеленых падкласаў сітуацый з розных класаў супадаюць (гл. мал.1). Гэта не выпадкова, бо, напрыклад, адны і тыя ж формулы ветлівасці могуць ужывацца як у вуснай, так і ў пісьмовай формах. Але ў кожным з выдзеленых падкласаў могуць сустракацца і тыя прагматэмы, якія не супадаюць. Інакш кажучы, мноства прагматэм прывітанняў з класу “Формулы ветлівасці” для вуснай размовы, напрыклад, не супадае цалкам з мноствам прывітанняў у пісьмовых тэкстах, але перасячэнне гэтых мностваў не пуста. Таксама ж зразумела, што ёсць пэўныя адрозненні паміж афіцыйнай і прыватнай перапіскай.



Малюнак 1 – Класіфікацыя сітуацый ужывання прагматэм па месцы ўжывання ў мове



Малюнак 2 – Класіфікацыя сітуацый ужывання прагматэм па месцы ўжывання ў жыцці

2.2. Мноствы і першапачатковае апісанне прагматэм

Разгледзем далей прыклады пэўных прагматэм і семантычныя асаблівасці адпаведных ім сітуацый ужывання.

Намі вылучаны і аднесены да класу **прывітанняў** наступныя прагматэмы:

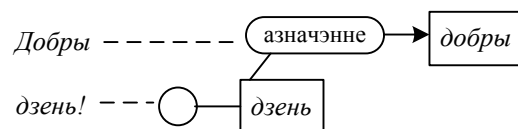
“Дзень добры!”, “Добры дзень!”, “Добрай раніцы!”, “Добры вечар!”, “Прывітанне!”, “Вітаю!”. Усе пералічаныя прагматычныя фраземы ўжываюцца як у вуснай, так і ў пісьмовай формах.

Па месцы ўжывання ў жыцці іх можна аднесці да агульнаўжывальных, бо ў любой сферы чалавечай дзейнасці ёсць месца для прывітанняў. Дадзеныя прагматэмы, адзначым, маюць семантычныя адрозненні. Так, фразама “Добры дзень!” з’яўляецца найбольш распаўсюджанай і універсальнай, ужываецца як у афіцыйных сітуацыях, так і ў неафіцыйных. Крыху менш ужывальная падобная да яе прагматэма “Дзень добры!”. Звычайна яна ўжываецца ў адказ на прывітанне суразмоўцы ці з мэтай павышэння афіцыйнасці камунікацыі.

Менш афіцыйныя формы прывітанняў – “Прывітанне!” і “Вітаю!”. Звычайна яны ўжываюцца паміж сябрамі ці добра знаёмымі людзьмі, але не забараняецца іх ужыванне і пры звароце да незнаёмых. Дадзеныя формы прадугледжваюць таксама сінтаксічныя варыянты зваротаў да адной ці некалькіх асоб: “Прывітанне, ...!”, “Прывітанне, сябры!” і інш., але “..., вітаю!” – звычайна ў звароце да адной даволі знаёмай асобы ці сябра.

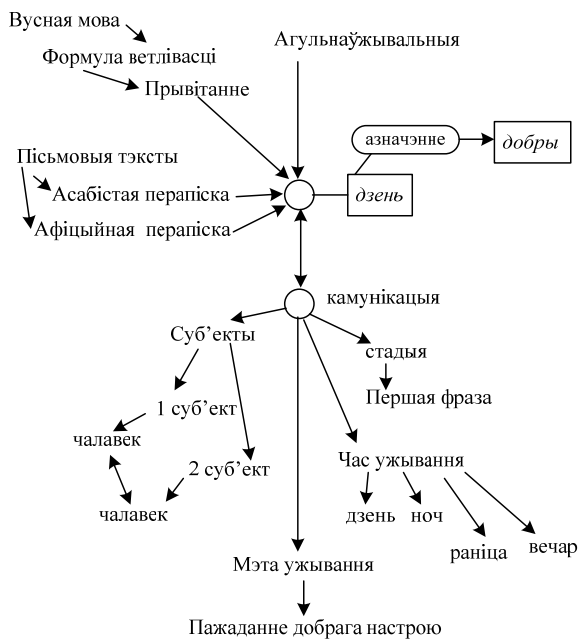
Прагматэмы “Добрай раніцы!” і “Добры вечар!” наўпрост суадносяцца з пэўным часам сутак. Відавочна, што іх немагчыма ўжываць удзень. У адрозненне ад іх прагматэмы “Добры дзень!” і “Дзень добры!” могуць ўжывацца ў любы час сутак. Аб’ядноўвае прыведзеныя чатыры фраземы слова “добры”, якое традыцыйна ўжываецца ва ўсіх выпадках з мэтай ветлівага пажадання суб’екта добра, добрага настрою і г.д.

На малюнку 3 прыведзена дрэва сінтаксічнага падпарадкавання для фразем “Добры дзень!” і “Дзень добры!”, якія на паверхневым сінтаксічным узроўні маюць розны выгляд.



Малюнак 3 – Дрэва сінтаксічнага падпарадкавання прагматэмы

На малюнку 4 прыведзена схематычная паўфармальнае апісанне семантыкі ўжывання прагматэмы “Добры дзень!”. На гэтым жа малюнку адлюстравана суаднясенне дадзенай фраземы да пэўных класаў, вызначаных на мал.1, 2.



На малюнку 4 бачна, што прагматэма “Добры дзень” з’яўляецца першай фразай камунікатыўнага акту паміж двума суб’ектамі – чалавекам і чалавекам, якія могуць “абменьвацца” адной і той жа прыведзенай фразай. Фразема ўжываецца амаль ў любы час сутак з мэтай пажадання добрага настрою. Да гэтага апісання можа быць дададзена спасылка на тое, што 2-гі суб’ект камунікацыі ў адказ на прывітанне 1-га суб’екта можа выкарыстаць іншую фразему з разгледзенага вышэй мноства.

Аналагічным чынам у прадстаўленай працы апісваюцца ўсе вызначаныя прагматэмы. Пасля ўдакладнення класіфікацыі і паўфармальнага семантычнага апісання сітуацый ужывання прагматэм плануецца абагульніць выкарыстання моўных сродкі апісання семантыкі з мэтай павышэння ўзроўню фармальнасці і прыгоднасці да далейшага практычнага прымянення.

Разгледзім прыклады прагматэм з іншых класаў.

У якасці **развітанняў** вызначаны наступныя: “Да сустрэчы!”, “Да хуткай сустрэчы!”, “Да хуткага спаткання!”, “Пакуль!”, “Абдымаю!”, “Цалую!”, “З павагай, ...”, “Дабранач!”, “Добрай ночы!”. У гэтым мностве прагматэм таксама выдзяляюцца тыя, якія ўжываюцца ў любых выпадках (універсальныя) ці ў афіцыйных абставінах (афіцыйныя), якія можна ўжываць толькі неафіцыйна і ў ніякім выпадку – у працэсе камунікацыі афіцыйных асоб (напрыклад, “Пакуль!”), а таксама асабістыя – тыя, якія дарэчы ў стасунках толькі паміж вельмі блізкімі асобамі. Таксама ж як прывітанні, развітання выкарыстоўваюцца паміж двума асобамі, якія ўзаемна абменьваюцца дадзенымі фразамі з мажлівасцю сінанімічных падменаў.

З разгледжаных вышэй прагматычных фразем прывітаннаў і развітанняў можна зрабіць выснову, што пры апісанні семантыкі сітуацый ужывання формул ветлівасці варта звяртаць увагу на

адпаведную мадэль камунікацыі, якая змяшчае характарыстыкі ўдзельнікаў камунікацыі, а таксама час, месца і форму камунікацыі (вусная ці пісьмовая).

У дадатак ці замест развітанняў, напрыклад, напрыканцы афіцыйнага дыпламатычнага ліста ўводзяцца **пратакольныя кампліменты**, якія ў агульнапрынятай форме выражаюць павагу да адрасата. Ступень ветлівасці залежыць ад вышыні пасады апошняга. Так, у лісце да міністра, намесніка міністра, дырэктара дэпартаменту МЗС (мае ранг надзвычайнага і паўнамоцнага пасла, альбо роўны яму ранг), сенатара, губернатара, пасла, пасланніка, былых міністра, пасла ці сенатара прынята пісаць “*Запэўніванне ў вельмі высокай павазе*”. Намесніку міністра, дырэктару дэпартаменту ці начальніку ўпраўлення ў МЗС, дэпутату парламента, паверанаму ў справах, саветніку-пасланніку або намесніку міністра, генеральнаму консулу адпавядае “*Запэўніванне ў высокай павазе*”. Да начальніка аддзела МЗС, мэра горада, саветніка ці консула пішуць “*Запэўніванне ў глыбокай павазе*”. Для параўнання, рускія эквіваленты трох узроўняў ветлівасці будуць адпаведна “*Заверение в весьма высоком уважении*”, “*Заверение в высоком уважении*” і “*Заверение в глубоком уважении*”.

У якасці **зваротаў** намі вылучаны: “*Паважаны(ая)...*!”, “*Паважаныя ...!*”, “*Паважаныя спадары!*”, “*Спадары!*”, “*Спадары і спадарыні!*”, “*Сябры!*”, “*Паважаныя калегі!*”, “*Паважаныя студэнты!*”, “*Алё!*”.

Запрашэнні: “*Шчыра запрашаем!*”, “*Сардэчна запрашаем!*”, “*Калі ласка!*”, “*Праходзьце, калі ласка!*”.

Падзякі і адказы на падзяку: “*Вельмі прыемна!*”, “*Дзякуй!*”, “*Вялікі дзякуй!*”, “*Дзякуй вялікі!*”, “*Няма за што!*”, “*Калі ласка!*”. Варта адзначыць, што пры апісанні семантыкі сітуацый ужывання пералічаных фразем неабходна ўлічваць іх узаемадапаўняльнасць і парнасць, якая можа мець строгі парадак. Напрыклад, у адказ на “*Дзякуй!*” звычайна гучыць “*Калі ласка!*” ці “*Няма за што!*”. У адваротнай паслядоўнасці калі-нікалі можа сустракацца спачатку “*Калі ласка!*” і ў адказ “*Дзякуй!*”, але ніколі – “*Няма за што!*” і “*Дзякуй!*”.

Пажаданні: “*Прыемнага апетыту!*”, “*Смачна есці!*”, “*Будзь здаровы!*”, “*Будзьце здаровы!*”. Сюды ж мы адносім пакуль і **тосты**: “*За ваша здароўе!*”, “*Будзьма!*”.

Віншаванні: “*З Новым годам!*”, “*З Раством!*”, “*З Раством Хрыстовым!*”, “*Хрыстос уваскрос!*”, “*Сапраўды ўваскрос!*”, “*З Днём нараджэння!*”, “*З Днём народзінаў!*”, “*Шчаслівых Калядаў!*”, “*Вясёлых Калядаў!*”.

Загады, забароны: “*Не курьце!*”, “*Курьце забаронена!*”, “*Па газонах не ходзіце!*”, “*Паркоўка забаронена!*”, “*Паркоўка асабістага аўтатранспарту забаронена!*”. Адзначым, што ў

адрозненне ад разгледжаных раней фразем, пералічаныя тут загады і забароны прадугледжаны ў іншай мадэлі камунікацыі, дзе адрасатам з'яўляецца жывы чалавек, але ў якасці адрасанта заўсёды выступае пісьмовы надпіс. Ніколі ў дадзеным выпадку не дазваляецца ўжываць прыведзеныя фраземы ў вуснай форме, бо гэта можа ўспрымацца як знявага да адрасату.

Тым не менш, у мове існуе шэраг **каманд**, якія маюць сваё выражэнне ў вуснай форме і не ўспрымаюцца непаважліва. Да больш пашыраных каманд належыць такая прагматэма, як *“Рукі дагары!”* (руская *“Руки вверх!”*). Яна выкарыстоўваецца ў прынцыпе любой узброенай асобай, якая праз наяўнасць зброі надзелена правам загадваць. А вось яшчэ адна вайсковая каманда: *“Зброю на плячо!”* (у рускай: *“На плечо!”*). Аднак апошні беларускі прыклад не з'яўляецца, наколькі нам вядома, зацверджаным паводле Статута і, тым болей, не ўжываецца ў войску ў рэчаіснасці. Варта адзначыць, што вайсковыя каманды адрозніваюцца асаблівай схільнасцю да кароткасці і лаканічнасці, і тлумачыцца гэта вельмі проста: ва ўмовах баявых дзеянняў ад апэратыўнасці залежыць жыццё. Любая каманда павінна быць кароткай і максімальна проста для разумення. Для забеспячэння хуткасці рэакцыі на каманды выкарыстоўваюцца ўсе даступныя сродкі – у тым ліку і прасадычныя. Напрыклад, у прагматычна абумоўленай лексеме *“Напра-ВО!”* выдзяляецца, нараспеў вымаўляецца апошні склад, што надае рухам салдат патрэбны імпульс, адзначае пэўным чынам рытм выканання каманды. Паколькі акадэмічны пераклад вядзецца па рускай версіі Статута, то, на нашу думку, непазбежна з'явіцца калькі рускіх каманд, бо яны першымі ўзнікаюць у розуме білінгва. Яскравым прыкладам такіх перакладаў нам здаецца флорская каманда *“Адаць швартовы!”*, якая хутчэй нагадвае вымаўлены з беларускім акцэнтам рускі выраз *“Отдать швартовы!”*. Магчыма, для пераходнага перыяду празмернае следаванне логіцы рускай мовы – з'ява не толькі непазбежная, але і ў нечым карысная. Людзям будзе нашмат прасцей засвоіць знаёмыя, пазнавальныя каманды ў арміі ці надпісы на дарожных знаках, чым нешта зусім новае, нібыта запазычанае з замежнай мовы.

Адзначым таксама, што ў судовай практыцы ў нас ужываецца руская мова, таму пакуль рана гаварыць пра беларускія эквіваленты выразаў кшталту *“Встать! Суд идет!”*, *“Я протестую!”*. Тое ж самае адбываецца і ў спорце, танцах, калі трэнер часта дае сігналы: *“На счет [NUM] + [Умн.л.,пр.ч.,законч.трив.]”*, *“На старт!”*, *“Внимание!”*, *“Марш!”* [Расейска-беларускі, 2006].

Разгледзем зараз некалькі прыкладаў **інфармацыйных надпісаў**. Адразу папярэдзім, што гэты пункт утрымлівае некалькі тэматычных раздзелаў. Гэта сфера, дзе яскрава адлюстраваны стан неўраўнаважанай двухмоўнасці ў Беларусі. Канешне ж, калі нават збор правіл дарожнага руху дагэтуль не быў цалкам перакладзены на

беларускую мову, то няздзіўна, што і на многіх дарожных знаках на сённяшні дзень пераважаюць рускія прагматэмы. Праўда, нам удалося знайсці такія прыклад, як: *“Акрамя аўтатранспарту NGen”* (МУС, напрыклад). Гэта, дарэчы, адзін з інтэрсемятычных знакаў, бо выкарыстоўвае і іканічную выяву, і тэкст.

На інфармацыйных надпісах ў грамадскіх месцах беларуская мова больш ужывальна. Напрыклад амаль кожнаму вядомы наступныя **папярэджэнні**: *“Працуем без выхадных”*, *“Працуем штодзённа”*, *“Аварыйны выхад”*, *“Службовы ўваход”*, *“Службовае памяшканне”*, загады: *“Захоўвайце цішыню”*, запрашэнні: *“Сардэчна (шчыра) запрашаем!”*, прапановы: *“Лічыце грошы, не адыходзячы ад касы”*.

Інфармацыйныя надпісы сустракаюцца таксама і ў **паліграфічнай прадукцыі**: *“Выдзелена нам”*, *“Курсіў наш”*.

Інфармацыйныя надпісы **на спажывецкіх таварах** утрымліваюць тэхнічную, прававую інфармацыю, інструкцыі: *“Усе правы абаронены”*, *“Захоўваць пры тэмпературы (...) адноснай вільготнасці наветра не большай за (...)”*, *“Дата вырабу (дата)”*, *“Зроблена ў (краіна)”*, *“Прыдатны да (дата)”*. Відавочна, што дадзенае мноства прагматычных фразем мае пэўныя сінтаксічныя асаблівасці і варыяцыі, якія неабходна дакладна апісаць у адпаведных фармальных мадэлях.

У **транспарце** ўжываюцца пісьмовыя і вусныя прагматычныя фраземы. Пісьмовыя **паведамленні**: *“Няма ўваходу”*, **забароны**: *“Не прыхіляцца”*, *“Не прыхінацца”*, **загады**: *“Не адцягвайце ўвагу вадзіцеля падчас руху”*. Вусныя **папярэджэнні**: *“Дзверы зачыняюцца!”*, **паведамленні**: *“Наступны прыпынак ...”*, *“Канцавая”*, *“Канечны прыпынак”*, **звароты**: *“Не забывайце аплачваць праезд”*, *“Аплачаны праезд - гарантыя якасці транспартных паслуг”*; **папярэджэнні і загады**: *“Поезд далей не ідзе - Калі ласка, пакіньце вагоны”*.

Асобным чынам варта адзначыць існаванне прагматычных фразем у інтэрфэйсах камп'ютарных ці іншых тэхнічных сістэм. Гэта фраземы мадэлі камунікацыі “чалавек – машына”. Тут маюцца на ўвазе фіксаваныя выразы, якія звязаны з вельмі канкрэтна акрэсленымі відамі працоўных сітуацый, дакладней кажучы, з апэрацыямі, якія можна выконваць у рамках пэўнай сістэмы. У якасці прыкладаў такіх прагматэм можна прывесці наступныя: *“Захаваць як”*, *“Новы файл”*, *“Дастаць картку”*.

Ступень выкарыстання беларускай мовы тут нізкая (за выключэннем беларускамоўных сайтаў (ці беларускамоўных версій сайтаў), банкаматаў, рэдкіх праграмных прадуктаў), але калі нейкі выраз увайшоў ва ўжытак моўнага калектыву (хай і нешматлікага, як гэта характэрна для беларускай мовы ў Беларусі), можна казаць пра прагматэму.

2.3. Кампаненты фармальнай мадэлі апісання прагматымы

Абагульняючы прыведзеныя вышэй апісанні прагматым, выдзелім тыя ўласцівасці, якія неабходна ўлічваць дзеля іх фармалізацыі:

- Форма ўжывання: вусная ці пісьмовая
- Месца ўжывання
- Час ужывання (абсалютны ці адносны, з нагоды некаторай падзеі)
- Мэта ўжывання
- Характарыстыкі суб'ектаў камунікацыі: колькасць, адушаўлёнасць (чалавек ці надпіс), пол, узрост (абсалютны ці адносны), камунікатыўная роля (адрасант, адрасат)
- Сінтаксічныя асаблівасці
- Існаванне і характар сувязі з іншымі прагматымамі (сінанімічнасць, супрацьлегласць, узаемная паслядоўнасць і інш.)
- Пазіцыя ў кантэксце (калі існуе, напрыклад, пачатак ліста ці афіцыйнага дакумента, завяршэнне размовы і г.д.)

3. Прыкладныя аспекты выкарыстання фармалізаванай семантыкі прагматым беларускай мовы

У якасці прымяненняў класіфікацыі і фармалізаванага апісання прагматым варта адзначыць наступныя:

- пераклад;
- камп'ютарная лінгвадыдактыка;
- семантычны пошук;
- стварэнне інтэлектуальных сістэм з моўным інтэрфейсам;
- аўтаматызацыя стварэння паралельных корпусаў тэкстаў;
- дапаўненне корпусаў тэкстаў прагматычнай інфармацыяй.

БІБЛІАГРАФІЧНЫ СПІС

[Мельчук, 2006] Мельчук И. А. Опыт теории лингвистических моделей «Смысл – Текст». Семантика, синтаксис / И. А. Мельчук. – Москва: Школа «Языки русской культуры», 1999. – 346 с.

[Мельчук, 2007] Мельчук И.А. Смысл и сочетаемость в словаре / Л.Н.Иорданская, И.А Мельчук. – Москва: Языки славянской культуры, 2007. – 672 с.

[Расейска-беларускі, 2006] Расейска-беларускі вайсковы слоўнік / пад рэд. П. Сцяцко. – Мінск: ТБМ, 1997. – 250 с.

[Morgan, 2006] Morgan J. L. Two Types of Convention in Indirect Speech Acts / J. L. Morgan // Syntax and Semantics vol. 9. – London: Academic Press, 1978. – pp. 261 – 280.

FORMALIZATION OF THE SEMANTICS OF PRAGMATIC PHRASEMES OF THE BELARUSIAN LANGUAGE

Kim Y.A., Yeliseyeva O. E.

*Belarusian State University,
Minsk, Republic of Belarus*

brikim@gmail.com

olac@open.by

Like any other type of idiomatic expressions, pragmatic phrasemes, or pragmatemes need to be translated as a whole, but in order to achieve that, they first need to be recognized as a whole. The problem is that many pragmatemes lack idiomaticity both on the morphosyntactic and semantic levels. Their semantics are nearly always represented by only one morphosyntactic representation. Although a new semantic and morphosyntactic representation can be constructed according to the rules of the language, it will not have the same effect in the given situation of usage. Therefore, it is the situation of usage that prescribes a particular morphosyntactic representation for the given semantic representation. In other terms, if we formalize the pragmatical component of these phrasemes, we can enable the machine to find adequate equivalents or provide more precise search results. The present article deals with pragmatemes in the Belarusian language and suggests a technique for describing the semantics of the situations in which pragmatemes are used. Such descriptions will serve as a start point for creating a knowledge base of the pragmatemes of the Belarusian language. A knowledge base, in its turn, can be applied in machine translation, semantic search and language learning systems, as well as in corpus linguistics.



УДК 004.822:514

**НЕКОТОРЫЕ АСПЕКТЫ СЕМАНТИЧЕСКОГО КОДИРОВАНИЯ: СПОСОБЫ
ВЫРАЖЕНИЯ ПРИЧИННОСТИ В ЯЗЫКЕ-СИСТЕМЕ**

Головня А.И.

*Белорусский государственный университет**г. Минск, Республика Беларусь***golovnjaai@bsu.by e-mail.address**

В статье рассматриваются некоторые аспекты семантического кодирования, связанного с причинностью. Доказывается, что причинность в языке-системе может быть связана с предлогами, союзами, наречиями и фразеологизмами. Определенная часть причинности в языке ничем не выражена, но контекстуально она присутствует. Частично причинность проиллюстрирована примерами из романа М. Булгакова «Мастер и Маргарита». Сделана попытка рассмотреть причинность на уровне ядерных композиций.

Ключевые слова: наречия, причинность, предлоги-детерминаторы, союзы, фразеологические обороты.

Среди способов выражения причины и причинности в языке как отражательной системе человеком были развиты несколько способов передачи причинно-следственных отношений. Грамматика их так и дает. Среди них – предлоги, союзы, наречия и фразеологизмы. Определенная же часть причинных отношений формально ничем не выражается.

Предлоги. Формальными показателями причинности в текстах служат предлоги, имеющие среди других и значение "причинности" или лексему "причина". Такое явление происходит потому, что предлоги – наиболее древние образования и в их семантике наслаивались самые различные значения. Практически однозначных предлогов нет, они имеют в норме по 2-3 значения.

Так, предлог *из-за* выражает два значения – причины и места: ср. предложения

Он выглядывал из-за забора и Он не смог приехать из-за болезни.

Предлог *от* аналогично передает два значения – места и причины, ср. *Он отошел от окна и Она заплакала от боли.*

Здесь имеются своеобразные лакмус-признаки, позволяющие формально отделить одно значение от другого – в виде наличия у глагола префикса, происшедшего от предлога своего рода симметрия – *ото-йти от..., ото-двинуться от... отъ-ехать от...,* однозначно представляющие значение места. Такие формализмы следует отыскивать перед автоматической обработкой текстов с целью отыскания глаголов, связанных лишь с причинно-следственными отношениями. Все это требует более тщательного анализа предлогов.

К причинным относят предлоги – *из-за, от, по, за, в, под, перед, без.*

ИЗ + род. пад. *из зависти, из жалости;*

С + род. пад. *с горя, с тоски;*

ПО + дат.пад. *по неведению, по глупости, по привычке;*

ЗА + вин. пад. *отругать за шалость, наказать за ошибку.*

О мощности явления можно косвенно судить по частоте употребления предлогов по данным частотного словаря: *в* = 42854, *на* = 17262, *с* = 12975, *к* = 6246, *по* = 5176, *за* = 5048, *у* = 4623, *из* = 4598, *о* = 4156, *от* = 3752, *для* = 3254, *до* = 2063. На их фоне явные причинные предлоги типа *из-за* = 238, *ввиду* = 22 выглядят скромно и это свидетельствует о том, что во многих случаях причинность формально не выражена и следует ее обнаруживать лишь по семантическим критериям, т.е. не в автоматическом режиме.

Само слово *причина* в этом словаре имеет частоту 242 (восьмикратно превышая среднюю – 30). Такое положение с причинными предлогами заставляет нас провести эксперимент на большом объеме текстов, чтобы выяснить хотя бы соотношение употребительности двух значений предлога. Пока же покажем направление дальнейшей работы на трех более важных причинных предлогах: *из-за, от и за.*

Сочетания с предлогом *из-за* (глагольные конструкции). В причинных конструкциях с этим предлогом отмечаются глаголы возвратные и невозвратные, совершенного и несовершенного вида. Первичный анализ позволил выделить несколько семантических подгрупп глаголов, связанных через предлог *из-за* причинно-следственными отношениями с существительными.

Таблица 1

Глагольные сочетания, представленные глаголами чувств, эмоций, состояний с предлогом *из-за*

глаголы чувств, эмоций, состояний	предлог	Причины (род.пад. ед. или мн. ч.)
Спорить, страдать, горевать, грустить, тужить, киснуть, скорбеть, вздыхать, унывать, хандрить, переживать, плакать Кукситься, гневаться, огорчаться, печалиться, сокрушаться, поругаться, сердиться, убиваться, расстроиться, разбушеваться, огорчиться, волноваться, беспокоиться, горюниться, кручиниться и др.	из-за	ерунды, чепухи, пустяков, мелочей процессы: опоздания, задержки негативы: неудачи, невнимания, непослушания, небрежности, несогласия, невежества и т.п. или бестактности, безразличия, бессердечия и т.п.

К ним примыкают фразеологические сочетания с тем же предлогом – *падать духом из-за...*, *вешать нос из-за...*, *приходить в уныние из-за ...* и ряд других.

Таблица 2

Глагольные сочетания, представленные глаголами результатами, обобщающими глаголами и глаголами минусовых детализаций с предлогом *из-за*

глаголы-результативы	предлог	Причины и следствия
проиграть, обанкротиться, погибнуть, пустовать, вибрировать	из-за	Невезения
обобщающие глаголы	предлог	Причины и следствия
Совершиться, сотвориться, получиться, проистекать, приключиться, происходить, страстись, произойти, нарушиться; словосочетания: выйти из строя	из-за	Квантор ситуации <i>это</i> чаще всего негативы неосторожности, неполадок, небрежности, неосторожности, неаккуратности и т.п.

минусовые (детализации)	предлог	причины и следствия
Переломиться, перегореть, прерваться, испортиться, сломаться, опоздать, задержаться, развалиться	из-за	Чаще суперлативные существительные с приставкой пере-: перенапряжения, перегрузок или с – недо-: недогрузок

Для анализа данных по употребительности предлога *из-за* был проведен эксперимент. Мы взяли для исследования достаточно большой текст – весь роман М.Булгакова «Мастер и Маргарита». Его объем свыше 112 тысяч словоформ, сжался с учетом повторяемости в тексте той или иной словоформы до 26 тысяч. На этом объеме предлог *из-за* отмечен в 43-х предложениях. В трех предложениях он отмечен по 2 раза (1, 16 и 43). Из этого числа 26 раз с причинным значением и 20 раз со значением места.

1. *Крысобою вообще все провожали взглядами, где бы он ни появлялся, из-за его роста, а те, кто видел его впервые, из-за того еще, что лицо кентуриона было изуродовано: нос его некогда был разбит ударом германской палицы.*

2. *И все из-за того, что он неверно записывает за мной.*

3. *Я вижу, что совершается какая-то беда из-за того, что я говорил с этим юношей из Кириафа.*

4. *Вспомни, как мне пришлось из-за вас снимать со стен щиты с вензелями императора, перемещать войска, пришлось, видишь, самому приехать, глядеть, что у вас тут творится!*

5. *План Берлиоза следует признать правильным: нужно было добежать до ближайшего телефона-автомата и сообщить в бюро иностранцев о том, что вот, мол, приезжий из-за границы консультант сидит на патриарших прудах в состоянии явно ненормальном.*

6. *Один лунный луч, просочившись сквозь пыльное, годами не вытираемое окно, скупо освещал тот угол, где в пыли и паутине висела забытая икона, из-за киота которой высовывались концы двух венчальных свечей.*

7. *Говорили, говорили мистики, что было время, когда красавец не носил фрака, а был опоясан широким кожаным поясом, из-за которого торчали рукояти пистолетов, а его волосы воронова крыла были повязаны алым шелком, и плыл в Караибском море под его командой бриг под черным гробовым флагом с адамовой головой.*

8. *Вчера днем он приехал из-за границы в Москву, немедленно явился к Степе и предложил свои гастроли в варьете.*

9. *И сразу поэт запутался, главным образом из-за слова "покойным".*

10. *Теперь Иван лежал в сладкой истоме и поглядывал то на лампочку под абажуром, льющую с потолка смягченный свет, то на луну, выходящую из-за черного бора, и беседовал сам с собою.*

11. *Почему, собственно, я так взволновался из-за того, что Берлиоз попал под трамвай?*

12. *И, повертев перед глазами Римского узловатыми пальцами, внезапно вытащил из-за уха*

у коты собственные Римского золотые часы с цепочкой, которые до этого были у финдиректора в жилетном кармане под застегнутым пиджаком и с продетой в петлю цепочкой.

13. Через минуту из-за занавески вышла брюнетка в таком платье, что по всему партеру прокатился вздох.

14. Успокоившись, он сел, сказал: – а впрочем, бог с ним, – и продолжил беседу с Иваном: – так из-за чего же вы попали сюда?

15. Из-за Понтия Пилата, – хмуро глянув в пол, ответил Иван.

16. Видите ли, какая странная история, я здесь сижу из-за того же, что и вы, именно из-за Понтия Пилата, – тут гость пугливо оглянулся и сказал: – Дело в том, что год тому назад я написал о Пилате роман.

17. Иван представлял себе ясно уже и две комнаты в подвале особнячка, в которых были всегда сумерки из-за сирени и забора.

18. Ну, одолели наконец, погрузили в машину, – гудел Варенуха, выглядывая из-за листа и ладонью прикрывая сияк.

19. Валютчик он! – Выкрикивали в зале, – из-за таких-то и мы невинно терпим!

20. Из-за закрытой двери кабинета доносился грозный голос, несомненно принадлежащий Прохору Петровичу – председателю комиссии. – До-ми-со-ль-до!

21. Вытащил наиболее застенчивых из-за шкафов, где они пытались спастись от пения, Косарчуку сказал, что у него абсолютный слух, занял, заскулил, просил уважить старого регента-певуна, стучал камертоном по пальцам, умоляя грянуть "Славное море".

22. «Гм... Спокойно... – Подумал профессор, – он влетел, когда я отходил от окна. Все в порядке», – приказал себе профессор, чувствуя, что все в полном беспорядке и, конечно, главным образом из-за этого воробья!

23. Вот из-за этого этот Желдыбин так и волнуется теперь.

24. Лезу я в какую-то странную историю, но, клянусь, только из-за того, что вы поманили меня словами о нем!

25. Но иду на все из-за него, потому что ни на что в мире большие надежды у меня нет!

26. Я погибаю из-за любви!

27. Фырканые стало приближаться, и из-за ракитовых кустов вылез какой-то голый толстяк в черном шелковом цилиндре, заломленном на затылок.

28. Нагие ведьмы, выскочив из-за верб, выстроились в ряд и стали приседать и кланяться придворными поклонами.

29. Тотчас из-за одного из памятников показался черный плащ.

30. Маркиза, – бормотал Коровьев, – отравила отца, двух братьев и двух сестер из-за наследства!

31. Она поднялась из-за стола.

32. Он мне ненавистен, этот роман, – ответил Мастер, – я слишком много испытал из-за него.

33. Услышав, что шаги стихают, Аннушка, как змея, выскользнула из-за двери, бидон поставила к стенке, пала животом на площадку и стала шарить.

34. Неужели вы, при вашем уме, допускаете мысль, что из-за человека, совершившего преступление против кесаря, погубит свою карьеру прокуратор Иудеи?

35. Ремиз, – заорал кот, – ура! – И тут он, отставив в сторону примус, выхватил из-за спины браунинг.

36. Швейцар выпучил глаза, и было отчего: никакого коты у ног гражданина уже не оказалось, а из-за плеча его вместо этого уже высовывался и порывался в магазин толстяк в рваной кепке, действительно, немного смахивающий рожей на коты.

37. Продавицы с визгом кинулись бежать из-за прилавка, и лишь только они выскочили из-за него, вспыхнули полотняные шторы на окнах и на полу загорелся бензин.

38. Публика, сразу подняв отчаянный крик, шарахнулась из кондитерского назад, смяв более ненужного Павла Иосифовича, а из-за рыбного гуськом со своими отточенными ножами рысью побежали к дверям черного хода продавцы.

39. Не хочешь ли ты ободрать весь земной шар, снеся с него прочь все деревья и все живое из-за твоей фантазии наслаждаться голым светом?

40. Он просит, чтобы ты, которая любила и страдала из-за него, вы взяли бы тоже, – в первый раз моляще обратился Левий к Воланду.

41. Я из-за тебя всю ночь вчера тряслась нагая, я потеряла свою природу и заменила ее новой, несколько месяцев я сидела в темной камерке и думала только про одно – про грозу над Ершалаимом, я выплакала все глаза, а теперь, когда обрушилось счастье, ты меня гонишь?

42. Когда же навстречу им из-за края леса начала выходить багровая и полная луна, все обманы исчезли, свалилась в болото, утонула в туманах колдовская нестойкая одежда.

43. И как раньше Римский страдал из-за Степы, так теперь Варенуха мучился из-за Алоизия.

Сочетания с предлогом от. Глагольные сочетания с предлогом *от* представляют несколько иные семантико-тематические группы:

Таблица 3

Глагольные сочетания с предлогом *от*

глаголы звучания	предлог	причина в род.пад.
взвыть, заорать, рывкнуть, закричать, замычать, зарычать, заплакать, фыркать	от	злости, обиды, боли, страха, ужаса (минусовые эмоции); радости, удовольствия, страсти, (плюсовые эмоции)
глаголы состояния чувств	предлог	причина в род.пад.
ошалеть, оторопеть, обезуметь, обомлеть, окаменеть, отупеть, одуреть, устать	от	ужаса, страха, боли, неожиданности, радости

Глаголы вышеуказанных групп предполагают субъект, но не имеют объекта и это их характерная особенность. Проиллюстрируем это примерами из романа М. Булгакова «Мастер и Маргарита».

1. *Фу ты черт!* – воскликнул редактор, – ты знаешь, Иван, у меня сейчас едва удар от жары не сделался.

2. *Позвольте вас поблагодарить от всей души.*

3. *От флигелей в тылу дворца, где расположилась пришедшая с прокуратором в Еришалаим первая когорта Двенадцатого Молниеносного легиона, заносило дымком в колоннаду через верхнюю площадку сада, и к горьковатому дыму, свидетельствовавшему о том, что кашевары в кентуриях начали готовить обед, примешивался все тот же жирный розовый дух.*

4. *Это она, опять она, непобедимая, ужасная болезнь гемикрания, при которой болит полголовы. От нее нет средств, нет никакого спасения.*

5. *Не удержавшись от болезненной гримасы, прокуратор искоса, бегло проглядел написанное, вернул пергамент секретарю и с трудом проговорил: – Подследственный из Галилеи?*

6. *Прокуратор был как каменный, потому что боялся качнуть пылающей адской болью головой.*

7. *Услышит нас всемогущий кесарь, укроет нас от губителя Пилата.*

8. *То пространство, которое он только что прошел, то есть пространство от дворцовой стены до помоста, было пусто, но зато впереди себя Пилат площади уже не увидел – ее съела толпа.*

9. *Пилат задрал голову и уткнул ее прямо в солнце. Под веками у него вспыхнул зеленый свет, от него загорелся мозг, и над толпой полетели хриплые арамейские слова:....*

10. – Ну уж это положительно интересно, – трясясь от хохота, проговорил профессор, – что же это у вас, чего ни хватились, ничего нет!

11. *Поэт бросился к турникету, как только услышал первый вопль, и видел, как голова подсказывала по мостовой. От этого он до того обезумел, что упавши на скамью, укусил себя за руку до крови.*

12. *А подозрительный профессор сделал надменное лицо, повернулся и пошел от Ивана прочь.*

13. *Сидящие на стульях, и на столах, и даже на двух подоконниках в комнате правления МАССОЛИТа серьезно страдали от духоты.*

14. *Бескудников стукнул пальцем по циферблату, показал его соседу, поэту Двубратскому, сидящему на столе и от тоски болтающему ногами, обутыми в желтые туфли на резиновом ходу.*

15. *Лишь тогда, когда перестали доноситься всякие звуки извне, гость отодвинулся от Ивануа заговорил погромче.*

16. *В голове сложилась праздничная картина позорного снятия Степы с работы. Освобождение! Долгожданное освобождение финдиректора от этого бедствия в лице Лиходеева.*

17. *Он поднялся с кресла (то же сделал и финдиректор) и отступил от стола на шаг, сжимая в руках портфель.*

18. *Итак, прошло со времени подъема процессии на горуболее трех часов, и солнце уже снижалось над Лысой Горой, но жар еще был невыносим, и солдаты в обоих оцеплениях страдали от него, томилась от скуки и в душе проклинали трех разбойников, искренне желая им скорейшей смерти.*

19. *Под этими шалашами и скрывались от безжалостного солнца сирийцы.*

20. *И тут от предчувствия радостного конца похолодело сердце бывшего сборщика.*

21. *Повешенный на нем Гестас к концу третьего часа сошел с ума от мух и солнца и теперь тихо пел что-то про виноград, но головою, покрытой чалмой, изредка все-таки покачивал, и тогда мухи вяло поднимались с его лица и возвращались на него опять.*

Сочетания с предлогом за:

Глаголы чувств: *любить, ненавидеть за...*

Глаголы-результативы: *арестовать, наказать, выдрать, побить, выслать, уволить за...*

Глаголы речевые: *проклинать, порицать, ругать, упрекать, журить, хаять, осуждать, хулить, чернить за...* и другие (с минусовой семантикой), а также и глаголы с плюсовой семантикой – *благодарить, хвалить, поощрить, наградить за...* (антонимичные группы, что заставляет тщательно анализировать и данные антонимического словаря) и сочетания типа – *предать анафеме за...* Глаголы этой группы обязательно имеют объект и субъект и это отличает их от глаголов, сочетающихся с предлогом *от*.

Кроме них следует отслеживать всю сочетаемость слова *причина* во всех падежах, так как выражение причинности отмечается отдельно и в комбинаторике двух падежей.

По причине праздников занятия отменили (дательный + родительный).

Это послужило причиной пожара (творительный).

Не устаешь удивляться причинам отсутствия студентов на лекции (дательный падеж).

О причинах можно узнать из газет (предложный).

Группа причинных предлогов небольшая для создания относительно полного списка детерминантов-глаголов и детерминантов-существительных. Взяв предлоги как детерминаторы причинных ситуаций, следует в автоматическом режиме накапливать множество предложений, содержащих омонимичный предлог. Затем вручную выделить нужные причинные значения предлога. После этого можно заниматься их классификацией с учетом семантики глагола и семантики существительного. Опыт накопления данных с предлогом *из-за* и *от* показывает реальные возможности создания большого банка данных.

Союзы. Академическая грамматика приводит список основных простых и составных (двух- и болееэлементных причинных союзов): одноэлементные – *поэтому, ибо, раз, поелику* (уст.); двусоставные – *так как, потому как, потому что, постольку поскольку, а то;* трехэлементные – *благодаря тому что, виду того что, за то что;*

четырёхэлементные – по причине того что, в результате того что. Например: *На лестнице трясущийся, дрожащий старик упал, потому что ему показалось, что на него сверху мягко обрушился Варенуха.*

С одним из союзов причины – *поэтому* мы провели тот же самый анализ, что и с предлогом *из-за*. На объеме всего романа отмечено лишь 11 случаев употребления этого причинного союза. Показательно, что в пяти случаях предложение начинается с союза. Это значит, что такое предложение можно считать полностью “следственным”, так как причина находится в предыдущем предложении. Это усложняет вдвое автоматическое извлечение материала, т.к. сначала нужно извлечь все предложения с союзом *поэтому* и только после этой операции извлекать предложения, содержащие собственно причину.

Она может находиться как перед следственным предложением, так и может быть отнесена вперед к началу текста на большее расстояние. Положительным моментом является то, что эти союзы однозначны и в отличие от предлогов они имеют только причинное значение.

1. *Поэтому нет ничего удивительного в таком хотя бы разговоре, который однажды слышал автор этих правдивейших строк у чугунной решетки Грибоедова: – Ты где сегодня ужинаешь, Амвросий?*

2. *Поэтому первый путь Иван отринул.*

3. *Поэтому Иван и от этого пути отказался, решив избрать третий: замкнуться в гордом молчании.*

4. *Вся свита оказывала ему знаки внимания и уважения, и вход его получился поэтому очень торжественным.*

5. *Он объявил, что у супруги его грипп и что он поэтому просит передать ей что-нибудь через него.*

6. *Стараясь не попадать своими глазами в мои, Лапшенникова сообщила мне, что редакция обеспечена материалами на два года вперед и что поэтому вопрос о напечатании моего романа, как она выразилась, отпадает.*

7. *Мухи и слепни поэтому совершенно облепили его, так что лицо его исчезло под черной шевелящейся массой.*

8. *Поэтому она стала делать что попало.*

9. *Вот поэтому я прошу вас заняться этим делом, то есть принять все меры к охране Иуды из Кириафа.*

10. *Поэтому он переменил позу, поднялся, зашел сбоку и передние лапы и голову положил на колени прокуратору, вымазав полы плаща мокрым песком.*

11. *Поэтому и лежит перед нею на скатерти под лампой заранее приготовленный шприц в спирту и ампула с жидкостью густого чайного цвета.*

Академическая грамматика отмечает, что иногда причинность может выражаться и с помощью наречий. Список их предельно мал и лишь показывает возможности их использования в качестве детерминаторов причинно-следственных отношений: *впопыхах, поневоле, сгоряча, назло,*

поэтому, посему, потому, попусту, понапрасну, насмарку.

Наречия этого типа являются указаниями, что чаще всего в таких предложениях причинно-следственные отношения не выражены предложной – явной связью.

Следует проанализировать весь корпус наречий и выделить из них те, которые служат детерминаторами причинно-следственных отношений. Они будут маркерами нестандартных, нештатных, непредвиденных ситуаций. Так, наречие *вдруг* или *ни с того ни с сего* показывают, что произошедшее действие не выражает явной причины, что эта причина скрыта или удалена по времени так, что она не ясна.

Дело будет осложняться тем, что большинство словарей при всей их полноте не приводит наречий, которые системно образуются от соответствующих прилагательных.

Фразеологические сочетания. В числе способов выражения причинности грамматика дает и немногочисленные фразеологические обороты: *от полноты души, по старой памяти, ни с того ни с сего, с легкой руки, в сердцах.* Естественно этот список обладает большой неполнотой, но для деловых и технических текстов подобное выражение причины не является характерным. Хотя об этом можно с уверенностью говорить лишь после тщательного анализа. На наш взгляд, их проще отнести к группе наречий, так как они выполняют именно наречную функцию.

Формально неопределяемая причинность. В ряде же случаев (количественных данных нет, это результат обнаружения таких ситуаций при разработке проблемы) причинность в рамках предложения формально не выражена (предлогами, союзами, наречиями, фразеологическими оборотами). Это происходит потому, что при создании текста говорящим или пишущим действуют следующие факторы:

– эллипсис (опущение, умолчание) информации, которую при непосредственном общении можно не создавать, она ясна для собеседников;

– закон экономии речевых усилий, требующий лишь скелетного представления информации по той причине, что понимание идет по образному выражению «с полуслова»;

– существование в начале текста информации, речь о которой идет в середине текста, или прежде упомянутой, связь с которой весьма сложно отыскивать и формализовать;

– пресуппозиция, выражающаяся в том, что мы знаем массу вещей в информационно “свернутом” виде.

Все указанные причины отсутствия формального выражения проявляют себя уже на уровне словосочетаний, нераспространенных предложений, или ядерных композиций. Вот несколько примеров из романа М. Булгакова «Мастер и Маргарита»:

1. *Поведение кота настолько поразило Ивана, что он в неподвижности застыл у бакалейного магазина на углу и тут вторично, но гораздо*

сильнее, был поражен поведением кондукторши.

2. *Говорить ему было трудно. При каждом слове кто-то втыкал ему иголку в мозг, причиняя адскую боль.*

4. *Я так увлекся чтением статей о себе, что не заметил, как она (дверь я забыл закрыть) предстала предо мною с мокрым зонтиком в руках и с мокрыми же газетами.*

5. *После лекарства, напоившего все его тело, успокоение пришло к нему, как волна, накрывшая его.*

6. *Солнце било прямо в кентуриона, не причиняя ему никакого вреда, и на львиные морды нельзя было взглянуть, глаза выедал ослепительный блеск как бы вскипавшего на солнце серебра.*

Причинность на уровне ядерных композиций. Синтаксический уровень является самым сложным из языковых уровней. Эта сложность объясняется тем реальным миллиардным многообразием только ядерных двухэлементных композиций, которое создается сотнями тысяч слов в сотнях грамматических позиций.

В ряде моделей причинно-следственные связи не видны, хотя они и присутствуют в нашем априорном знании. Так, в предложении *Лошадь упала* нас не волнует причина, мы констатируем факт ее падения, а причинами могут быть – скольжение (поскользнулась – причина и упала – следствие), или упала от усталости, от перегрузки.

В атрибутивных комплексах, если атрибут выражен не причастием, причинности нет, ср. *тяжелая папка, веселая вдова* и миллионы подобных. А вот в композициях с двумя глаголами уже может появляться определенный детерминизм в виде следования одного действия только после другого, ср. *Заболел и попал в больницу*, где невозможным является вариант: – *Попал в больницу и заболел*. В больницу попадают уже заболевшие. Необходима тщательная проработка предложений, содержащих два глагола с союзом «И».

Аналогичными примерами будут композиции типа *Знаю – переводил...* и *Переводил – знаю...*, где разница в смыслах проявляется при их "переводе": 'знаю, потому что переводил' и 'переводил, поэтому знаю'. Композиции такого рода выявляют причинно-следственные отношения.

Иной структурный тип композиций выявляет свои особенности в выражении причинности: ср. *Не разобьется, не стеклянное!* и *Не стеклянное, не разобьется!*. Первое предложение «переводится» с помощью союза *потому что*, а второе уже с помощью другого союза – *поэтому*. Здесь мы также выходим на причинно-следственные отношения, которые показывают связь качеств объекта и действий, еще не произошедших, но о результатах которых мы судим заранее. Это очень интересный тип детерминизма, обладающий прогностической функцией.

Примеров такого типа в художественных текстах обнаруживается достаточно много, чтобы предполагать их наличие и в деловой прозе – ср. *Красивая, не забудешь* (поэтому) и *Не забудешь красивая* (потому что), *Не укроешь – пропадет* и *Не*

пропадет – укроешь. В первом примере условие (если не совершить первое действие, то произойдет второе). Во втором утверждается первое действие, потому что с необходимостью должно произойти второе. А в предложении *Не дашь – отниму* отмечается уже другой оттенок смысла – условие. Отношение условия – это необходимое дополнение причинно-следственных отношений и их также следует рассматривать в комплексе.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Ахманова, 1966] Ахманова О.С. Словарь лингвистических терминов / О.С. Ахманова. – М.: Советская энциклопедия, 1966. – 607 с.

[Карпов, 2003] Карпов, В.А. Язык как система / В.А. Карпов. – М.: Едиториал УРСС, 2003. – 304 с.

[Кондраков, 1975] Кондаков, Н.И. Логический словарь-справочник / Н.И. Кондаков. – М.: Наука, 1975. – 720 с.

[Лапутина, 1998] Лапутина, Т.А. Категория причинности и средства ее выражения в современном болгарском и русском языках / Т.А. Лапутина // Автореферат дисс. на соискание ученой степени канд. филологических наук. – Минск, БГУ, 1998. – 18 с.

[Урманцев, 1974] Урманцев, Ю.А. Симметрия природы и природа симметрии / Ю.А. Урманцев. – М.: Мысль, 1974. – 229 с.

[Бондарко, 1986] Уровни языка в речевой деятельности. К проблеме лингвистического обеспечения автоматического распознавания речи – Л.: изд-во ЛГУ, 1986. – 260 с.

[Засорина, 1977] Частотный словарь русского языка / под ред. Л.Н. Засориной. – М.: Русский язык, 1977. – 936 с.

SOME ASPECTS OF SEMANTIC CODING: METHODS OF EXPRESSION OF CAUSALITY IN LANGUAGE-SYSTEM

Golovnja A.I.

*Belarusian State University,
Minsk, Republic of Belarus*

golovnjaai@bsu.by e-mail.address

The article considers some aspects of semantic encoding linked with causality. In this article is proved that the causality in the language system can be linked with prepositions, conjunctions, adverbs and fixed phrases. A certain portion of causality in language does not expressed, but contextually it is present. Partially the causality is illustrated by examples from the novel "The Master and Margarita" by Mikhail Bulgakov. An attempt was made to consider causality at the level of nuclear composition.

СЕКЦИЯ № 5

МЕТОДОЛОГИЯ И МЕНЕДЖМЕНТ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ, ОСНОВАННЫХ НА СЕМАНТИЧЕСКИХ СЕТЯХ, В РАМКАХ OPEN SOURCE ПРОЕКТОВ

METHODOLOGIES AND MANAGEMENT OF DEVELOPMENT OF INTELLECTUAL SYSTEMS ON THE BASIS OF SEMANTIC TECHNOLOGIES WITHIN THE LIMITS OF OPEN SOURCE PROJECTS

Специфику интеллектуализации современных человеко-компьютерных систем определяет расширяющееся внедрение методов и средств управления знаниями в творческо-производственной коллективной деятельности. Актуальность их внедрения обусловлена тем, что прогресс информационных технологий привёл к практическим задачам, решение которых в человеко-компьютерных средах невозможно без оперативного использования коллективных знаний и их моделей. Одним из наиболее дорогостоящих подтверждений того, что игнорировать управление знаниями в коллективной деятельности опасно, является чрезвычайно низкая успешность в разработках систем, интенсивно использующих программное обеспечение.

Ряд ответов на вопросы, связанные с управлением знаниями, представлен в докладах данной секции. В докладах неявно и явно в управлении знаниями выделяется два аспекта – моделирование знаний без акцента на источники их порождения и с акцентом на источник, функции которого выполняет применение опыта в процессах разработки сложной системы.

Более конкретно, доклады затрагивают особенности управления знаниями в российских софтверных компаниях, оценивание качества разработок интеллектуальных систем, модели и методы интеграции информационных и вычислительных ресурсов в формах интернет-порталов и вопросы создания и использования баз опыта проектных организаций.

В докладе Гавриловой Т.А. и Мельникова Н.Е. представлены результаты анализа онлайн-опроса группы софтверных компаний, ориентированного на оценку самого факта использования в компании средств управления знаниями, а также их места и роли в технологических процессах производства программных продуктов.

Анализ позволил выявить проблемы с систематическим планомерным руководством процессами управления знаниями и повышения профессионального уровня сотрудников за счёт обучения сотрудников и заимствования чужого опыта.

В докладе Бахтизина Н.Н. и Неборского С.Н. рассматривается проблема оценки качества интеллектуальных информационных систем (ИИС) в контексте современных международных стандартов в области качества и указывается на целесообразность адаптации стандартов к специфике интеллектуальных систем. Отмечается, что проблема оценки качества ИИС остается нерешенной в силу многогранности ИИС.

В докладе Глобы Л.С. и Новогрудской Р.Л. предлагается подход к построению функционального интернет-портала знаний, ориентированного на представление инженерных знаний с учётом приемлемых характеристик онлайн-доступа.

Особое внимание уделяется вопросам интеграции информационных и вычислительных ресурсов портала, в первую очередь, интеграции онтологической модели, моделей бизнес процессов и описаний элементов на основании формальных теорий. Предложена модель, интегрирующая расчетные задачи в информационную модель портала.

В докладе Маклаева В.А. и Соснина П.И. представлена модель коллективного опыта проектной организации, разрабатывающей семейство автоматизированных систем. Специфику модели определяет репозиторий активов, для формального представления которых используется продукционная и вопросно-ответная логики. Требования и спецификации, вложенные в реализацию модели, согласованы с базовыми стандартами на разработку автоматизированных систем, интенсивно использующих программное обеспечение.

Создание предложенной БАЗЫ ОПЫТА, аккумулирующей модели активов проектной организации, и её рациональное внедрение в проектную деятельность ориентированы на увеличение производительности труда проектировщиков, существенному повышению качества проектов, уменьшению их стоимости. Внедрение БАЗЫ ОПЫТА в процессы проектирования согласовано с практикой разработки семейств систем.

В работе Клещева А.С. и Шалфеевой А.С. исследуется попытка создания единого подхода к автоматизации интеллектуальных процессов в отдельно взятой профессиональной области деятельности. Авторы провели системный анализ и моделирование интеллектуальных процессов и предложили онтологии для основных информационных компонентов, используемых экспертными системами, средствами документирования, системами для обучения, для накопления знаний, интегрируемыми для единых целей.

ВОПРОСЫ, ВЫНОСИМЫЕ НА ОБСУЖДЕНИЕ КРУГЛОГО СТОЛА СЕКЦИИ № 5:

- *Какова целесообразность и перспектива создания и развития технологий проектирования интеллектуальных систем на основе Open Source проектов.*
- *Какова специфика наукоемких Open Source проектов.*
- *Как обеспечить семантическую совместимость проектируемых интеллектуальных систем.*



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПРЕЦЕДЕНТНО-ОРИЕНТИРОВАННАЯ БАЗА ОПЫТА ПРОЕКТНОЙ ОРГАНИЗАЦИИ

Маклаев В.А.* Соснин П.И.**

* *Федеральный Научно-Производственный Центр ОАО «НПО «МАРС»,
г. Ульяновск, Россия*

mars@mv.ru

** *Ульяновский государственный технический университет, г. Ульяновск, Россия*

sosnin@ulstu.ru

В статье представлена модель коллективного опыта проектной организации, разрабатывающей семейство автоматизированных систем. Специфику модели определяет репозиторий активов, для формального представления которых используется продукционная и вопросно-ответная логики. Требования и спецификации, вложенные в реализацию модели, согласованы с базовыми стандартами на разработку автоматизированных систем, интенсивно использующих программное обеспечение.

Ключевые слова: база опыта, вопросно-ответный, прецедент, псевдопрограммирование.

ВВЕДЕНИЕ

На настоящий момент времени наименее освоенным типом инженерной деятельности считают «разработку систем, интенсивно использующих программное обеспечение (Software Intensive Systems, SIS)» [Software, 2011]. В качестве основания для такого утверждения принято приводить чрезвычайно низкую степень успешности (около 30%), которая с 1994 года (каждые два года) регистрируется Корпорацией Standish Group [Reports, 2011] в США и многократно подтверждалась и подтверждается другими исследователями [Charette, 2005].

За проблемой успешности стоят ежегодные потери в сотни миллиардов долларов, что, разумеется, требует анализировать причины негативов и средства их устранения или снижения, а также искать то новое, ещё не применявшееся в разработках SIS, что будет способствовать повышению успеха такой деятельности. В попытках решить проблему уже созданы, проверены и освоены разнообразные полезные средства, к числу которых, например, относится инженерный опыт, представленный в стандартах.

Можно называть и перечислять другие уже предложенные средства, но статистика разработок SIS продолжает подтверждать, что проблема успешности сохраняется и её основной причиной являются негативные проявления человеческого

фактора в условиях коллективной интеллектуальной деятельности.

По глубокому убеждению авторов, одним из принципиальных направлений повышения успешности разработок систем класса SIS, в который входят автоматизированные системы (АС), является **моделирование коллективного опыта**, которое должно сохранять **сущность индивидуального опыта** членов коллектива. В статье представляется текущее состояние разработки комплекса средств «База опыта проектной организации» (в дальнейшем БАЗА ОПЫТА), специфику которого определяют ориентация на прецеденты, их вопросно-ответное моделирование и псевдопрограммирование в условиях создания организацией семейств АС. Комплекс создаётся как открытая информационная система, развитие которой согласовано с управляемым (запланированным) повышением профессиональной зрелости производственных процессов и коллектива проектировщиков.

1. Исходные предпосылки

В проектных организациях, разрабатывающих семейства АС, для достижения успеха открывается возможность для совершенствования не только инвариантных составляющих проектной деятельности, но и специализированных составляющих, обусловленных спецификой семейства. На первый план выходит **«повторное использование (reuse)»**, за осуществление которого

отвечает **уникальный опыт коллектива проектной организации.**

Опыт индивида – это **природно-искусственный феномен**, проявляющий своё существование через действия человека по их успешным образцам в прошлом. За этим феноменом стоит интеллектуальная обработка «условных рефлексов» (как определённого рода экспериментов, experience) для их управляемых повторных применений в будущем. Такое природное образование является динамическим и локализовано в мозговых структурах индивида.

Опыт индивида неотделим от моделей опыта, которые он освоил как член человеческого общества. Именно в этом смысле феномен опыта содержит искусственную составляющую. Модели опыта осваиваются индивидом в процессах взаимодействия с теми их источниками, с которыми он сталкивался в своей жизни. В рассматриваемом случае в число таких источников входит согласованная коллективная профессиональная деятельность членов проектной организации.

Уникальный опыт коллектива проектной организации представляет собой интегрированное образование, включающее индивидуальный опыт членов коллектива и ту совокупность моделей опыта, которая используется в разработке семейства АС, включая модели опыта созданные в такой работе. Легко согласиться с тем, что успешность проектной организации принципиальным образом зависит от (эффективности) интеграции опыта, существующей в организации.

Другими словами, одной из наиболее важнейших составляющих организации коллектива является **связывание индивидуального опыта в коллективный опыт**, для осуществления которого используют **модели опыта** в разных версиях (**язык предметной области семейства продуктов, инструментарий проектирования, онтология проектов, концептуальные схемы, руководства** разных типов, **схемы потоков работ** и другие модели).

В число основных требований к разрабатываемой БАЗЕ ОПЫТА была включена следующая установка: **модель коллективного опыта**, связывающая в единое целое опыт членов коллектива, должна **представлять и интегрировать единицы опыта подобно** тому, как такие единицы представляются и интегрируются в **опыте индивида.**

Если **представление и интеграция единиц индивидуального и коллективного опыта подобны**, то (в рамках подобия) для индивида его **взаимодействие**, как с моделью своего опыта, так и с моделями опыта других индивидов (в рассматриваемом случае членов коллектива), будет осуществляться по **единым схемам**. В этом случае модель коллективного опыта образует **расширенные модели индивидуального опыта.** Вторая

установка, положенная в основу моделирования опыта, связана с **сохранением самого существенного в представлении единиц опыта** как индивидуального, так и коллективного.

В основе опыта лежат прецеденты, с каждым из которых связан интеллектуально обработанный объём определённой активности, моделирующий определённую единицу опыта. Моделирование опыта в мозговых структурах человека является прецедентно-ориентированным

Любой **прецедент** – это **активность** человека или группы лиц, связанная с действием или решением или поведением, осуществлённым **в прошлом**, которая полезна как **образец для повторных использований** и/или **оправдания** повторных действий по такому образцу.

Обобщением вышесказанного с учётом первой установки является следующее требование к интеграции опыта: в **основу модели коллективного опыта**, материализованной в компьютерной среде, должны быть положены **модели прецедентов**, согласованные с моделями прецедентов в мозговых структурах человека.

Прецеденты и их модели обеспечивают повторное реагирование по успешным образцам [Precedent, 2011], что имеет принципиальное значение для проектирования, особенно при создании семейств продуктов (в рассматриваемом случае **семейств АС**).

В моделировании коллективного опыта, учитывающей специфику семейств АС, авторы ориентировались на стандарт FSPLP (V5.0), разработанный в Институте Программной Инженерии (Software Engineer Institute, SEI) Университета Карнеги-Меллон. В этом стандарте для выделения единиц повторного использования используется понятие «**актива**», референты которого могут иметь различную природу.

Особо важным в стандарте является то, что в нём называются и растолковываются те **активности, которые способствуют повышению успеха в создании линеек программных продуктов.** Эти активности, применяя аналогии и адаптацию, можно использовать и для повышения успешности в разработках семейств АС.

2. Средства моделирования опыта

Ориентируясь на прецеденты, к повторному использованию можно подготовить любые активности, полезные в разработках АС. Такая подготовка требует определиться с **моделями коллективной работы** на любом этапе жизненного цикла АС. В современных технологиях разработок SIS и АС коллективная работа обычно представляется в форме **потоков работ**. Наиболее полно и последовательно потоки работ применяются в технологии Rational Unified Process (RUP) [Kroll, 2003].

В основу комплекса средств формирования и использования модели коллективного опыта положена специализированная система потоков работ «**Взаимодействие с опытом**», исполнение каждого из которых в инструментально моделирующей среде WIQA (Working In Questions and Answers) [Sosnin, 2009]. Специфика инструментария WIQA определяет его ориентация на конструктивное использование в процессах решение задач рассуждений вопросно-ответного типа (другими словами, ориентация на диалоговые механизмы решения задач). Можно считать, что WIQA инструментально обеспечивает построение концептуального решения задачи в виде его разложения в базе вопросов и ответов.

Выбор вопросно-ответного базиса для реализации потоков работ «**Взаимодействие с опытом**» обусловлен тем, что сознание человека является **диалогичным по своей природе**, поскольку оно **специализируется на решениях задач доступа к моделям прецедентов**, локализованным в его мозговых структурах. А в соответствии с первой установкой взаимодействия с моделями коллективного опыта должны сохранять самое существенное из взаимодействия индивида с освоенным им опытом.

Компонентная структура базового комплекта инструментария WIQA обобщённо представлена на рис. 1, на котором (без их распределения между серверной и клиентской частями) отражены основные функциональности.



Рисунок 1 □ Компонентная структура базового комплекта

Для создания базы опыта (в зависимости от структуры проектной организации, текущего фронта разработок и существующей корпоративной сети) необходимо развернуть, настроить и связать в единое целое определённое число базовых комплектов. Каждый развёрнутый комплект будет выполнять функции клиент-серверного приложения, обеспечивающего работу определённой группы членов проектной организации. Один из комплектов должен выполнять функцию репозитория базы опыта. Информационная связность комплектов в единое целое обеспечивается за счёт механизмов репликации.

Базовыми структурами данных репликации являются «дерево задач» и «вопросно-ответная

модель задачи», специфика и связность которых отражены на рис. 2.



Рисунок 2 □ Вопросно-ответные структуры данных

Специфика представленных структур определяет атрибутика объектов моделирования, называемых «задачами», «вопросами» и «ответами», которые проявляют себя в процессах формирования, структуризации, систематизации и использования индивидуального и коллективного опыта. За каждой «задачей» стоит выявление соответствующего ей «прецедента» и его интеллектуальное освоение, включающее решение задачи и завершающееся введением «модели прецедента» в «модель опыта», в рассматриваемом случае в базу опыта. Любая «задача» понимается как тип вопроса, ответ на который связан с решением задачи.

За любым «вопросом» стоит обращение к индивидуальному или коллективному опыту, активизированное, в рассматриваемом в статье случае, ситуациями проектирования, затребовавшими обращение к доступному опыту. В этом плане «вопрос» - это природно-искусственный феномен, важнейшей искусственной составляющей которого является его знаковая модель в виде «формулировки вопроса» на языке, сопровождающем процесс проектирования АС. У «вопроса» как у любого природного образования имеются определённые свойства и отношения с его окружением, которые в системе WIQA представляются определённой атрибутикой и операциями над составляющими атрибутами, которые следует понимать как операции над «моделями вопроса». Всё сказанное про представление «вопросов» справедливо для «ответов», поскольку любой «вопрос» с соответствующим ему «ответом» представляют единое «вопросно-ответное целое».

Обобщая вышесказанное, подчеркнём, что структуры данных, представленные на рис.2, вместе с операциями над такими данными выполняют функции **интерфейса** между проектировщиками и их индивидуальным и коллективным опытом, а также опытом, который оперативно ими привлекается из любых полезных источников, находящихся за рамками опыта проектной организации.

Вопросно-ответный интерфейс, обеспечиваемый системой WIQA, активно обслуживает доступ, как к опыту проектировщиков, так и к моделям опыта, вложенным в базу опыта проектной организации. Средства вопросно-ответного интерфейса активно

используются в потоках работ «Взаимодействие с опытом». Именно такое понимание вложено в статью в спецификатор «вопросно-ответный» (Question-Answer, или короче QA).

3. Моделирование прецедентов

3.1. Типовые активы

Одним из основных понятий семейства АС являются «активы», под которыми понимают любые конструкты и средства предшествующих разработок АС, подготовленные для их повторных использований в будущих проектах. К типовым активам относятся следующие их виды:

- реализованные проекты, размещённые в архиве проектов;
- исполняемые проекты, каждый из которых при наличии разрешения на доступ открыт для заимствований из других исполняемых проектов;
- библиотека типовых нормативных документов;
 - типовые архитектурные модели;
 - стандарты и руководящие принципы;
 - образцы;
 - схемы моделирования;
 - структуры программного обеспечения;
 - пакеты исходного текста;
 - выполнимые компоненты;
 - интерфейсные образцы;
 - схемы протоколов;
 - общая модель области, на которой построены другие модели;
 - инструменты, технологии, платформы, инфраструктура.

Эффективность повторного использования представленных конструктов в существенной мере зависит от их представления, включающего идентифицирующую и сопроводительную информацию, например:

- имя, версии и варианты;
 - отношения совместимости с предыдущими версиями;
 - определенные, но не слишком многочисленные отношения с другими активами;
 - идентификатор (или другая версия) адресного входа в каталог с разумными сроками поиска (так, чтобы проектировщики могли найти актив),
 - документация для пользователей (сжатая, но достаточная, особенно относительно того, как специализироваться или параметризовать выбранный актив);
 - образцы тестовых проверок в условиях комплексирования с другими активами.
- Активы принято размещать в специальных репозиториях, которые часто называют «фабриками опыта» (Experience factory) [Tian, 2011], в которых, обычно, унифицируется представление активов в каталоге репозитория. А для каждого типа актива

создаётся специализированная библиотека с собственными средствами доступа.

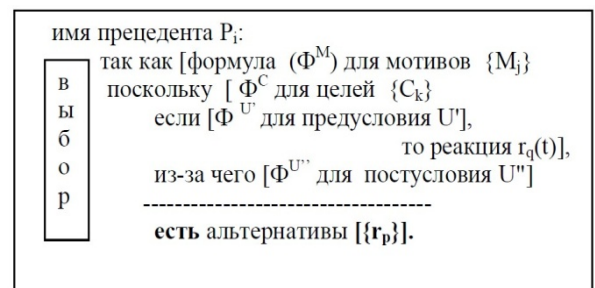
Представляемая БАЗА ОПЫТА также относится к классу «фабрик опыта», но отличается от известных хранилищ активов тем, что унификация в существенной мере распространена за рамки каталогов. Унификация распространена на задачи, которые помогают решать активы и на задачи, которые приходится решать пользователю, применяющему каждый выбранный им актив. Более того, все взаимодействия с базой опыта унифицированы в форме потоков работ, система которых названа «Взаимодействие с опытом». В названных средствах унификации центральное место занимают модели прецедентов.

3.2. Логическая схема прецедента

Специфику моделирования прецедентов для их представления в БАЗЕ ОПЫТА определяют:

- унифицированная логическая схема прецедента, ориентированная на представление и выбор прецедента с учётом его ценностных характеристик, конкретнее с учётом позитивов, которые достигаются от применения прецедента;
 - связывание каждого прецедента с задачей, в результате решения которой была создана его модель;
 - выделение в формировании (интегральной) модели прецедента подчинённых моделей, каждая из которых представляет состояние жизненного цикла прецедента;
 - псевдокодирование программирования, по крайней мере, тех составляющих модели прецедента, в исполнении которых принимает участие человек (проектировщик).

Представление отмеченной специфики начнём с логической схемы прецедента, которая имеет следующий вид:



Ценностное содержание прецедента в этой схеме представляет учёт мотивов и целей, содержание которых фиксируется, например, в следующих формах «повышает точность, за счёт.....», «снижает стоимость, за счёт,», «предотвращает ошибки, обусловленные,». В простейшем виде логическая формула Φ^M или Φ^C может быть подобна списку, с которым сопоставляются ценностные ожидания пользователя.

Одним из применений ценностных характеристик является решение задачи выбора прецедента на

множестве альтернатив. Для обеспечения выбора в состав инструментальных средств включена библиотека методов принятия решений.

3.3. Интегральное представление прецедента

Включение очередной модели прецедента P_i в базу опыта осуществляется в определённый момент времени t_i его жизненного цикла, который начинается с формулировки его исходной постановки задачи $P^T_i(t_{0i})$.

По ходу жизненного цикла постановка задачи изменяется $P^T_i(t_{0i}), P^T_i(t_{1i}), \dots, P^T_i(t_{ki})$, исправляясь и детализируясь, отражая тем самым интеллектуальное освоение прецедента. Общий случай интеллектуального освоения прецедента свяжем с состояниями, а вернее со специализированными моделями прецедента, представленными на рис. 3.

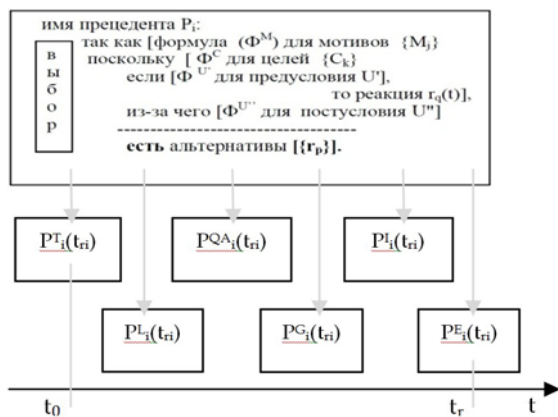


Рисунок 3 □ Состояния жизненного цикла прецедента

4. Псевдокодирование программирование

4.1. Дополнительная атрибутика

В число полезных расширений потенциала QA-данных входит механизм дополнительной атрибутики, предназначенный для расширения атрибутики, вызванного любыми причинами (например, разработка новых плагинов или создание пользовательских QA-программ), причём, такого расширения, в котором не требуется вводить в QA-базу данных новые отношения или изменять уже существующие.

Такое расширение обеспечивает служебный плагин, получивший название «**Дополнительная атрибутика**» (**Additional Attributes, AA**). Разумеется, этот плагин, вводит в базу данных собственные дополнительные отношения, которые открывают возможность для **творческого приписывания** свойств любой единице дерева задач или любой единице QA-протокола, но они используются для эмуляции очередных полезных отношений.

В основу расширения набора (базовых) атрибутов любого из объектов типа «вопрос» или

«ответ» положены механизмы объектно-реляционного преобразования, в результате которых для «объекта», хранимого в базе данных, создаётся его версия, адаптированная к выбранному языку программирования (в рассматриваемом случае к языку C#). В таком преобразовании можно расширить атрибутику и операционную часть (методы объекта).

По сути дела плагин AA предоставляет создателям объектов, использующих его средства, упакованные в объекты (на языке C#) атрибуты реляционных отношений, дополненные специализированными атрибутами. Плагин также обеспечивает связывание «**упакованных атрибутов**» в комплексы, а также предоставляет интерактивный доступ к таким конструктам.

Как отмечалось выше, функциональность плагина определяет набор отношений AA, дополнительно включённых в QA-базу, в частности, для эмуляции очередных отношений (виртуальных отношений), полезных для развития функционального потенциала инструментария WIQA или приложения, созданного с помощью инструментария WIQA. Объектно-реляционное развитие информационного потенциала QA-базы схематично представлено на рис. 4.



Рисунок 4 □ Объектно-реляционное развитие базы данных

Набор отношений AA выбран и материализован из расчёта на объявление оказавшегося необходимы дополнительного атрибута AA_i , возможно включающего подчинённые атрибуты AA_{ij} , с учётом типа и необходимых свойств, причём, принадлежащих не только виртуальному отношению V_k . Созданный атрибут AA_i связывается с определённым узлом дерева задач или вопросно-ответного протокола или узлами тех конструкций, которые QA-данные представляют.

Введение средств дополнительной атрибутики открывает возможность для создания из любого объекта QA-базы «перфокарты», на которой можно «записать» и специфицировать любой декларативный оператор (для объявления переменных любого типа) или любой императивный оператор (например, для представления операторов языка псевдокодов).

4.2. Средства псевдокодowego программирования активностей

Ничто не мешает использовать структуры данных, представленные на рис. 2, для создания с их помощью информационных объектов, представляющих определённые единичные объекты (константы) или их классы (переменные) с помощью необходимой совокупности пар «ответов» и «вопросов».

В этом случае для именования объектов логично использовать «вопросы» (имя ставит вопрос о значении), а для представления их значений - «ответы». В дальнейшем тексте такие информационные объекты будем называть QA-данными, поскольку для их создания используются объекты QA-базы

Семантический потенциал QA-данных (как класса информационных объектов) существенно увеличивается за счёт присоединения к базовой атрибутике QA-данных («уникальный индекс –

адрес», «текстовое описание», «тип», «автор», «время», «состояние завершенности» и другие атрибуты как вопросной, так и ответной частей) необходимых пользователю (проектировщику) дополнительных атрибутов.

Возможности создания и использования QA-данных были проверены в ряде приложений, созданных на базе инструментально-моделирующей среды WIQA, в частности в «Системе многоагентного моделирования окружающей обстановки судна».

Важнейшим результатом таких проверок оказалось то, что опыт, накопленный в использовании QA-данных, привёл к созданию, встроенного в инструментарий WIQA, комплекса средств псевдокодowego программирования, ориентированного на прецеденты. Взаимодействие проектировщика с этим комплексом обобщённо представлено на рис 5.

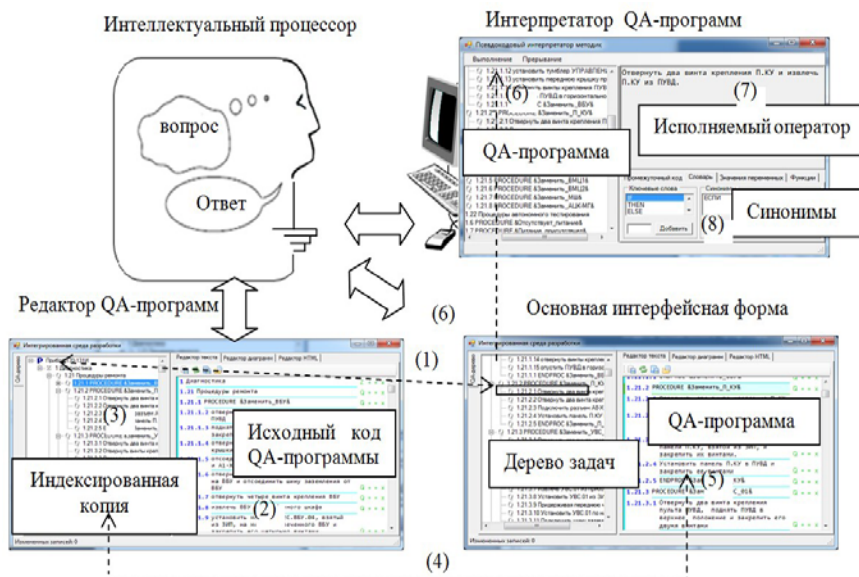


Рисунок 5 □ Инструментальная среда псевдокодowego

Создание в этой среде определённой псевдопрограммы (QA-программы для того, чтобы отличить от псевдопрограмм других типов) начинается с выбора «точки» в дереве задач и объявления новой задачи. Индексное имя новой задачи (1) будет использоваться как начальный адрес для приписывания индексных имён каждому оператору будущей программы, записанному в области (2) текстового редактора. Индексированная копия редактируемой QA-программы регистрируется в рабочей памяти редактора, а её копия (3), форма которой тождественна основной форме визуализации QA-данных, визуализируется в левом окне редактора. После любого сохранения исходного кода в редакторе текущее состояние кода переносится (4) в QA-базу и визуализируется в области (5) основной интерфейсной формы. Любая QA-программа, хранящаяся в дереве задач в любом состоянии, в любой момент времени может быть

загружена в редактор для продолжения её редактирования, исправления ошибок или для адаптации кода.

Любая QA-программа в любом её состоянии может быть загружена (6) в интерпретатор для исполнения. Каждый исполняемый оператор загруженного кода визуализируется в специальной области (7). В любой момент времени проектировщик может ввести (объявить) новый синоним для выбранного ключевого слова.

Реальность проектирования – параллельная работа проектировщика с совокупностью задач. Для обеспечения такой работы в инструментарий WIQA встроена система прерываний, позволяющая прервать любую исполняемую QA-программу и перейти к другой задаче (работе). Система прерываний поддерживает возврат в точку прерываний прерванной QA-программы.

4.3. Образцы QA-программ

В инструментальной среде, представленной на рис. 5, проектировщик специфицируется как «интеллектуальный процессор». Такая спецификация отражает тот факт, что основным исполнителем QA-программы является человек, считывающий оператор за оператором и выполняющий их действия, возможно, какие-то и без компьютера.

«Интеллектуальный процессор» - это «субъект», создающий для решения задач псевдопрограммные модели прецедентов (и их совокупностей) и исполняющий такие псевдопрограммы вместе и согласованно с компьютерным процессором или процессорами. Имитируя роль процессора, человек инициирует переход к очередному оператору псевдопрограммы.

Примером такого вида псевдопрограмм может служить методика (прецедент) «Переустановить Outlook Express», в которой приведена только вопросная часть каждого оператора, помеченная символом «О», а не «Q»:

QA-Program: Установка Microsoft Outlook

- O1. Закройте все программы.*
- O2. На **Запустить** меню, нажмите кнопку **Запустить**.*
- O3. В **Открыть** поле, тип **regedit**, а затем нажмите кнопку **OK**.*
- O4. Переместить и выделите следующий раздел:
HKEY_CURRENT_USER/Software/Microsoft/Office/9.0/Outlook/*
- O5. В имени списка выбор **FirstRunDialog**.*
- O6. **If** требуется включить только **Добро пожаловать в Microsoft Outlook** **Then** Нажмите меню **Правка Изменить***
- O6.1. Введите **Значение true** в поле значение,*
- O6.2. Затем нажмите кнопку **OK**.*
- O7. **If** вы хотите повторно создать все элементы вашему вниманию примеры **Then** переместить и выделите следующий раздел:
HKEY_CURRENT_USER/Software/Microsoft/Office/9.0/Outlook/Setup*
- O8. В списке **Имя** выберите и удалите следующие разделы: **Create Welcome** и **Первый запуск**.*
- O9. В подтверждение удаления параметра диалоговом окне, нажмите кнопку **Да** для каждой записи.*
- O10. Меню в реестр, нажмите кнопку **Выход**.*
- O11. **End***

Ответ на вопрос оператора фиксируется как факт его выполнения (интерпретатор регистрирует символ «*» в текстовом поле соответствующего «ответа»). Для QA-функций интерпретатор в поле «ответа» регистрирует вычисленное значение функции.

Напомним, что псевдопрограммирование, причём в вопросно-ответных формах, нацелено

авторами на управление активностью проектировщика (интеллектуального процессора), создающего и использующего прецеденты. Разумеется, в своей работе, если это полезно, он будет использовать компьютер, но как подчинённый процессор.

Для того чтобы продемонстрировать некоторые детали языка псевдопрограммирования приведём фрагмент ещё одной QA-программы:

```
O 1.11 Procedure &DiscardPriority&
O 1.11.1 &P& := &Pmax&
O 1.11.2 Label &DP1&
O 1.11.3 &Priority& := &P&
O 1.11.4 CALL &GetTaskByPr&
O 1.11.5 &base& -> &TaskPriority& := &base& ->
&TaskPriority& + 1
O 1.11.6 CALL &ChangeTask&
O 1.11.7 &P& := &P& - 1
O 1.11.8 IF &P& < &base& -> &Pmin& THEN &base&
-> &NewPriority& := &Pmin& ELSE GOTO &DP1&
O 1.11.9 ENDPROC &DiscardPriority&
```

Процедура используется в Системе прерываний комплекса WIQA для вычисления приоритета прерванной QA-программы. Система прерываний специально создана с помощью средств псевдопрограммирования. Цель – определить потенциал созданных средств псевдопрограммирования. Задачи такого рода уже не интерпретируются, а компилируются с помощью Компилятора псевдопрограмм, встроенного в инструментарий WIQA.

4.4. Псевдопрограммирование в прецедентах

В псевдопрограммировании прецедентов следует различать кодирование условий доступа и кодирование реактивной части. Общий случай программирования доступа представляет следующая схема:

```
OA-PROGRAM_1 (функциональный тип):
O1. Строка @Переменная V_1 @ Строка,
Комментарий?
A1. Значение V_1.
O2. Строка @Переменная V_2 @ Строка,
Комментарий?
A2. Значение V_2.
.....
ON. Строка @Переменная V_M @ Строка,
Комментарий?
AN. Значение V_M.
ON+1. Выражение_W1 (V_1, V_2, ..., V_M)?
AN+1. Значение выражения W1.
O N+2. Выражение_W2 (V_1, V_2, ..., V_M, W1)?
A N+2. Значение W2.
.....
O N+P. Выражение_WP(V_1, V_2, ..., V_M, W1, ...,
WP-1),?
A N+P. Значение WP. (результат проверки условия)
End
```

Типовая QA-программ «условия» указывает на то, что в общем случае её экземпляр может включать связную совокупность алгебраических и логических выражений, используемых в оценках правомерности и полезности включения прецедента в текущую или предполагаемую активность человека. Так, что на самом деле за «условием» стоит совокупность взаимосвязанных алгебраических и условных функций, причём, одной из разновидностей содержания условных функций является «оценивание».

В то же время, так как формирование и проверки «условия» нацелены в основном на решение задач доступа к «реакции» и её выбора с позиций альтернатив и полезности, то главным для «условия» является устойчивое распознавание адекватного прецедента.

Для решения такой задачи обычно пригодны различные версии «условий» доступа, одна из которых обычно получает преференции и совершенствуется в многократных обращениях к прецеденту (обучается на примерах). А значит, для псевдокодовых программ «условий», исполняемых интеллектуальным процессором, принципиально важным является возможность их оперативного совершенствования (изменения) и исполнения.

Перейдём к псевдокодovому программированию «реакций», исходный псевдокод которой формируется как «процедура», поскольку за реакцией стоит поведение, состоящее из связной совокупности действий. Операторы «реакции» включают в общем случае целенаправленную совокупность действий двух типов, один из которых требует активности интеллектуального процессора, а второй – активности компьютерного процессора.

Типовая схема смешивания действий, исполняемых процессорами двух типов, имеет следующий вид:

```
QA-PROGRAM_2 (процедурный тип):
O1. K_i, K_j, ..., PL_k ?
A1. *
O2. K_m, QA-P_n, ..., K_q?
A2. *
.....
ON. K_s, PL_t, ..., QA-P_v?
AN. #
End.
```

где K_i – команда, активизируемая пользователем в акте интерактивного взаимодействия с компьютером; PL_k – вызов пользователем плагина и переход к его функционалу; $QA-P$ – вызов QA-программы.

4.5. Структура БАЗЫ ОПЫТА

В структурном плане БАЗА ОПЫТА представляет собой связную совокупность (Рис. 3) сетевых приложений WIQA.Net, каждое из которых обслуживает работу определённой группы

проектировщиков [Соснин, 2010]. Связность обеспечивается как на уровне взаимодействий между группами, так и за счёт доступа к общему для всех групп репозиторию.

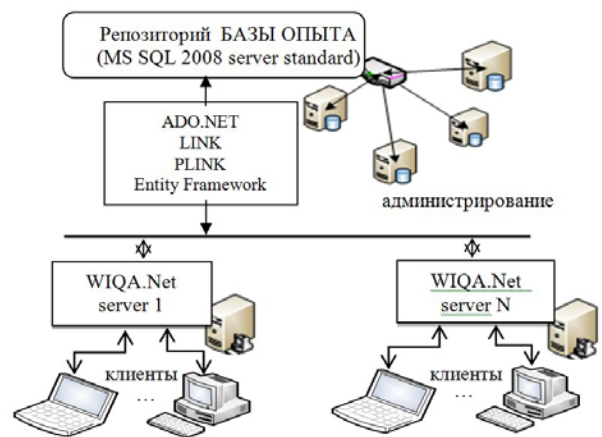


Рисунок 6 □ Структура БАЗЫ ОПЫТА

Каталог активов (с учётом их типологии) в репозитории реализуется с помощью связной совокупности моделей, базовый тип которых приведён на рис. 2. В информационных потоках используются механизмы репликации и синхронизации. Типовой единицей обмена между репозиторием и серверами групп является представление актива, с каждым из которых связана «задача» и соответствующая задаче псевдопрограмма или связная совокупность псевдопрограмм.

5. Заключение

Создание предложенной в статье БАЗЫ ОПЫТА, аккумулирующей модели активов проектной организации, и её рациональное внедрение в проектную деятельность способно привести к увеличению производительности труда проектировщиков, существенному повышению качества проектов, уменьшению их стоимости. Внедрение БАЗЫ ОПЫТА в процессы проектирования согласовано с современными стандартами и практикой разработки семейств систем, которая подтверждает достижимость отмеченных позитивов.

К специфике БАЗЫ ОПЫТА относится комплексирование требований, извлекаемых из российских и международных стандартов для систем, интенсивно использующих программное обеспечение, а также применение для комплексирования инструментария вопросно-ответного моделирования задач.

В настоящий момент времени работы по созданию БАЗА ОПЫТА продолжаются. Основные проектные решения, которые вложены в БАЗУ ОПЫТА, прошли проверку на задачах концептуального проектирования АС, проектного документирования, информационной безопасности, контроля и диагностики изделий, технологической

подготовки производства, а также на предметных задачах многоагентного моделирования и экспертного мониторинга окружающей обстановки морского судна.

Потоки работ «Взаимодействий с опытом» прошли этап опытной отладки. Сборка и испытания БАЗЫ ОПЫТА как целого запланированы на 2012 год.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Соснин, 2011] Соснин, П.И. Вопросно-ответное программирование человеко-компьютерной деятельности / П.И. Соснин – Ульяновск: УлГТУ, 2010. – 240 с.

[Charette, 2005] Charette, R.N. Why software falls // IEEE Spectrum, vol. 42, #9, pp. 36-43, 2005.

[Kroll, 2003] Kroll, P. and Ph. Kruchten, The Rational Unified Process Made Easy: A Practitioners Guide to the RUP / Addison-Wesley, 2003.

[Precedent, 2011] Precedent. – Режим доступа: <http://dictionary.reference.com/browse/precedent>. – дата доступа: 18.11. 2011

[Reports, 2011] Reports of the Standish Group – Режим доступа: <http://www.standishgroup.com>. – дата доступа: 18.11. 2011.

[Software, 2011] Software Intensive systems in the future // Final Report//ITEA 2 Symposium, – Режим доступа: http://symposium.itea2.org/symposium2006/main/publications/TNO_IDATE_study_ITEA_SIS_in_the_future_Final_Report.pdf – дата доступа: 18.11. 2011

[Sosnin, 2009] Sosnin, P.I. Means of question-answer interaction for collaborative development activity // Hindawi Publishing Corporation, Advances in Human-Computer Interaction. – 2009 – Vol.2009, Article ID 619405, 18 pages

[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.

[Tian, 2011] Tian, J. An Emerging Experience Factory to Support High-Quality Applications Based on Software Components and Services// Journal of Software, Vol. 6, No. 2, 2011, pp. 14-32.

PRECEDENT-FOCUSED BASE OF EXPERIENCE OF THE DESIGN ORGANIZATION

V.A. Maklaev*, P.I. Sosnin**

*Federal Science-Production Centre “MARS”,
Ulyanovsk, Russia*

mars@mv.ru

*Ulyanovsk State Technical University,
Ulyanovsk, Russia*

sosnin@ulstu.ru

In paper the model of collective experience of the design organization developing the family of automated systems is presented. Specificity of the model is defined by a repository of reuse actives for which the formal representation of precedents and question-answer logic are used. Requirements and the specifications enclosed in realization of the model are coordinated with basic standards on the development of the automated systems intensively using the software.

INTRODUCTION

Nowadays the most problematic area of computer applications is “Development of Software Intensive Systems (SIS)”, within the frame of which the collaborative works of developers and other stakeholders are being fulfilled in corporate networks. **“A software intensive system is a system where software represents a significant segment in any of the following points: system functionality, system cost, system development risk, development time”.**

The significant number of SIS developments (about 65 percent) either is being stopped, or is exceeding planned time and/or finance, or reach the end in the poorer version. Such situation indicates that developers have not received very important means for successful developing the SIS till now.

In accordance of the author our opinion one of the problems of successful designing is caused by natural restrictions of designer intelligence which can be overcome with the help of adequate modeling of the human experience adjusted to the collective designing in the corporate network.

By the other words the developers of SISs need the effective means for adequate defining of the essential experience units, their modeling for achieving the necessary understanding and also for testing the units in appropriate conditions of designing and using. The named experience units are to be distinguished, defined, modeled, understood, coded and tested as precedents. **“Precedents are actions or decisions that have already happened in the past and which can be referred to and justified as an example that can be followed when the similar situation arises”**

In article the precedents models are used for the creation and usage the base of experience models of the organization developing the family of SISs. The specificity of the suggested solutions is being defined by the question-answer approach (QA-approach) to the work with the experience and its models. This work supports by the special instrumental system named WIQA (Working In Questions and Answers). The choice of the QA-approach to the experience base is explained that consciousness has a dialog nature the existence and process of which are being opened via the usage of the natural language.

MAIN PART

The positive changes in the development of the SIS can be connected with the creation and usage of means for programming the designer interactions with the experience and its models in the conceptual designing. The degree of the successfulness in the conceptual designing will be positively increased if any designer will play the role of the “processor” which executes the programs managing his(her) activity in the appropriate instrumental medium. Such role of the designer was named “intellectual processor”. This role is being supported by means of WIQA.

The system WIQA has been developed as the QA-processor for the conceptual designing of the SIS by the method of conceptual solving the project tasks. This method is based on the stepwise refining and QA-reasoning which are being evolved in the frame of incremental designing.

The base component of WIQA is the QA-database supported the real time work of designers with solving tasks of designing for the current project. All tasks are combined in the tasks tree with the visual access of designers to its units. Each such unit is interpreted as the definite precedent used in design process or will be executed (in the future) by the user of the SIS.

The conceptual solution of any task Z_i is being built with the help of QA-reasoning (sequence of questions $\{Q_{ij}\}$ and answers $\{A_{ij}\}$) which are being registered in the QA-protocol interpreted as the QA-model of the corresponding task. Thus interactive tasks tree and a set of corresponding interactive QA-models of tasks present the “intermediate” between developers and the current state of the designed SIS existed in the computer environment.

This intermediate link opens opportunities for the realization of new forms human-computer interactions which are additional for forms of interactions put into the practice now. New forms are oriented to precedents and have the question-answer type. QA-means for the implementation of such forms consist of declarative and imperative parts.

The declarative part is based on data models named QA-data which present the tasks tree, QA-models, Z-, Q-, A-objects and all useful transformations of these units supported in WIQA. QA-data is their interpretation from the informational point of view.

The specificity of QA-data is defined by typical forms of Z-, Q- and A-objects each of which includes the numerous diversity of attributes (for example, index-name, type of object, creator, time of creation and many others) and the hierarchical combining of such objects. The central place among attributes occupies the attribute “textual description” presenting the content of the corresponding object (“task”, “question” or “answer” of the definite type).

The imperative part consists of a set of commands (QA-commands) with QA-data, their useful sequences and more complicated behavioral units from QA-commands which are specified as a kind of QA-programs. For this reason WIQA is constructed as the specialized QA-processor.

Most essential feature of the QA-approach is the possibility of programming (preliminary or in the real time) of human-computer interactions (HCI). Moreover, this approach assumes that necessary means of HCI are being embedded to the definite model of the precedent. For supporting such possibility the means of QA-modeling have been evolved till their usage for pseudo-programming oriented on precedents and the execution of pseudo-programs by the intellectual and computer processors collaboratively.

One direction of broadening the interpretation of QA-data is defining the abstract type of data with named attributes and features including the accessible set of commands. Such interpretation allow to developers to use the abstract QA-type for the emulation of the types of data which are needed for pseudo-programming.

For solving the emulation tasks in WIQA there is a special mechanism for assigning the necessary characteristics to the definite unit of QA-data. It is the mechanism of additional attributes (AA) which gives the possibility to expand the set of basic attributes for any Z-, Q- or A-object keeping in the QA-database.

The mechanism of AA implements the function of the object-relational mapping of QA-data to programs objects with planned characteristics. One version of such objects is classes in C#. The other version is fitted for pseudo-code programming.

Broadening of the abstract type of QA-data by means of additional attributes helps to emulate any traditional data types such as scalars, arrays, records lists and the others. Moreover, means of additional attributes open the possibilities for assigning to simulated data their semantic features.

The set of basic operators includes traditional pseudo-code operators but each of which inherits the feature of the appropriate QA-units also. Hence, the basic attributes of the QA-unit and necessary additional attributes can be taken into account in processing the operator and not only in its translation. In order to underline the specificity of the operator emulation they will be indicated as QA-operators.

Means for QA-programming have been developed and embedded to WIQA as its evolution. They include the special editor, pseudo-code interpreter and compiler. The reality of the designer activity is a parallel work with many tasks at the same time. Therefore the special system of interruptions is included into WIQA.

CONCLUSION

Told above contains sufficient arguments to assert that the real time pseudo-programming of interactions of designers with experience and its models embedded to the base of experience leads to many positive effects in the usage of SISs and their development. QA-programming is the rational way for such work which can be implemented with the help of WIQA means. QA-programming of precedents can be implemented at the project level (as the creation of the tasks tree) and at the pseudo-code level (as writing QA-programs for intellectuals and computer processors).

QA-programs are useful means of HCI which are additional for traditional means of HCI. Such means of HCI are adjusted for the access to the human experience in the precedents forms which were used in creating the library of the usability metrics implemented as the set of tasks with embedded interfaces precedents.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89

ОНТОЛОГИЯ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Клещев А.С., Шалфеева Е.А.

*Институт автоматизации и процессов управления,
г. Владивосток, Россия*

kleshev@iacp.dvo.ru

shalfe@iacp.dvo.ru

В работе исследуется возможность единого подхода к автоматизации интеллектуальных процессов в отдельно взятой профессиональной области деятельности. Проводится системный анализ и моделирование интеллектуальных процессов. Предлагаются онтологии всех информационных компонентов, используемых экспертными системами, средствами документирования, системами для обучения, для накопления знаний, интегрируемыми для единых целей.

Ключевые слова: знания, интеллектуальный процесс, профессиональная деятельность, онтологии информационных компонентов.

ВВЕДЕНИЕ

Достигнуты значительные успехи в автоматизации коммерческих предприятий, в разработке информационных и других систем для более эффективного управления бизнесом [Битек, 2007]. Однако не создано инструментария, позволяющего комплексно автоматизировать сферы экономики, процессы в которых связаны с интеллектуальной деятельностью и использованием постоянно обновляемых знаний. К таковым относятся многие сферы финансируемые государством, - медицинское обслуживание, образование, наука, оборонная деятельность и др. К настоящему времени используются разрозненные подходы к автоматизации в таких сферах, как, например, медицинская, хотя ведено и стандартизировано такое понятие, как *электронная история болезни* [ГОСТ, 2008]. При автоматизации лечебно-диагностических процессов отдельными средствами автоматизируют взаимодействие между участниками лечебно-диагностического процесса, с акцентом на документирование всех шагов этого взаимодействия. Лечебно-диагностический процесс рассматривают как бизнес-процесс, как технологию информационного взаимодействия, аналогичного любым другим управленческим процессам.

Задачи автоматизации интеллектуальной деятельности постановки диагноза, назначения и прогнозирования лечения и т.п., если и решаются, то другими средствами. При этом есть проблемы с их полноценным использованием (часто приходится

упрощать процессы), есть проблемы с их сопровождением.

К системам, разрабатываемым для профессиональных интеллектуальных деятельностей государственного значения, предъявляются требования обеспечения всех специалистов адекватными средствами, повышающими их качество работы. Эти средства должны быть интегрированы друг с другом по управлению и по информации, должны быть адаптивными и управляемыми в силу влияния на деятельность специалистов внешних (таких как изменения в законодательстве), и внутренних факторов (непрерывного усовершенствования знаний).

Целью исследования является идентификация интеллектуальных видов деятельности в отдельно взятой достаточно сложной области профессиональной деятельности, системный анализ и моделирование интеллектуальных процессов, разработка онтологии всех используемых информационных компонентов, обеспечивающей их повторную использование при создании сообществ экспертных систем, систем для обучения, средств накопления знаний и других инструментов.

1. Идентификация интеллектуальных видов деятельности (на примере медицинских)

Виды профессиональных деятельностей, не затрагивающие обучение профессии и научную

составляющую, обычно представляют тремя группами: *основные процессы, процессы управления, обеспечивающие процессы.*

Традиционно (в рамках автоматизации коммерческих предприятий медицинской сферы) к основным процессам относятся те, исполнители которых - лечащий врач и регистратор / администратор [Битек, 2007, Каштальян, 2007].

При рассмотрении *интеллектуальных деятельностей* (требующих принятия ответственных решений) целесообразно обратиться к известной классификации.

Традиционно при разработке систем, основанных на знаниях, выделяют следующие классы интеллектуальных задач: задачи *классификации, диагностики, интерпретации, мониторинга, задачи управления, планирования, прогноза, задачи ремонта, задачи проектирования.* Кроме того, включают сюда и такие виды задач как: *поддержка принятия решения; обучение.*

К интеллектуальным (и тесно связанным с ними) задачам медицинской деятельности в рамках текущего исследования отнесены следующие.

Из числа так называемых «основных, интеллектуальных»:

- *установление диагноза* пациента (это пример задачи диагностики);
- *назначение лечения* пациенту (это пример задачи ремонта);
- *до-обследование* (это пример задачи «получения недостающей информации», т.е. планирования того, за какими параметрами надо понаблюдать, и в какие моменты времени);
- *планирование контрольных обследований и очередного посещения* врача (тоже);
- *коррекция лечения*, которая включает в себя коррекцию назначения лечения, если прежняя схема лечения не дала ожидаемых изменений состояния, и коррекцию диагноза, если изменилось представление о состоянии пациента, (это пример задачи управления на основе результатов задачи мониторинга);
- *прогнозирование* лечения / развития болезни (это пример задачи прогноза) и
- *выполнение функциональной диагностики* (это пример задачи измерения вместе с задачей интерпретации).

Автоматизация этих деятельностей состоит в построении соответствующих ЭС. Программно-технические системы реального времени для проведения функциональной диагностики остаются за рамками автоматизации интеллектуальной деятельности, однако вербальное представление их результатов врачами должно быть рассмотрено как часть автоматизируемого интеллектуального процесса – это вербальное представление является

входной информацией для других автоматизируемых деятельностей.

Из числа «обеспечивающих, интеллектуальных» задач:

- обучение проведению обследования (осмотра),
- обучение интерпретации результатов функциональной диагностики,
- обучение диагностике (и, возможно, необходимому для нее планированию до-обследования),
- обучение назначению лечения.

Автоматизация деятельности состоит в построении тренажеров, проверяющих знания.

Из числа «основных, неинтеллектуальных» - задача ведения истории болезни (это пример задачи документирования).

Задача выполнения инструментального обследования (пример задачи измерения) здесь интересна с точки зрения представления результатов в вербальном виде.

Автоматизация деятельности состоит в построении АРМов.

Из числа научных (т.е. «интеллектуальных»):

- формирование знаний о диагностике заболевания,
- формирование знаний о схеме лечения заболевания,
- формирование знаний о лечебных мероприятиях.

Автоматизация деятельности состоит в построении редакторов знаний и программ индуктивного формирования знаний (частный случай математического моделирования).

2. Взаимосвязи подзадач в интеллектуальной профессиональной деятельности

Упрощенная схема взаимосвязи основных медицинских деятельностей представлена на рис.1.

В этой схеме показаны некоторые задачи, решаемые на протяжении периода взаимодействия пациента со специалистами медицинского учреждения, вырабатываемые документы (пунктиром) и другая информация (в точечных обозначениях).

Многие задачи из числа вышеперечисленных на сегодняшний день автоматизированы, для них созданы поддерживающие принятие решений ЭС (или исследовательские прототипы) в соответствии с классами интеллектуальных задач (диагностики, ремонта и т.д.).

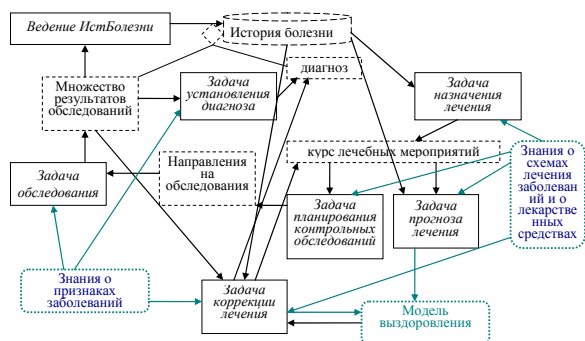


Рисунок 1 – Упрощенная схема взаимосвязи деятельности.

Однако автоматизация интеллектуальной профессиональной деятельности в комплексе требует интеграции поддержки принятия решений всех интеллектуальных и связанных с ними задач, а также задач обучения и формирование знаний.

Поддержка принятия решений специалистов-медиков на протяжении периода выздоровления пациента с точки зрения «автоматизации в комплексе» наиболее близка задачам теории управления сложными системами [Грибова 2010].

Если обратиться к терминологии, выработанной теорией управления, то объектом управления оказывается **пациент**, целью управления становится **отсутствие** (устранение) **заболевания**, она достигается за счет **управляющих воздействий – лечения** (которое пациенту назначено и которое он с большой вероятностью проведет). **Управление – выработка** этих **управляющих воздействий**, включающая мониторинг состояния (обследование и до-обследование), обработку информации (диагностика), принятие решений (назначение лечения).

Система управления состоит из управляющего объекта (медперсонала) и объекта управления (пациента). *Функцией управляющего объекта является совокупность его действий (скорее, не «однородная по некоторому признаку») – сбор информации, установление диагноза пациента, назначение лечения, планирование, коррекция, прогнозирование лечения\развития, обследование, ведение истории болезни, - подчиненных общей цели управления.*

Упрощенная схема управления состоянием пациента представлена на рис.2. У **пациента происходят процессы в организме, показателями которых являются значения наблюдений признаков** в моменты времени (они составляют дневник наблюдений – важную часть **истории болезни**). На основе значений наблюдений и собственных знаний о признаках заболеваний и их динамике врач ставит диагноз; далее на основе информации об особенностях пациента и собственных знаний о схеме лечения заболевания и знаний о лекарственных средствах и других лечебных мероприятиях готовит план лечебных мероприятий, которые должны воздействовать на больного – вести его к выздоровлению. В процессе

выздоровления осуществляется слежение за изменением состояния пациента и, в случае отклонений от прогнозируемого изменения, могут быть скорректированы лечебные мероприятия или даже диагноз.

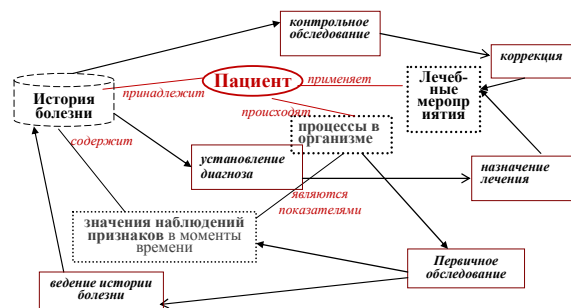


Рисунок 2 – Упрощенная схема управления объектом.

Примечание. Задачи обучения врачей и формирования новых знаний непосредственно к схеме управления состоянием пациента не относятся (скорее, к управлению медперсоналом). Но получаемые с их помощью знания являются важной информационной составляющей всего комплекса интеллектуальных профессиональных задач. Знания, приобретаемые в процессе обучения, являются основой правильных решений врачей. А наличие формализованных знаний повышает возможность каждого отдельного врача знать как можно больше (владеть более глубокими и/или широкими знаниями). Индуктивное формирование знаний приводит к новым знаниям о признаках заболеваний - по архивам историй болезни; к новым знаниям о схемах лечения - по протоколам корректировок планов лечебных мероприятий; к новым знаниям о лекарственных средствах\мероприятиях – по фрагментам дневников наблюдений от начала применения этих средств\мероприятий.

3. Структура и характеристики интеллектуальных деятельностей (на примере медицинских)

Некоторые из деятельностей могут рассматриваться как составные (состоящие из множества задач):

$$\text{Основная деятельность} = 1..*\{\text{задача}\},$$

$$\text{где задача} = 1..*\{\text{под}|\text{задача}\}.$$

Любая интеллектуальная деятельность связана, как правило, с использованием существующих знаний и может рассматриваться как получение искомой/результатирующей информации на основе имеющейся входной. Особенность интеллектуальных деятельностей медицинской сферы такова, что они подразумевают возможность, а иногда и необходимость консультирования, главным результатом которого является *объяснение* результата деятельности. Это объяснение является дополнительным входным данным специалисту в

его решении. Автоматизация консультирования связана с формализацией знаний (для выполняемой деятельности) и их использованием при построении объяснения.

Поэтому структура *основной интеллектуальной деятельности* (или каждой из ее подзадач) может быть представлена так:

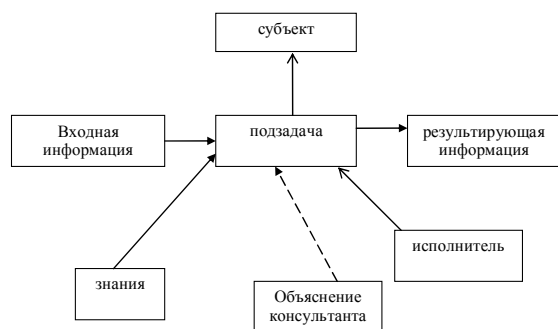


Рисунок 3 – Структура интеллектуальной деятельности.

Пунктирная стрелка означает, что «объяснение консультанта» может использоваться для принятия решения ответственным специалистом.

Для «основных» деятельностей и подзадач медицинской сферы (*диагностика заболевания, назначение лечебных мероприятий, прогнозирование выздоровления, коррекции лечения*) основным субъектом является пациент, а информационным компонентом (входным и результирующим) является сложно устроенный документ *история болезни пациента*. Содержимое этого документа расширяется в процессе применения различных подзадач медицинской деятельности.

Схематически **история болезни** может рассматриваться так:

- паспортная часть +
- особенности пациента +
- жалобы +
- дневник наблюдений
- [+ 0..N (диагноз консультанта)]
- + диагноз
- + план лечения
- [+ дневник лечения].

Дневник наблюдений = 0..N {наблюдение}

- Наблюдаемый признак =
- момент +
- значение наблюдения.

Значение наблюдения =

- результат осмотра (цвет, величина...)|
- результат объективного инструментального измерения (*пульс, давление, анализ крови*) |

результат функциональной диагностики (флюорография, УЗИ).

Дневник лечения =

0..N {контрольная точка в лечении}.

Контрольная точка в лечении =

- момент +
- [+ отклонения в выполнении плана лечения]
- + фрагмент дневника наблюдений для текущего момента.

Структура подзадачи «диагностика заболевания» до автоматизации такова (значком % помечены необязательные составляющие).

Субъект: **пациент;**

Входная информация:

история болезни (уточненная значениями потребованных лабораторных исследований);

Знания:

Персональные:

Теоретические знания о признаках заболеваний и их динамике;

%Собственная база прецедентов признаков заболеваний;

Общедоступные: методическая литература (книга, инструкция),

Исполнитель: **врач;**

%Объяснение консультанта: объяснение диагноза, выполненное консультантом (другим врачом, консилиумом);

Результирующая информация: **диагноз.**

При автоматизации этой деятельности (разработке и внедрении средства просмотра формализованных знаний и диагностической ЭС) структура подзадачи «*диагностика заболевания пациента*» становится такой:



Рисунок 4 – Структура автоматизируемой интеллектуальной деятельности.

В том случае, когда разрабатывается «автоматизированный консультант» (ЭС) в помощь

специалисту, «объяснение» формируется автоматически.

Знания:

персональные: Теоретические знания о признаках заболеваний и их динамике (возможно существование а врача собственной «базы прецедентов признаков заболеваний и их динамики»;

общедоступные:

формализованные: Компьютерная база знаний о заболеваниях и нормах;

текстовые: методическая литература.

Объяснение консультанта: **объяснение диагноза,** выполненное диагностической экспертной системой.

Инструмент или средство:

компьютерная база знаний о заболеваниях и нормах,

диагностическая экспертная система.

Аналогично представляется структура подзадач *назначение лечебных мероприятий, прогнозирование выздоровления, коррекция лечения,* подобным образом - *до-обследование и выполнение функциональной диагностики.*

4 Онтологии информационных компонентов деятельностей

Информационными компонентами интеллектуальных деятельностей медицины являются:

история болезни пациента,

формализованные знания о заболеваниях,

формализованные знания о схемах лечения заболеваний,

формализованные знания о лекарствах и других лечебных мероприятиях,

объяснение диагноза,

лечебные мероприятия (план лечения),

объяснение плана лечения,

объяснение коррекции лечения.

«История болезни» может включать в себя не только паспортную часть, особенности пациента, его жалобы на первичном приеме, результаты внешнего осмотра врачом на первичном приеме, но и дневник наблюдений (жалобы и результаты внешнего осмотра врачом на каждом повторном приеме), диагноз, а возможно еще и объяснение диагноза консультанта, план лечения, дневник лечения и план контрольных посещений врача, план дообследования.

Таким образом: история болезни =

паспортная часть +

особенности пациента +

жалобы +

дневник наблюдений

[+ диагноз]

[+ лечебные мероприятия]

[+ дневник лечения]

Согласно национальному стандарту РФ «ЭЛЕКТРОННАЯ ИСТОРИЯ БОЛЕЗНИ» от 2008-01-01 каждая персональная медицинская запись включает в себя:

дату и время события (осмотра пациента, проведения манипуляции, забора биоматериала для анализа и др.),

автора данной медицинской записи,

медицинское содержание (результат анализа или обследования, статус, эпикриз, назначение лекарств и т.д.) - текст или файл (медицинские изображения, графические материалы, тексты в различных форматах и т.д.) или построенные формализованные данные, позволяющие производить отбор и фильтрацию, проводить статистическую обработку, формировать отчеты.

Поэтому *дневник лечения* охватывает информацию о *проведении манипуляций\процедур, а дневник наблюдений* – не только информацию о *функциональной диагностике*, но и о *результатах обследования узкими специалистами.*

Для лечения и для накопления фактического материала для будущих обобщений важен такой дневник лечения, который включает правдивое перечисление того, чем и в каком количестве, в каком режиме на самом деле лечился пациент.

*Дневник наблюдений / лечения = * {запись}*

Запись =

момент +

дата и время записи,

автор записи,

медицинское содержание.

Примеры области значений *медицинского содержания записи:* строковое - *результат осмотра (цвет...)*, числовое значение - *результат объективного инструментального измерения (пульс, давление)*, таблица - *анализ крови*, изображение и текст - *результат функциональной диагностики (флюорография, УЗИ).*

Для профессиональных деятельностей (таких, как медицина), характерно использование единой терминологии при представлении данных и знаний. Явное представление ее в отдельном ресурсе, от которого зависят ресурсы-данные и ресурсы-знания, гарантирует и согласованность при сотрудничестве

всех специалистов, и согласованность элементов данных, которыми оперирует прикладная логика программных компонентов, автоматизирующих или поддерживающих деятельность всех этих специалистов.

Онтология (структура) всех информационных ресурсов (документов) должна обеспечивать их эффективное использование экспертными системами, системами для обучения, средствами накопления знаний, средствами документирования, интегрируемыми для единых целей.

Один из языков, ориентированных на единообразное представление информационных ресурсов и их онтологий для создаваемых сообществ программных систем, - декларативный язык, разрабатываемый в ИАПУ ДВО РАН.

Структура информационного ресурса, представляющего все виды наблюдений на состоянии пациента может быть представлена как на рис.5.

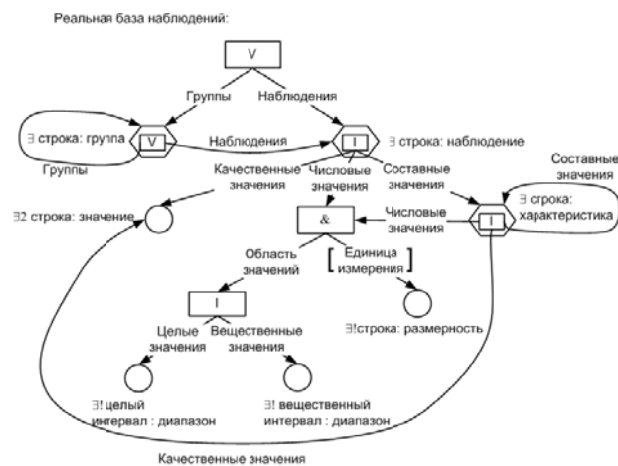


Рисунок 5. Онтология наблюдений в медицине.

Контекстно-зависимое представление структуры *дневника наблюдений / лечения* на этом языке выглядит так:

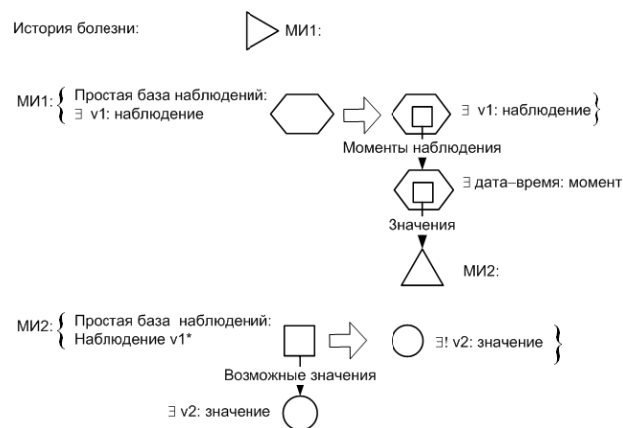


Рисунок 6. Фрагмент онтологии истории болезни.

ЗАКЛЮЧЕНИЕ

Построенные модели интеллектуальной профессиональной деятельности и онтологии

необходимых ей информационных ресурсов являются основой концепции комплексной автоматизации процессов в медицинской области. Показано, что совокупность решаемых задач и процессов поддержки принятия решений на протяжении периода достижения главной цели (выздоровления пациента) близка теории управления сложными системами.

Построенные модели и онтологии станут основой для формирования *методологии* системного анализа и моделирования произвольных сфер деятельности с интеллектуальными процессами управления. В свою очередь методология даст возможность построить единую *онтологию профессиональной деятельности* специалистов, на базе которой могут быть разработаны технология и инструментарий автоматизации интеллектуальной профессиональной деятельности.

Работа выполнена при поддержке грантов РФФИ № 10-07-00089-а и ДВО РАН № 12-III-A-01И-006.

Библиографический список

[Битек, 2007] «Процессно-организационная бизнес-модель стоматологической клиники, занимающейся лечением зубов, пародонто, зубопротезированием, имплантологией, исправлением прикуса и профилактикой.»/ материалы открытого семинара-практикума «Совершенствование системы управления предприятием на основе описания и оптимизации бизнес-процессов», 5-7 ноября 2007 г., компания «БИТЕК», Москва // Информационный портал *Betec.Ru*. <http://www.betec.ru/secure/index.php?id=5&sid=14&tid=03>.

[ГОСТ, 2008] «Электронная история болезни. Общие положения» (). / ред. Гелемеева О.В. // ГОСТ Р 52636-2006. Дата введения 2008-01-01.

[Грибова, 2010] Грибова В.В., Клещев А.С., Шалфеева Е.А. Управление интеллектуальными системами // Известия РАН. Теории и системы управления. 2010. № 6. С. 122-137.

[Каштальян, 2007] Каштальян А.А. Анализ затрат труда врачей амбулаторно-поликлинического приема (по материалам хронометражного исследования) // Журнал «Медицинские новости», 2007, С. 71—74. - <http://www.mednovosti.by/journal.aspx?article=204>.

PROFESSIONAL ACTIVITY ONTOLOGY Kleshev A.S., Shalfееva E.A.

*The Institute of Automation and Control
Processes, Vladivostok, Russia*

kleshev@iacp.dvo.ru

shalf@iacp.dvo.ru

The possibility of the uniform approach to automation of intellectual processes in some professional sphere is investigated in this work. The system analysis and modeling of intellectual processes is carried out. The ontologies of all information components used by expert systems, documentation programs, coursewares, knowledge-mining systems, integrated for the common purposes, are offered.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 65.01

ОБ ОСОБЕННОСТЯХ УПРАВЛЕНИЯ ЗНАНИЯМИ В РОССИЙСКИХ СОФТВЕРНЫХ КОМПАНИЯХ

Гаврилова Т.А.* , Мельников Н.Е.**

* *Высшая школа менеджмента СПбГУ, г. Санкт-Петербург, Россия*

tgavrilova@gsom.pu.ru

** *Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»,
г. Санкт-Петербург, Россия*

melnikov.n.e@gmail.com

В работе разбираются основные особенности компаний, работающих в сфере информационных технологий, в части управления знаниями. Исходные данные были получены в результате онлайн-опроса на популярном тематическом Интернет-ресурсе. Также был произведен анализ российских публикаций, затрагивающих данную проблематику.

Ключевые слова: базы знаний, софтверные компании, управление знаниями.

ВВЕДЕНИЕ

Отрасль информационных технологий (ИТ) по праву можно назвать одной из самых динамичных и бурно развивающихся. Постоянное появление новых решений, опережение фактического применения ИТ перед нормативным регулированием, инновационный характер разработок, положительный рост рынка – всё это доказывает стремительность, с которой развивается данная сфера. Компании, работающие на этом рынке, постоянно имеют дело с изменяющейся внешней средой и претерпевают постоянное развитие. Управление знаниями, как один из подходов к управлению организацией, как нельзя лучше подходит для подобных областей, где условия работы крайне непостоянные, а актуальность информации и знаний особенно важна. В данной статье затрагиваются вопросы и проблемы управления знаниями в российских компаниях, занимающихся разработкой программного обеспечения.

1. Управление знаниями и ИТ-компания

1.1. Взаимосвязь софтверных компаний и менеджмента знаний

Об эффективности и положительном влиянии

управления знаниями в организации в общем независимо от отрасли широко известно. Вот что, например, сказано в предисловии относительно недавнего опубликованного ГОСТ Р 53864-2010 «Менеджмент знаний. Термины и определения»:

Опыт показывает, что успешное использование системы менеджмента знаний в бизнесе способствует решению самых разнообразных проблем: человеческих и культурных аспектов, персональных мотиваций, выбора современной методологии управления и современных технологий.

Софтверным организациям реализовать комплекс мер по управлению знаниями гораздо проще, потому что у них апостериори присутствуют одна из двух составляющих данного понятия. Традиционно управление знаниями рассматривается как направление менеджмента и как направление в информатике [Гаврилова, 2008]. Для софтверных компаний вторая составляющая уже отработана, а значит организовать поддержку процессов создания, распространения, обработки и использования знаний для них задача более реальная и выполнимая.

Функция управления знаниями неразрывно связана и с процессами кодификации. Для софтверных компаний опять же данный процесс является более знакомым, а значит и проще внедряемым. Проекты управления знаниями обычно связаны с сетевыми технологиями, технологиями

совместной работы и средствами информационного поиска, с созданием каталогов знаний, систем коллективного обмена знаниями, технической инфраструктуры. Многие из этого применяется софтверными компаниями в своей повседневной рабочей деятельности. Разработка программного обеспечения – результат именно совместной работы, сетевые технологии на текущем этапе развития софтверного бизнеса неразрывно связаны с большинством разрабатываемых программных продуктов, то же можно сказать и про информационный поиск – многие компании сталкиваются с этим набором в своей работе.

Однако теория в данном случае может сильно расходиться с практикой. Чтобы проверить данное предположение был проведен онлайн-опрос и изучены публикации по данной теме.

1.2. Общая информация по онлайн-опросу

Опрос проводился среди сотрудников софтверных организаций, цель которого была - выявить общее состояние дел в компании в части управления знаниями и найти ответы на вопросы, связанные с корпоративной политикой данной области и личного отношения к ней самих сотрудников. Опрос был разбит на две части. В исследовании приняло участие 74 человека.

Первая часть, более общая, состояла только из закрытых вопросов, и ее цель – определить уровень развития компании в области управления знаниями. Было предложено 12 вопросов с вариантами ответов (да; нет; возможно). Ответы «да» были равнозначны 2 баллам, «возможно» - 1 баллу и «нет» - 0 баллов.

Вопросы для первой части были составлены по книге «Управление знаниями» [Джанетто, 2005]. Данная книга содержит в себе пошаговую подробную стратегию по внедрению системы управления знаниями и вопросники, ответы на которые помогут во внедрении. Данные вопросники были сжаты и переработаны в 12 вопросов, которые и задавались респондентам в нашем онлайн-исследовании (в скобках указан средний балл по данному вопросу среди всех участвовавших в опросе):

1. Вся информация, нужная для решения производственных и бизнес-задач документируется, структурируется и в дальнейшем доступна сотрудникам организации (0,78);
2. В компании есть сотрудники, в должностные обязанности которых входит сбор, классификация, распространение информации (0,43);
3. В компании известно, к кому обращаться за конкретной информацией (1,2);
4. Существуют ли в компании устройства, хранящие наилучшую возможную информацию по широкому кругу ключевых вопросов (0,8);

5. Разработаны ли способы документирования и обмена информацией между сотрудниками (0,78);
6. Существует ли сотрудники в обязанности, которого входит координация процесса обмена знаниями (0,36);
7. Осуществляется поиск путей преодоления барьеров, существующих при обмене знаниями (0,7);
8. Все работники вовлечены в процесс обмена и обновления знаний (0,7);
9. Информационные системы соединяют нас с источниками информации, необходимыми для работы (1,08);
10. Высшее руководство компании заинтересовано в эффективном управлении знаниями (1,08);
11. Мы объединяемся с другими организациями для разработки новых продуктов (0,68);
12. Персонал направляется в различные организации для приобретения новых знаний и опыта (0,68);

Максимальное количество баллов, которая могла набрать компания по результатам ответов – 24, минимальное – 0.

Вторая часть опросника носит менее формальный характер. Здесь вопросы преимущественно открытые. Цель второй части – выяснить более детально, какие технологии используются для управления знаниями в компаниях софтверного профиля, и получить личные мнения респондентов по смежному кругу вопросов. Попробуем подвести итог ответам респондентов.

1.3. Результаты онлайн-опроса

Как уже было сказано, в опросе приняло участие 74 человека. Большинство из них относят себя к техническим специалистам, около четверти опрошиваемых указало себя как менеджерский персонал. Принимали участие и представители высшего руководства.

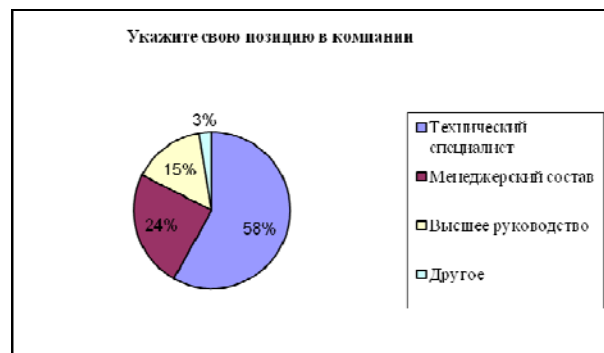


Рисунок 1 – занимаемые позиции респондентов.

По первой части стоит остановиться на

вопросах, по которым средний балл получился наименьшим и наивысшим. Очевидно, что данные пункты представляют наибольший интерес.

Меньше всего баллов участники опроса получили по вопросам 2 и 6. Данные вопросы пересекаются и сводятся в итоге к наличию в компании должности, например, директора по управлению знаниями (СКО), отвечающего за разработку процессов, способствующих накоплению, генерации и передаче знаний, применению их сотрудниками компании. Данная должность может звучать и проще: брокер знаний или инженер по знаниям. Так же все обязанности могут вписываться в должностные инструкции какой-либо другой роли в компании. Тем менее отсутствие такого человека показывает, что в данных компаниях вопросом управления знаниями серьезно никто не занимается. Мало баллов получили также вопросы 11 и 12. Их также можно рассматривать совместно. Одной из возможностей повышения профессионального уровня сотрудников является обучение и заимствование чужого опыта. Эти два вопроса относятся именно к сфере корпоративного обучения и обмена опыта посредством сотрудничества с другими организациями. Судя по небольшому количеству набранных баллов, данной проблеме также уделяется мало внимания.

Однако стоит заметить, что есть и положительные результаты опроса. Более чем половина ответивших участников утверждает, что «высшее руководство компании заинтересовано в эффективном управлении знаниями». Хорошо так же обстоят дела с информационными системами, они активно используются в повседневной деятельности. Наибольшее количество баллов получил вопрос «в компании известно, к кому обращаться за конкретной информацией». Это тоже важный момент.

Среднее количество баллов по всем опрошенным получилось 9,28, т.е. меньше половины. Это говорит о том, что управление знаниями в программных компаниях находится на начальном уровне, что серьезных работ и проектов в этой области не велось. Правда надо учитывать, что выборка онлайн-опроса нерепрезентативна и откладывает свой отпечаток на результаты исследования. Стоит отметить, что именно из-за нерепрезентативности и простоты исследования результаты никак не обрабатывались и представлены в «сыром» виде.

Результаты второй части опросника в части отношения работников к принципам управления знаниями носят оптимистический характер. Так на вопрос «Какими знаниями вы готовы делиться» только 5% опрошиваемых ответило, что «никакими». 86% опрошиваемых готовы уделять время для пополнения базы знаний. А 65% респондентов согласны, что такая известная форма представления знаний, как ментальные модели, удобна для структурирования знаний.



Рисунок 1 – ответ респондентов на вопрос «Какими знаниями вы готовы делиться».

Во второй части также задавался вопрос ведутся ли в компаниях базы по таким актуальным для области ИТ аспектам как: ценные наработки, популярные баги (ошибки при разработке программного обеспечения) и реализованные проекты. Оказалось, что даже по реализованным проектам в более чем 50% компаниях не ведется никаких баз. Процент ценных наработок и популярных ошибок еще меньше.

Во второй части была свободная графа, в которой участники опроса могли в произвольной форме выразить свои пожелания относительно управления знаниями и трудностями, с которыми они сталкиваются. В большинстве случаев люди упоминали, что не хватает четкой структуризации информации, что нужен инструмент, который бы облегчил управление знаниями, что не хватает общего единого хранилища всей информации, необходимой для работы. Были так же выдвинуты предложения по увязке знаний с потоком работ (workflow) и необходимость кодификации знаний. Кодификацию хочется отметить отдельно, потому что без нее невозможна последующая автоматическая обработка, например, сортировка, категоризация, фильтрация. Это также отмечалось участниками опроса.

Вторая часть опроса позволила более точно определить отношение самих работников к управлению знаниями в компании и их субъективное мнение на этот счет. Приведенные результаты дали наглядное представление и понимание, чего действительно не хватает в управлении знаниями в организации программного профиля, и что действительно нужно и выполнялось бы самими работниками. Например, результаты опроса выявили либеральное отношение сотрудников к возможной политике по созданию базы знаний, показали важность использования ментальных моделей.

2. Публикации по управлению знаниями в программных компаниях

Вопросу управлению знаниями в общем посвящено достаточное количество публикаций, предметно же к области деятельности ИТ-компаний в России печаталась всего одна защищенная диссертация «Стратегия и методы управления знаниями в ИТ-компаниях». Написана она была под

руководством к.т.н. Тузовского А.Ф. генеральным директором фирмы «ЭлиСи», занимающейся АСУ ТП в нефтегазовой отрасли. Диссертацию отличает ярко выраженный практический уклон, все выкладки были апробированы и внедрены на предприятии. Особое внимание уделялось взаимосвязи явных и неявных знаний с основными бизнес-процессами компании. *Для того, чтобы корпоративные знания активно воздействовали на эффективность и инновационность производства, они должны быть вовлечены в реальную бизнес-среду компании, то есть в процессы управления, процессы производства, процессы маркетинга и реализации продуктов и услуг* [Чириков, 2006]. В качестве основы корпоративной системы управления знаниями предлагается внедрять депозитарии знаний с использованием метаданных, метаописаний и онтологий. Так, по мнению, автора можно найти компромисс между необходимой кодификацией знаний и дороговизной этого процесса. Т.е. в качестве системы управления знаниями компании будет выступать семантическая надстройка над информационной системой предприятия. Автор по своему положению в организационной структуре является топ-менеджером (генеральным директором) и в его исследовании много уделяется внимания гуманитарной составляющей процессу управления знаниями: стимуляции работы экспертов, инициации проекта со стороны высшего руководства и т.д.

Другим примером является внутренний проект лидера консалтинга ИТ-услуг компании IBS. Основной целью создания базы знаний указывается сохранение инвестиций компании в интеллектуальный капитал. База знаний рассчитана на всех сотрудников, начиная от руководителя и заканчивая стажерами, у каждого свои цели использования. Например, для продавца это получение сведений о продуктах, проектах и экспертах компании. В IBS база знаний рассматривается в контексте работы сотрудников, проектов компании и предлагаемых продуктов. Так же введена система мотивации сотрудников по работе с базой знаний и подсчитан экономический эффект от использования.

ЗАКЛЮЧЕНИЕ

Управление знаниями для компаний, занятых в области информационных технологий, является важной и необходимой составляющей общей концепции управления организацией. Данная работа подтвердила актуальность рассматриваемого направления, как с точки зрения самой компании, так и с точки зрения понимания сотрудников о важности ведения подобной политики.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Гаврилова, 2008] Гаврилова, Т. А. Интеллектуальные технологии в менеджменте/ Т. А. Гаврилова, Д.И.; - СПб.: Изд-ва

«Высшая школа менеджмента»; Издат. Дом С.-Петерб. гос. ун-та, 2008.

[Джанетто, 2005] Джанетто К., Уиллер Э. Управление знаниями. Руководство по разработке и внедрению корпоративной стратегии управления знаниями / Пер. с англ. Е.М. Пестеровой – М.: Добрая книга, 2005.

[Чириков, 2006] Чириков С.В. Стратегия и модели управления знаниями в ИТ-компаниях / диссертация на соискание ученой степени к.т.н, 2006 – 166 с.

ГОСТ Р 53894-2010 Менеджмент знаний. Термины и определения. – Москва: Стандартинформ, 2011 – 11 с.

ABOUT PECULIARITIES OF KNOWLEDGE MANAGEMENT IN RUSSIAN SOFTWARE COMPANIES

Gavrilova T.A. *, Melnikov N.E. **

* Graduate School of Management St.Petersburg University, Russia, St.Petersburg

tgavrilova@gsom.pu.ru

** Saint Petersburg Electrotechnical University "LETI", Russia, St.Petersburg

melnikov.n.e@gmail.com

In work main peculiarities of software companies concerning knowledge management are considered. Source data were obtained through online survey on popular thematic Internet resource. Also analysis of Russian papers connected with this issue was performed.

INTRODUCTION

Information Technology can undoubtedly be named as one of the most dynamic and fast-growing fields. The rapid development of information technology can be seen in the constant emergence of new solutions, creation of new and innovative IT applications long before any normative standards exist for them, and strong market growth. Companies that working on this market have a deal with changing external environment and undergo constant development. Knowledge management as one of approaches to company management satisfies the requirements of this kind of fields where operation conditions are extremely changeable and actuality of information and knowledge is especially important. In this article questions and issues of knowledge management in Russian software companies are touched upon.

MAIN PART

Report theses in volume of 3-5 paragraphs.

CONCLUSION

Knowledge management for IT-companies is an important and necessary part of overall concept of organization management. This work confirmed actuality of selected direction both in terms of company and in terms of understanding of employees about significance of conducting such policy.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

Модели и методы интеграции информационных и вычислительных ресурсов

Глоба Л.С., Новогрудская Р.Л.*

Национальный Технический Университет Украины «Киевский Политехнический Институт», г. Киев, Украина

rinan@ukr.net

lgloba@its.kpi.ua

В работе показан подход к построению функционального интернет-портала знаний, решены проблемы качественного представления информации на портале, а также рассмотрены вопросы интеграции информационных и вычислительных ресурсов портала.

Ключевые слова – портал, представление знаний, расчетные задачи, онтология, модель, бизнес процессы.

ВВЕДЕНИЕ

Для повышения эффективности доступа к инженерным знаниям за счет создания специализированного Интернет-портала инженерных знаний необходимо решить следующие задачи [Глоба Л.С., Новогрудская Р.Л., 2010], [Глоба Л.С., Новогрудская Р.Л., 2011]:

- Качественное представление знаний на портале
- Систематизация и структуризация информации
- Формализация инженерных знаний
- Эффективный содержательный поиск
- Описание сервисов, используемых для решения расчетных задач на портале.

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

- Построение качественной модели представления знаний на портале;
- Описание связности элементов функционирования портала;
- Организация правильной последовательности задействования (активации) элементов функционирования портала.

Также необходимо учесть, что важной особенностью портала является интеграция информационных и вычислительных ресурсов в структуру портала, для чего необходимо корректно определить и задать взаимосвязь и взаимодействие знаний и данных, их хранилищ и сервисов, реализующих поиск, расчет и различные вычисления, доступные пользователю на портале.

1. Модели и методы портала

Структуризация и систематизация информации доступной на портале осуществляется с помощью построения **онтологической модели представления знаний, формализация элементов функционирования портала**(вычислительных и информационных ресурсов)– с помощью **теории множеств и алгебры логик**, а реализация *правильной последовательности действий и активации элементов портала*– с помощью **моделирования бизнес процессов (рабочих процессов и потоков данных)**

Структуризация и систематизация информации (онтология)

Важным этапом построения портала является структуризация и систематизация информации и знаний в области прочности материалов, что позволит конечному пользователю осуществлять просмотр и поиск конкретных сведений в рассматриваемой сфере. Структуризация и размещение на портале организуется удобным для конечного пользователя образом, реализуя проблемно-ориентированные средства навигации и поиска по информационному пространству портала. При этом поиск информации организован так, что пользователь имеет возможность задавать запрос не только и не столько по ключевым словам, сколько в знакомых ему терминах предметной области портала [Андреева О.А. и др., 2006].

Бизнес процессы

В результате использования моделей бизнес процессов стало возможным определить последовательность взаимодействия и кооперации данных и сервисов, информационных и вычислительных ресурсов, баз данных и баз знаний портала, задействованных при его

функционирования [Август-Вильгельм Шер, 2000].

Важное место в анализе процессов, происходящих на портале во время его функционирования, занимают выявление, описание и представление взаимосвязанности сервисов, которые реализуют различные вычислительные задачи, такие как: моделирование, построение, подбор материала по определенному критерию, выбор материала по заданным пользователем характеристикам. Также, необходимо определить механизм связывания информационных и вычислительных ресурсов. Одним из самых важных требований является идентификация последовательности активации операций, инициализируемых различными запросами пользователей 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.

Построение моделей бизнес-процессов интернет портала в области прочности материалов позволяет понять и правильно определить взаимосвязь и кооперацию данных и сервисов, которые используются в поисковых запросах и вычислительных задачах, реализуемых на портале. С точки зрения бизнес моделирования функциональность портала полностью описывается с помощью диаграммы Вариантов Использования нотации UML, а связность и последовательность инициализации различных элементов портала определяется с помощью диаграмм кооперации и диаграммы последовательности.

С помощью анализа и моделирования бизнес процессов портала осуществляется реализация правильной последовательности взаимодействия сервисов и хранилищ данных.

Формальное описание

После построения онтологической модели, а также моделей бизнес процессов портала было проведено формальное описание элементов моделей, для определения формальных структур, которые в дальнейшем будут использоваться при проектировании портала и построении программного обеспечения для реализации портала. При формальном описании использовалась теория множеств и алгебра логик [Александров П. С., 1977], [Загорюлько Ю.А. и др., 2007].

Приведем пример формального описания части онтологии портала, которая задает структуры для описания инженерные знания в рассматриваемой предметной области. Формально данная онтология описывается как $O_3 = \{C_{O_3}, A_{O_3}, R_{O_3}, T_{O_3}, F_{O_3}, D_{O_3}\}$ [Загорюлько Ю.А. и др., 2007], где

C – множество классов, описывающих понятия некоторой предметной или проблемной области $C_{O_3} = \{C_1, C_2, C_3, C_4, C_5, \dots\}$;

A – множество атрибутов, описывающих свойства понятий и отношений

$A_{O_3} = (A_{C_1}, A_{C_2}, A_{C_3}, A_{C_4}, A_{C_5})$;

R – множество отношений, заданных на классах, $R = \{R_{AS}, R_{LA}, R_n, R_{CD}\}$ – существуют отношения следующих типов: R_{AS} – ассоциативное отношение, R_{LA} – отношение «часть-целое», R_n – отношение наследования, R_{CD} – отношение «класс-данные» $R_{O_2} = (R_{AS_1}(O_3), \dots, R_{AS_n}(O_3), R_{LA_1}(O_3), R_{n_1}(O_3), \dots), R_{CD_1}(O_3), \dots, R_{CD_s}(O_3))$;

T – множество стандартных типов значений атрибутов (string, integer, real, date);

F – множество ограничений на значения атрибутов понятий и отношений (используется не для всех атрибутов, а лишь для тех значения которых должны лежать в некоторой области, не могут быть меньше/больше заданной величины, либо определяются, каким либо другим правилом. Так, например, на значение атрибута дата начала класса мероприятия онтологии инженерной деятельности накладывается ограничение, $T(\text{дата}) = \text{date}, F(T) > 0$.);

D – множество экземпляров классов

Таким образом, онтология портала представляет собой иерархию понятий (или классов), связанных отношениями. Различные свойства каждого понятия описываются на основе атрибутов понятий и ограничений, наложенных на область их значений. Рассматриваемая онтология включает следующие классы понятий:

Исследовательское оборудование - $C_5(O_3)$,

Метод исследования - $C_1(O_3)$,

Объект исследования - $C_2(O_3)$,

Результат исследования - $C_3(O_3)$,

Назначение исследования - $C_4(O_3)$.

На классах данной онтологии реализованы следующие отношения:

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

отношение «применяется к» - связывает метод и тип объектов исследования, к которым он применяется -

$R_{AS_3} = \{C_1(O_3) \times C_2(O_3)\}$, аналогично формально описаны еще 4 отношения:

$R_{AS_4} = \{C_1(O_3) \times C_5(O_3)\}$, $R_{AS_5} = \{C_4(O_3) \times C_1(O_3)\}$,

$R_{AS_6} = \{C_3(O_3) \times C_5(O_3)\}$, $R_{AS_7} = \{C_3(O_3) \times C_2(O_3)\}$.

Структурные отношения («общее-частное»)

$R_{LA}(O_3) = C_k(O_3) \subset C_m(O_3)$: - описывается связь между классом Исследовательское оборудование и установки для проведения опытов разного типа, измерительное оборудование, криогенное оборудование, тепловые установки.

$C_5 \subset C_{5_1} \wedge C_{5_2} \wedge C_{5_3} \wedge C_{5_4}$.

Отношение наследования

$R_n(O_3) = a_i, r_i | A_{C_n}(O_3) \rightarrow a_i, r_i | A_{C_i}(O_3)$ - используется для передачи атрибутов и отношений от родительского класса к дочернему: Класс Исследовательское оборудование его подклассы установки для проведения опытов разного типа, измерительное оборудование, криогенное оборудование, тепловые установки.

$A(C_5), R(C_5) \rightarrow A(C_5), R(C_5)$ $A(C_5), R(C_5) \rightarrow A(C_5), R(C_5)$.

$A(C_5), R(C_5) \rightarrow A(C_5), R(C_5)$. $A(C_5), R(C_5) \rightarrow A(C_5), R(C_5)$.

Отношения «класс-данные»

$R_{CD}(O_3) = C_j(O_3) \subseteq D_i(O_3)$: - используется для связки

конкретных экземпляров понятий с классом, оно актуально для всех классов данной онтологии: $C_1(O_3) \subseteq D$, $A_{C_1} \subseteq A_D$, $C_2(O_3) \subseteq D$, $A_{C_2} \subseteq A_D$, $C_3(O_3) \subseteq D$, $A_{C_3} \subseteq A_D$, $C_4(O_2) \subseteq D$, $A_{C_4} \subseteq A_D$, $C_5(O_3) \subseteq D$, $A_{C_5} \subseteq A_D$.

Таким образом, после построения онтологии портала, все ее классы получили формальное описание, что также оптимизирует организацию содержательного поиска по онтологии портала.

2. Онтология расчетных задач

Как уже было сказано, важное место в построении портала занимает интеграция расчетных задач, которых существует огромное количество и которые занимают важное место при проведении исследований в рассматриваемой предметной области. Существует необходимость корректно внедрить реализацию этих расчетных задач в информационную среду портала.

В процессе разработки онтологической модели портала были выделены 4 базовые онтологии: онтология инженерной деятельности, онтология инженерного знания, онтология инженерных расчетов и онтология предметной области. Онтология инженерных расчетов (ОИР) была выделена для описания расчетных возможностей, реализованные на портале. После анализа в ОИР были выделены следующие классы: 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.

ОИР играет важную роль при построении общей онтологии портала. С ее помощью становится возможным описать не только доступные пользователю на портале расчетные задачи, такие как сервисы вычислений, но и установить связи между конкретными сервисами (набором сервисов) для реализации конкретных бизнес-процессов, а также связи сервисов с информационными ресурсами, что необходимо для решения производственных задач пользователя. Однако вопрос об оптимизации этих связей остается нерешенным. В зависимости от параметров заданных пользователем и тематике расчетной задачи она может идти по тому, или иному алгоритму. Также при построении реальных инженерных задач, используется их декомпозиция на подзадачи, которые в зависимости от направленности общей задачи и параметров, которые используются при расчете, могут компоноваться в общую задачу динамически, и одна подзадача, может использоваться в нескольких общих задачах. Исходя из этого, необходимо учесть и описать логику связности расчетных задач. Таким образом, возникает необходимость в метамодели расчетных задач. На рисунке 1 представлена модель интеграции расчетных задач в среду портала. Становится понятным, что онтология расчетных задач служит для определения структур для расчетных задач, а для связи непосредственно

расчетных задач с физическим уровнем (информационными ресурсами, хранилищами данных и знаний, сервисами) нужна модель, которая будет играть роль некоего интегрирующего звена.

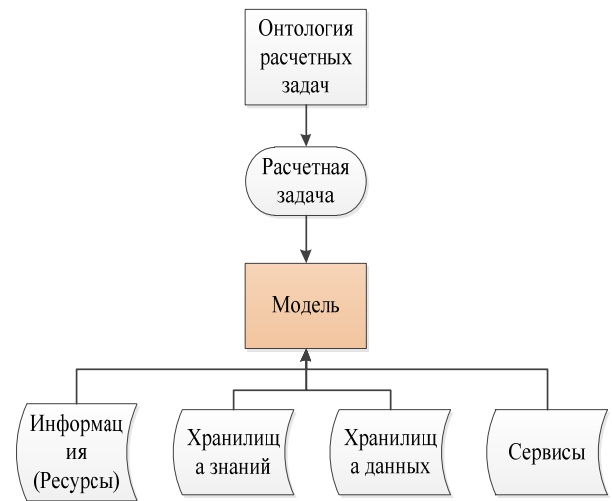


Рисунок 1 – Методология интеграции расчетных задач в пространство портала

МОДЕЛЬ РАСЧЕТНЫХ ЗАДАЧ

Поскольку главным отличием предложенного портала от других является наличие в его функционале возможности реализации различных расчетных задач, то важным этапом проектирования портала является выбор моделей и методов анализа связности и последовательности активации элементов функционирования портала (сервисы, данные, информационные ресурсы, хранилища данных). Интеграция вычислительных ресурсов в информационную среду портала предполагает решение следующих задач:

- описание элементов функционирования портала;
- описание последовательности действий при активации элементов;
- описание связности элементов.

Указанные выше требования предложено решить следующим образом:

- создание метаописаний для каждого элемента функционирования портала;
- алгоритмизация процессов происходящих на портале в процессе его функционирования;
- описание связей между элементами с использованием теории множеств.

Расчетные задачи: $Расч = \{Расч_1, Расч_2, \dots, Расч_n\}$. $Расч_j = \sum_{i=1, k}^{j-1} Расч_i$ - расчет состоит из суммы расчетов. $Расч_j = \{ \sum M_j, \sum P_j, \sum S_j \}$ - модель каждого расчета включает (каждый расчет характеризуется) множество компонент - набор некоторых мета-описаний, набор некоторых алгоритмов и набор некоторых логических связей.

Первым элементом метамодели являются *метаописания*, они используются для описания расчетов, их идентификации. Учитывая мета-описания можно строить уровни иерархии для расчетов, а также определять какой подрасчет к

какому расчету относится. Например: для расчета на устойчивость будут актуальными следующие мета-описания: Параметры, Нагрузки, Предельные состояния, Принципы оценок, и т.д.[Временные нормы расчета на прочность]

Таким образом, за счет этого элемента метамодели расчетных задач устанавливается жесткая иерархия и принадлежность на уровне мета-описаний (Определенное мета-описание отвечает за определенный параметр).

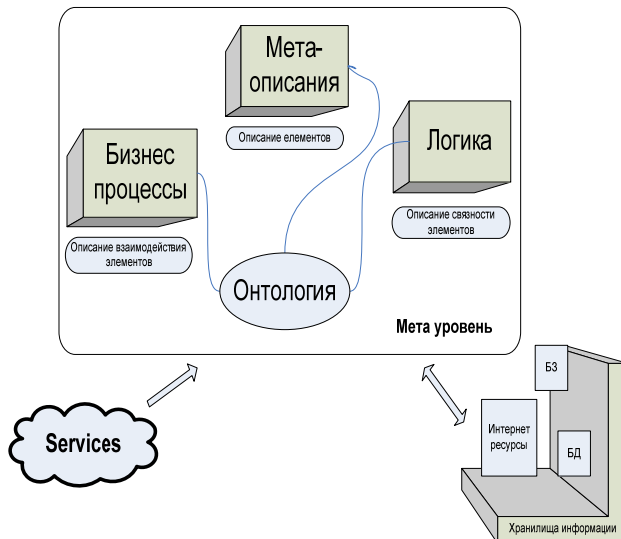


Рисунок 2 – Метамодель расчетных задач

Алгоритмизация определяет последовательность, заданную по иерархии подрасчетов в общем расчете и по очереди выполнения работ в расчете. Например: Расчет «Коэффициент снижения прочности сварными швами» состоит из следующих подрасчетов:

- «Определение коэффициента снижения прочности»,
- «Определение расстояния от края любого отверстия до оси сварного шва», и является подрасчетом расчета «Выбор основных параметров» который в свою очередь включает:
- «Определение размеров корпусов катушек реакторов с арочным распором»,
- «Коэффициент снижения прочности сварными швами»,
- «Фланцы, нажимные кольца и крепежные детали»; этот расчет также является подрасчетом общего расчета «Расчет на прочность силовых элементов магнитных систем ИТЭР». [А. С. Зубченко, 2001]

В алгоритме устанавливается жесткая последовательность действий (Расчет А - Р_1: 1,2; 1:1.1, 1.2, 1.3, 2: 2.1, 2.2, 2.3, 2.4). Таким образом, алгоритмизация это организация реализации правильной последовательности взаимодействия в расчете.

Третьим элементом модели является логика, она позволяет, описывая формально расчеты, их мета-

описания с помощью теории множеств, алгебры логик, задавать правила для связности. Логика – описание связности (динамичность изменения связи элементов один с другим и какой расчет с каким связывать): подтягивается «на ходу» (не жесткая привязка расчет- расчет), не «вертикальная» последовательность (выбор расчетов для задействования в данном из пространства, а не по вертикали).

Приведем примеры разных типов связности по логике:

а) В зависимости от «принципов оценок» - мета-описания (по различным предельным состояниям) подрасчеты компоуются в общий расчет. Можно определить какие именно расчеты подтягивать.

Например: если в мета-описании есть какая-то из принципов оценок (кратковременное разрушение, пластическая деформация и т.д.), то это «Расчет на прочность силовых элементов магнитных систем ИТЭР.»

б) В зависимости от того, какие параметры учитывать, подгружается тот или иной расчет (для общего расчета подтягиваются подрасчеты, для подрасчетов определяется общий расчет, этапами которого они являются)

Например: если в мета-описании есть какой-то из параметров (нагрузки разных видов: пондеромоторные, силовые), то это расчет по выбору основных параметров – «Расчет на прочность силовых элементов магнитных систем ИТЭР»

с) В зависимости от методов положенных в основу расчета употребляются те, формулы, которые необходимы. (методы можно определить по соответствию предельным состояниям)

Например: есть предельное состояние *вязкое разрушение* – значит это метод предельных нагрузок – он используется при расчете по выбору основных параметров.

Каждому расчету ставится в соответствие мета-описания (не только стандартные, но и специфичные, для предметной области), в результате чего эти мета-описания описывают формально каждый расчет и позволяют определить какой сервис использовать при наличии определенного мета-описания.

СЕРВИСЫ

Для реализации определенных расчетных задач использовались различные сервисы [AnArchitectureforDifferentiatedServices]. Далее приведено модель сервисов и их краткое описание.

При построении портала использовались следующие виды сервисов:

- Внешние (общие) – относящиеся к общей работе (общим функциям) портала
- Внутренние (частные) / ориентирование на специфичные задачи – сервисы, использующиеся в

специфических (присущих именно этому portalу) заданиях

Общие сервисы

- Сервис поиска (для реализации функции стандартного поиска, для поиска данных, для поиска расчетных задач, для поиска сервисов реализующих расчетные задачи)
- Сервис просмотра (для просмотра информации)
- Сервисы отображения (информация представлена в различных видах – диаграммы, текстовые файлы, таблицы, и . д.)
- Сервис обработки входных данных (для подхвата введенной пользователем информации)
- Сервис передачи информации (для передачи информации после ее обработки – например, после выполнения определенных расчетных задач, либо для передачи информации из хранилищ после ее нахождения сервисом поиска, и т.д.)
- Сервис извлечения (для извлечения информации из хранилищ)
- Сервис авторизации/аутентификации
- Сервис подключения к интернет ресурсам (portal предоставляет возможность пользователю работать с информацией, размещенной на других сайтах, библиотеках, и т.д.)



Рисунок 3 – Модель сервисов portalа

Частные сервисы

- Сервисы расчетных задач (специфические не стандартные сервисы, которые используются при реализации расчетных задач portalа)
 - сервис 1 (для расчетной задачи 1)
 - сервис 2 (для расчетной задачи 2)
 - сервис 3 (для расчетной задачи 3)
- Вычислительные сервисы (сервисы, которые используются для расчетов - калькуляции)
 - Сервисы подбора (реализующие следующие функции portalа: подбор материала по заданному пользователем свойствам/характеристикам, подбор свойств/характеристик материалов заданных пользователем, подбор свойств/материалов наиболее удовлетворяющих заданию/требованиям пользователя)
 - Сервисы построения/моделирования (сервисы, использующиеся для построения ответа

на задание пользователя: графики, зависимости, диаграммы, и т.д.)

- Сервисы связности информации (нужны для оптимизации процессов на portalе, определяют связность информации по определенным критериям).

ЗАКЛЮЧЕНИЕ

В результате проведенных исследований была предложена методика и средства, которые предлагается использовать при построении portalов инженерных знаний: интеграция онтологической модели, моделей бизнес процессов и описаний элементов на основании формальных теорий. Также была предложена модель, интегрирующая расчетные задачи в информационную модель portalа.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Глоба Л.С., Новогрудская Р.Л., 2010] Глоба Л.С., Новогрудская Р.Л. Подход к построению интернет-portalа инженерных знаний. // Труды X международной научной конференции имени Т.А. Таран «Интеллектуальный анализ информации». Киев, 2010, С. 53-62
- [Глоба Л.С., Новогрудская Р.Л., 2011] Глоба Л.С., Новогрудская Р.Л. Систематизация информационных ресурсов Интернет-portalа «Прочность материалов», Вісник Харк. нац. ун-ту імені В. Н. Каразіна, серія "Математичне моделювання. Інформаційні технології. Автоматизовані системи управління". - Вип. 16, № 927, 2011.
- [Андреева О.А. и др., 2006] Андреева О.А., Боровикова О.И., Загоруйко Ю.А. и др. Археологический портал знаний: содержательный доступ к знаниям и информационным ресурсам по археологии // Труды 10-й нац. конф. по искусственному интеллекту с международным участием КИИ'2006. М.: Физматлит, 2006. -Т.3, - С.832–840.
- [Август-Вильгельм Шеер, 2000] Август-Вильгельм Шеер. Бизнес-процессы. Основные понятия. Теория. Методы. Пер с англ. / Август-Вильгельм Шеер. - М.: Весть - МетаТехнология, 2000. – 182 с.
- [Александров П. С., 1977] Александров П. С. Введение в теорию множеств и общую топологию. - М., 1977. – 368с.
- [Загоруйко Ю.А. и др., 2007] Загоруйко Ю.А., Боровикова О.И. Технология построения онтологий для portalов знаний по гуманитарным наукам // Труды Всероссийской конференции с международным участием “Знания-Онтологии-Теории” (ЗОНТ-07). Новосибирск, 2007. Т.1, С.191-200.
- [Боровикова О.И. и др., 2002] Боровикова О.И., Загоруйко Ю.А. Организация portalов знаний на основе онтологий. // Труды международного семинара Диалог'2002 “Компьютерная лингвистика

и интеллектуальные технологии”. Протвино, 2002. Т.2, С.76-82.

[Временные нормы расчета на прочность] Временные нормы расчета на прочность силовых элементов магнитных систем ИТЭР. НИИ ЭФА.- М Л: Ин-т маш. АН СССР.- 1990.-73с.

[А. С. Зубченко, 2001] А. С. Зубченко Марочник сталей и сплавов, Издательство "Машиностроение", 2001. – 663с.

[An Architecture for Differentiated Services] An Architecture for Differentiated Services[Электронный ресурс]– Режимдоступу: <http://tools.ietf.org/html/rfc2475>

MODELS AND METHODS OF INFORMATIONAL AND COMPUTATIONAL RESOURCES INTEGRATION

Globa L.S., Novogradskaya R.L.

National Technical University of Ukraine

rinan@ukr.net

lgloba@mail.ru

The report presents the approach to solving the informational and computational resources integration into Internet portal of different direction. The possibility of designing the Internet portal in the field of strength of materials for providing meaningful accesses to knowledge and data is described.

INTRODUCTION

The aim of the article is to improve access to the engineering knowledge by the specialized knowledge internet-portal in the strength of materials development. Some tasks to be solved are showed and approaches to their solving are described.

MAIN PART

Structuring and systematization of information is organized with the help of ontology model designing [1], analyses of coherence and cooperation of informational and computational resources is made using set theory and algebra of logic theory and realization of proper sequence of services and data cooperation is realized by portal business processes analyses and modeling.

Structuring and placement on the portal is organized in friendly for the end user way by implementing problem-oriented navigation tools. Information search is organized in such a way that user can specify the request not only by keywords, but in familiar to him terms of subject field.

While constructing the business process models for the Internet portal in the field of strength of materials it is possible to understand the collaboration and cooperation of data and services used in different search requests and calculation tasks realized on the portal. The need of business processes analyses is followed from next requirements:

- Important place is taken to the identification, description and presentation of services (used for different calculation tasks realization) coherence.

- It is necessary to determine binding mechanism of informational and computational resources.

- It is required to identify the sequence of operation initialized on different users request execution.

The formalizing is holding after description of all elements of ontology model using set theory. Here is an example of a formal description of the portal ontology, which sets the framework for the description of engineering knowledge in the subject area. Formally, this ontology is described as

$$O_3 = \{C_{o_3}, A_{o_3}, R_{o_3}, T_{o_3}, F_{o_3}, D_{o_3}\}.$$

It is clear that the ontology of computing tasks is used to determine the structures for computational tasks, but for connection directly computation tasks with the physical layer (information resources, data warehousing and knowledge services) a model, which will play the role of an integrator, is in need.

The process of computing resources in information environment of the portal integration includes the following tasks:

- A description of portal functioning elements;
- A description of the workflow appeared while activation of portal elements;
- A description of the elements coherence.

Calculation tasks model is proposed. It consists of 3 elements: meta-description, algorithm and logical connection. Meta-descriptions are used to describe the calculation tasks and their identification. By giving meta-descriptions the levels of hierarchy can be built for the calculation tasks and it become possible to determine which small calculation task to which general calculation task referees. Algorithmization determines the sequence defined by the hierarchy of small calculation tasks in general calculation task. Also algorithmization shows the order of work in the calculation. The logic allows, by describing calculation tasks and their meta-descriptions in the formal way with the help of set theory and logic algebra, to set the rules for computational tasks coherence.

Services that are used while portal construction are also described, there are two types of services: general and specialized. To general services search, browse, display, input data processing, transmission, retrieval, authorization, authentication, connection to internet resources services refers. Various computational tasks services, calculation services, services of selection, building / modeling services, services of information connection are specialized services.

CONCLUSION

In the given document the methodology and facilities for knowledge internet-portal construction were described. The methodology presents integration of ontology model, business processes models and description of elements that are based on formal theories. As well the meta-model of portal calculation tasks was proposed and services for calculation tasks realization were described.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОЦЕНКА КАЧЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Бахтизин В.В., Неборский С.Н.

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

bww@bsuir.by

sergei.neborski@gmail.com

Рассматривается проблема оценки качества интеллектуальных информационных систем в контексте современных международных стандартов в области качества, приводятся примеры мер для оценки качества.

Ключевые слова: качество интеллектуальных информационных систем, оценка качества интеллектуальных информационных систем.

ВВЕДЕНИЕ

Проблема оценки качества информационных систем (ИС) является актуальной и важной, т. к. именно результаты оценки качества дают понимание того, что пользователь может ожидать от рассматриваемой ИС. Оценка качества системы должна производиться на всех этапах жизненного цикла ИС, ведь только так разработчик может корректировать как техническую составляющую процесса разработки (принимать архитектурные решения), так и организационную (например, усилить команду разработчиков специалистами, способными решить конкретную проблему в конкретном модуле, выявленную в результате оценки).

1. Проблема оценки качества интеллектуальных информационных систем

Оценка качества программного обеспечения (ПО) не является новой. Первые модели начали появляться еще в 80-х годах XX века (модели Боэма, Дроми, FURPS), и сегодня существует ряд международных стандартов, посвященных именно качеству ПО (ISO/IEC 9126, ISO/IEC 15504, ISO/IEC 15939, ISO/IEC 25000). Однако данные стандарты не учитывают особенностей интеллектуальных информационных систем (ИИС), а сама проблема оценки качества ИИС не является проработанной и хорошо изученной. Данная работа стремится соотнести текущие наработки в области оценки качества ИИС с актуальными действующими стандартами качества ПО.

Оценивая качество ИИС, в настоящее время

оценивают преимущественно функциональность. Очевидно, оценка функциональности имеет свои особенности, и классические подходы к оценке (в частности, использование метрик оценки качества программных средств (ПС) по стандарту 9126-2 [ISO/IEC TR 9126-2, 2003]) не всегда применимы. Существующие подходы к оценке качества ИИС можно условно разделить на два типа [Лахутин, 1970]:

1) качество интеллектуальной системы посредством формальной процедуры ставится в зависимость от меры ошибок системы, определяемой тем или иным способом;

2) качество интеллектуальной системы оценивается экспертами.

В силу проблем, связанных с вовлечением эксперта в процесс оценивания (“эффект персоны”, когда психологический настрой человека, испытывающего систему, ориентирован прежде всего не на знакомство с ней, а на подтверждение собственных сложившихся взглядов и идей или на подтверждение своей компетенции, и “эффект персонализации”, когда в процессе знакомства эксперта с системой происходит отождествление им потенциальных возможностей системы с возможностями человека [Григорьев, 2004]), процедура оценки должна быть максимально объективна и не зависеть от участвующих в ней лиц.

В литературе можно встретить отождествление качества ИИС и качества реализации ИИС, однако их необходимо разделить. Качество реализации ИИС зависит от характеристик качества используемых моделей знаний и обрабатываемых концептуальных информационных ресурсов

[Shalfeeva, 2010].

Существуют структурные подходы для оценивания качества различных классов информационных ресурсов интеллектуальных программных систем. Эти подходы позволяют обнаруживать дефекты и недостатки представления информации, несогласованности между определениями и фактами. Однако данные подходы в большей степени перекликается с качеством процесса разработки, но не непосредственно качеством ИИС.

2. Стандарты качества программного обеспечения

Актуальными стандартами в области качества ПО являются стандарты ISO/IEC 25000 SQuaRE (Software Quality Requirements and Evaluation – требования к качеству и оценка программного продукта). В Республике Беларусь введен в действие стандарт СТБ ISO/IEC 25000-2009 «Разработка программного обеспечения: Требования к качеству и оценка программного продукта (SQuaRE): Руководство по SQuaRE» [СТБ ISO/IEC 25000, 2009]. Отдельные группы стандартов SQuaRE приведены на рисунке 1.

Требования к качеству 2503n	Модель качества 2501n	Оценка качества 2504n
	Управление качеством 2500n	
	Измерение качества 2502n	

Рисунок 1 – Группы стандартов SQuaRE 25000

Данные стандарты регламентируют иерархическую модель качества ПС. На верхнем уровне находятся характеристики, каждая из которых разделяется на подхарактеристики. Подхарактеристики определяются мерами. Мера – это атрибут объекта и метод его измерения. Концептуально, такая модель качества справедлива и для ИИС.

Однако если стандарты ISO/IEC 25000 жестко определяют характеристики и подхарактеристики качества, то для оценки конкретных ИИС определенные стандартами характеристики не всегда подходят в полной мере, и необходимо определять актуальные характеристики и ранжировать их по степени важности.

Стандарт ISO/IEC 25000 SQuaRE [СТБ ISO/IEC 25000, 2009] выделяет две модели качества ПО:

- качество в использовании;
- качество продукта.

Модель качества в использовании (рисунок 2) определяет 5 характеристик качества, связанных с

взаимодействием пользователя с системой: продуктивность, эффективность, способность к удовлетворению, отсутствие риска, покрытие контекста.

Модель качества продукта декомпозирует свойства ПО по 8 характеристикам:

- функциональная пригодность,
- производительность,
- совместимость,
- удобство использования,
- надежность,
- защищенность,
- сопровождаемость,
- мобильность.



Рисунок 2 – Модель качества в использовании

Так как одним из характерных признаков ИИС является способность к самообучению, можно утверждать, что ИИС является постоянно развивающейся системой. Значит, для ИИС большой интерес имеет модель качества продукта (рисунок 3), нежели модель качества в использовании.



Рисунок 3 – Модель качества продукта

3. Метод оценки качества интеллектуальных информационных систем

Для оценки качества ИИС предлагается использовать метод оценки качества по ISO/IEC 25020 [ISO/IEC 25020, 2006], который по сути приходит на смену стандарту ISO/IEC 15939:2002 [ISO/IEC 15939, 2002].

Схематично модель, положенная в основу данного метода, показана на рисунке 4: меры определяют подхарактеристики качества, на основе подхарактеристик получается оценка отдельных характеристик, и уже на основе характеристик определяется интегральная оценка качества.

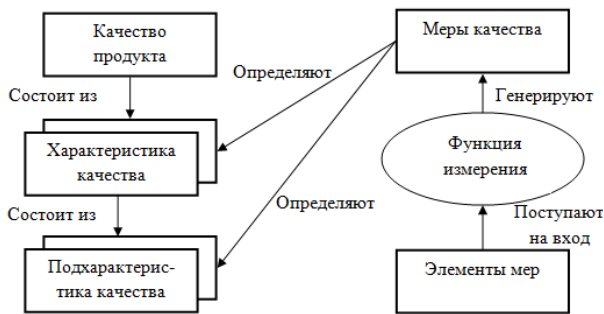


Рисунок 4 – Модель оценки качества

Интегральная оценка качества P^Q рассчитывается по формуле (1):

$$P^Q = \frac{\sum_{k=1}^K (P_k^X \cdot V_k^X)}{\sum_{k=1}^K V_k^X}, \quad (1)$$

где P_k^X - оценка k -й характеристики качества,

V_k^X - весовой коэффициент k -й характеристики качества, причем $\sum_{k=1}^K V_k^X = 1$.

Для определения степени важности отдельных подхарактеристик и характеристик используются весовые коэффициенты V_k , определяемые методами экспертных оценок (например, могут быть использованы методы коллективной работы экспертной группы или метод Дельфи).

4. Выбор мер качества

И все же, основной проблемой оценки качества ИИС можно считать правильный выбор мер качества, т. к. меры позволяют дать объективную количественную оценку конкретных свойств системы. Наиболее проработанной здесь является оценка качества поисковых машин. Можно привести методику оценки, суть которой состоит в получении интегральной оценки экспертом на основе значений вычисленных мер качества [Масевич, 1996]. Основная идея состоит в сборе нескольких параметров поисковой машины, описывающих поведение пользователей и свойства самой системы:

- насколько быстро выполняются запросы;
- сколько занимает поисковый индекс;
- насколько хорошо удовлетворяются информационные потребности пользователя;
- сколько информации удалось получить по запрашиваемой теме;

- насколько удобно пользоваться системой.

Соответственно, можно привести примеры следующих мер для оценки поведения пользователей при поиске в интернете [Rambler, 2011]:

- вероятность возврата в поисковую выдачу после перехода на найденную страницу;
- вероятность просмотра результата менее чем 20 секунд и возвращения в поисковую выдачу;
- средняя позиция ссылок на страницы, на которые осуществлялся переход пользователя;
- вероятность полного отсутствия переходов на страницы поисковой выдачи.

Приведенные меры согласуются с требованиями к мерам стандарта ISO/IEC 25020 [ISO/IEC 25020, 2006], они могут быть представлены в относительных единицах и нормированы с тем, чтобы значения изменялись на отрезке $[0, 1]$, причем чем ближе к 1, тем выше качество ИИС. Однако поисковые машины – лишь один класс ИИС, и для других классов характеристики и меры качества будут значительно отличаться.

ЗАКЛЮЧЕНИЕ

Обобщая вышесказанное можно отметить, что проблема оценки качества ИИС остается нерешенной в силу многогранности ИИС и узкой специфики их отдельных классов. В то же время, модели и методы оценки качества ПО, определяемые новыми международными стандартами ISO/IEC 25000, могут и должны быть адаптированы с целью их применения к ИИС.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [ISO/IEC TR 9126-2, 2003] ISO/IEC TR 9126-2:2003: Программная инженерия – Качество продукта – Часть 2: Внешние метрики. – Предст. 2003-07-01. – Женева: ISO/IEC-IEEE, 2003. – 86 с.
- [Лахутин, 1970] Лахутин Д.Г. О проблеме оценки поисковых систем. Ч. 1. – Д.Г. Лахутин, С.О. Чернявский // НТИ. Сер. 2. №1. 1970. С. 24-34.
- [Григорьев, 2004] Григорьев В.А. Оценка качества интеллектуальных систем, оперирующих неформальными понятиями / В.А. Григорьев // Наука и образование, – 2004 г. – №10
- [Shalfeeva, 2010] Shalfeeva E.A. Monitoring of conceptual informational resources for intelligent software systems / E.A. Shalfeeva // Proc. First Russia and Pacific Conference on Computer Technology and Applications, – 2010. – С.140-144
- [СТБ ISO/IEC 25000, 2009] СТБ ISO/IEC 25000-2009: Разработка программного обеспечения. Требования к качеству и оценка программного продукта (SQuaRE). Руководство по SQuaRE. – Введ. 2010-01-01. – Минск : Госстандарт Респ. Беларусь, 2009. – 44 с.
- [ISO/IEC 25020, 2006] ISO/IEC 25020: Software engineering – Software product quality requirements and evaluation (SQuaRE) – Measurement reference model and guide. – Предст. 2006-08-08. – Канада: ISO/IEC-IEEE, 2006. – 24 с.

[ISO/IEC 15939, 2007] ISO/IEC 15939: Software engineering – Software measurement process – Предст. 2007-07-23. – Женева: ISO/IEC-IEEE, 2007. – 38 с.

[Масевич, 1996] Масевич А.Ц. Качество информационного продукта автоматизированных библиотечных систем / А.Ц. Масевич, В.П. Захаров // Материалы 3-й Международной конференции “Библиотеки и ассоциации в меняющемся мире: новые технологии и новые формы сотрудничества. Крым – 96” (Форос, 1—9 июня 1996г.) / Ялта, 1996. Т. 1. С. 178—189.

[Rambler, 2011] Rambler pFound - метрика качества поиска [Электронный ресурс] / Рамблер Интернет Холдинг, 2011. – Режим доступа: http://rcdl2011.vsu.ru/doc/full_text/Report.5.6.pdf – Дата доступа: 17.12.2011.

INTELLIGENT SYSTEMS QUALITY EVALUATION

Bakhtizin V.V., Niaborski S.N.

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

bww@bsuir.by

sergei.niborski@gmail.com

This paper analyses the problem of intelligent systems quality evaluation considering current stage of development of international standards devoted to software quality. It gives examples of quality measure for intelligent systems and shows how to apply quality evaluation method defined in ISO/IEC 25020 to intelligent systems.

INTRODUCTION

The problem of quality evaluation of intelligent systems is essentially important because results of quality estimation allow to understand what a user can expect from the system. Quality evaluation must be conducted on all stages of system development, thereby allowing a developer to adjust technical and administrative approaches to the project. This paper aims to provide a link between international quality standards for software evaluation and intelligent systems and shows how to apply those standards to measure intelligent systems quality.

MAIN PART

When trying to evaluate intelligent system quality, in most cases staff deal with functionality. The functionality of intelligent system is observed from two perspectives:

1) intelligent system quality is defined as a function of error quantity the number of which is captured in a given way;

2) intelligent system quality is estimated by an expert.

System quality, however, is not only functionality – it includes much more characteristics each one of which is further decoupled into sub-characteristics. Software quality evaluation techniques can be applied to intelligent systems estimation as well.

In particular, international standard ISO/IEC 25000-2009 SQuaRE can be used to define product quality model by choosing 8 characteristics: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability. These 8 after certain modifications can be applied to intelligent systems as well.

To evaluate intelligent system quality an evaluation method defined in ISO/IEC 25020 can be used. The standards ISO/IEC 25020 is coming into replacement of ISO/IEC 15939:2002 with some improvements and adaptation to ISO/IEC 25000 line. The model which lies underneath of the evaluation method is the following: measures define sub-characteristics, having estimate on a given sub-characteristics, quality characteristic is defined. From a given estimates of quality characteristics an integrated quality estimate is coming.

The method of intelligent system quality evaluation implies that given measures, sub-characteristics and characteristics participate in the estimate with specific weights. The weights are defined by experts considering experience, statistical results of previous developments, and other factors.

Major problem of quality evaluation is not characteristic definition – but definition of quality measures. For intelligent systems most developed is the area of search machines. Examples of measures for Internet searches are:

- probability of return to search results;
- probability of result analysis for 20 seconds and returning back to result set;
- average position of link to target pages in result set;
- probability of no clicks on result set links.

The given measures are compatible with requirements and recommendation of ISO/IEC 25020 standard to quality measures. They are defined in relative numbers and normalized in a way that values are changing on [0, 1] interval, closer to 1 corresponds to better intelligent system quality.

Search machines, however is only one class of intelligent systems and for other systems characteristics and measures to evaluate quality will differ significantly.

CONCLUSION

The problem of intelligent system quality evaluation remains unresolved because of different applications of intelligent systems and their variety. At the same time, international standards ISO/IEC 25000 in software quality can be applied to intelligent systems as well, providing guidance on precious estimation of intelligent systems quality.



УДК 004.822:514

КОМПЛЕКСНАЯ МЕТОДИКА ПРОЕКТИРОВАНИЯ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ИНТЕЛЛЕКТУАЛЬНЫХ СПРАВОЧНЫХ СИСТЕМ

Давыденко И.Т.

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

ir.davydenko@gmail.com

В работе рассматривается методика проектирования семантических моделей одного из важнейших классов интеллектуальных систем – интеллектуальных справочных систем. Отдельное внимание уделяется методике проектирования баз знаний интеллектуальных справочных систем.

Ключевые слова: интеллектуальная система, интеллектуальная справочная система, методика проектирования, база знаний.

ВВЕДЕНИЕ

Проблема разработки практически полезных, реально и широко используемых интеллектуальных систем является одной из важнейших в области искусственного интеллекта [Грибова, 2010].

Широко используемым классом интеллектуальных систем являются интеллектуальные обучающие системы [Башмаков, 2003], [Башмаков, 2004], [Голенков и др., 2001]. На рисунке 1 приведена структура интеллектуальной обучающей системы, включающая в себя ряд типовых подсистем, обеспечивающих ее эффективную эксплуатацию и сопровождение.

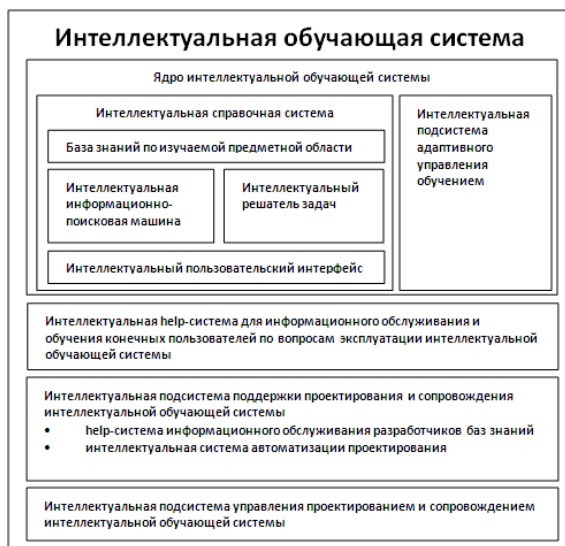


Рисунок 1 – Структура интеллектуальной обучающей системы

Важнейшим видом интеллектуальных систем, входящих в состав интеллектуальной обучающей системы являются интеллектуальные справочные системы, которые обеспечивают информационное обслуживание пользователей, как по изучаемой предметной области, так и по вопросам эксплуатации данной интеллектуальной обучающей системы, а также вопросам ее дальнейшего развития и сопровождения. Подчеркнем, что интеллектуальные справочные системы могут быть самодостаточными и представлять самостоятельный интерес.

Под интеллектуальной справочной системой (ИСС) будем понимать систему, способную отвечать на различные свободно конструируемые вопросы пользователя, а также решать задачи из соответствующей предметной области, то есть осуществлять информационное обслуживание пользователя в заданной предметной области. Такая система включает в себя интеллектуальную информационно-поисковую систему и интеллектуальный решатель задач.

К функциям интеллектуальной справочной системы относятся:

- предоставление пользователю возможности навигации по семантическому пространству предметной области;
- интерпретация любых вопросов пользователя, поиск необходимой информации и представление ее пользователю в удобной для него форме;
- интерпретация формулировок задач пользователя, поиск способов их решения и генерация решений, если они не были найдены в базе знаний;

- анализ деятельности пользователя для оказания ему помощи, а также обучения, что является следующим этапом развития интеллектуальных справочных систем.

Особенностью и интеллектуальностью таких систем является обеспечение возможности пользователю задавать широкий спектр вопросов системе в рамках некоторой предметной области. Интеллектуальная справочная система осуществляет поиск и навигацию по базе знаний, а также генерацию ответа, если он не найден в базе знаний.

Несмотря на большое число работ, ведущихся в данном направлении [ЭЗОП, 2007], [Грибова, 2011], [Geogebra, 2011], [Владимиров и др., 2008] до настоящего времени не существует технологии проектирования интеллектуальных справочных систем, предоставляющих возможность быстро и качественно разрабатывать системы такого класса.

В качестве подхода к построению такого класса систем в работе предлагается комплексная методика проектирования интеллектуальных справочных систем. В основе данной методики лежат следующие принципы:

- поэтапное эволюционное проектирование интеллектуальных справочных систем на основе быстрого прототипирования;
- ориентация на коллективное проектирование компонентов интеллектуальной справочной системы в рамках Open Source проекта;
- ориентация на семантическое представление знаний;
- унификация моделей баз знаний интеллектуальных справочных систем;
- модульное проектирование на основе библиотек типовых многократно используемых компонентов;

и другие принципы массовой семантической технологии проектирования интеллектуальных систем OSTIS [OSTIS, 2011].

1. Принципы организации проектирования семантической модели интеллектуальной справочной системы

Интеллектуальная справочная система состоит из следующих компонентов:

- декларативная часть базы знаний;
- процедурная часть базы знаний, которая представляет собой пакет программ для решения задач из заданной предметной области;
- интеллектуальная информационно-поисковая машина;
- интеллектуальный решатель задач;
- пользовательский интерфейс.

Соответственно, для разработки каждого из этих компонентов используются ряд частных семантических технологий:

- семантическая технология проектирования декларативной части базы знаний;
- семантическая технология проектирования процедурной части базы знаний;
- семантическая технология проектирования интеллектуальных информационно-поисковых машин;
- семантическая технология проектирования интеллектуальных решателей задач;
- семантическая технология проектирования пользовательских интерфейсов интеллектуальных систем.

Разработка интеллектуальной системы сводится к разработке ее логико-семантической модели, то есть формальному описанию структуры интеллектуальной системы и всех ее компонентов. Достоинством такого подхода к проектированию систем является независимость от платформ реализации этих систем, т.е., разработав только лишь семантическую модель интеллектуальной справочной системы, появляется возможность реализовывать эту модель на различных платформах, существенно не изменяя при этом саму модель системы.

Семантическая модель интеллектуальной справочной системы включает в себя семантическую модель базы знаний интеллектуальной системы и семантическую машину обработки знаний этой интеллектуальной системы. Исходя из того, что формальные описания операций, работающих над семантической памятью, также хранятся в базе знаний и являются ее частью, то, по сути, проектирование семантической модели интеллектуальной системы сводится к проектированию семантической модели базы знаний этой системы.

В основе комплексной методики проектирования семантических моделей интеллектуальных систем лежат два аспекта: эволюционное проектирование и коллективное проектирование интеллектуальных справочных систем.

1.1. Эволюционное проектирование интеллектуальной справочной системы

Данный аспект методики проектирования заключается в поэтапной разработке системы на основе быстрого прототипирования. На каждом новом этапе проектирования интеллектуальная система приобретает новые функциональные возможности, которые соответствуют разработке определенных компонентов системы (см. табл. 1).

При таком подходе, изначально необходимо выделить линии развития системы. Они соответствуют разрабатываемым компонентам системы (см. табл. 1), причем учитывается приоритет разработки следующим образом. Изначально проектируется семантическая модель базы знаний, затем происходит непосредственное наполнение базы знаний до заранее оговоренной степени полноты. На следующем эволюционном

этапе развития системы происходит разработка интеллектуальной поисковой машины, которая изначально формируется из уже разработанных стандартных многократно используемых компонентов интеллектуальной поисковой машины, хранимых в библиотеке многократно используемых компонентов. Взяв необходимый набор поисковых операций интеллектуальной машины, мы получим уже готовый работающий прототип системы с базой знаний и минимальным набором поисковых операций (рис.1).

	Версия 1	Версия 2	Версия 3	Версия 4
БЗ	1-я версия	2-я версия	3-я версия	4-я версия
ИШМ	типовое ядро	1-я версия	2-я версия	3-я версия
ИР	типовое ядро	типовое ядро	1-я версия	2-я версия
ШИ	типовое ядро	типовое ядро	типовое ядро	1-я версия

Рисунок 1 – Эволюция версий ИСС

Далее полученный прототип развивается путем наполнения базы знаний и разработкой новых необходимых поисковых операций интеллектуальной поисковой машины до тех пор, пока не достигнет заранее оговоренной степени полноты в наборе операций интеллектуальной поисковой машины.

Таблица 1. Соответствие между компонентами системы и ее функциональными возможностями

Компоненты системы	Функциональные возможности системы
1. База знаний	Хранение знаний о предметной области
2. Пакет программ для решения задач	Решение задач с помощью некоторого хранимого в базе знаний алгоритма
3. Интеллектуальная поисковая машина	Обеспечение интеллектуального поиска и навигации по базе знаний
4. Интеллектуальный решатель задач	Решение задач с применением логического вывода
5. Пользовательский интерфейс	Адаптация к пользователю
6. Help-система с различными уровнями интеллектуализации	Обеспечение помощи пользователю, управление обучением пользователя
7. Подсистема управления проектирования и жизненным циклом системы	Управление процессом проектирования системы (в том числе коллективным), а также управление жизненным циклом системы

Аналогичным образом происходит разработка остальных компонентов системы, то есть на каждом этапе разработки системы в процесс проектирования включается ветвь разработки нового компонента системы, при условии, что остальные компоненты системы достигли минимального заранее оговоренного уровня

полноты разработки. При этом разработка остальных компонентов системы не прекращается, а сам процесс проектирования повторяется циклически на протяжении всего жизненного цикла системы, в том числе при сопровождении системы.

После того, как все этапы, соответствующие линиям развития, пройдены и системой накоплен определенный набор необходимых знаний и навыков, система вводится в эксплуатацию для конечных пользователей, которые в процессе эксплуатации тестируют эту систему на предмет ошибок и недостающих знаний и навыков. Тем самым завершается процесс проектирования интеллектуальной системы как прототипа и начинается процесс проектирования интеллектуальной системы в процессе ее эксплуатации. Этот процесс поддерживается подсистемой управления процессом проектирования и управления жизненным циклом системы. Данная подсистема является частью основной системы и разрабатывается на основе тех же методов и средств, что и основная.

Отличительной особенностью такого похода к проектированию является наличие *на каждом этапе проектирования реально работающего прототипа системы*, что значительно ускоряет введение разрабатываемой системы в эксплуатацию, как для тестирования, так и для конечных пользователей, а также существенное увеличение ее жизненного цикла.

1.2. Коллективное проектирование интеллектуальной справочной системы

Данный аспект заключается в распараллеливании процесса проектирования на независимые ветви, которые могут выполняться различными разработчиками, что существенно сокращает время разработки системы. Однако, возникает вопрос, каким образом можно выделить в рамках проектирования отдельные задачи.

Первый критерий декомпозиции процесса проектирования на отдельные задачи является разделение работ по различным компонентам системы, такие как база знаний, интеллектуальная поисковая машина, решатель задач и пользовательский интерфейс. Все эти компоненты являются важными в рамках справочной системы, однако наиболее значимой является база знаний, так как база знаний для каждой разрабатываемой справочной системы является уникальной, в то время как другие компоненты могут использоваться как предметно независимые в других системах. К тому же, как уже говорилось ранее, проектирование интеллектуальной системы есть не что иное, как проектирование базы знаний, то выделение задач будем рассматривать в рамках проектирования семантической модели базы знаний, что рассмотрим далее более подробно.

2. Методика коллективного проектирования баз знаний интеллектуальных справочных систем

База знаний является одним из ключевых компонентов интеллектуальной справочной системы [Гаврилова и др., 2001], [Гаврилова, 2008], [Хорошевский, 2008]. Разработка этого компонента является трудоемким и продолжительным процессом, а поэтому важным аспектом проектирования базы знаний является организация коллективного проектирования.

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.

Рассмотрим предложенную методику на примере проектирования интеллектуальной справочной системы по геометрии [Давыденко, 2011].

На сегодняшний день существует большое число прикладных систем по геометрии, которые имеют преимущества разного характера (большое число информации различного характера в рамках предметной области, поиск и навигация, решение задач, пользовательский интерфейс и др.), однако не существует такой прикладной системы, которая сочетала бы в себе все эти качественные особенности.

Данная система предназначена как для школьников, преподавателей геометрии, так и для людей, интересующихся геометрией и является попыткой учесть все достоинства и недостатки существующих справочных систем по геометрии в одной системе.

К достоинствам интеллектуальной справочной системы по геометрии можно отнести богатое разнообразие представленных знаний в ее базе (теоретико-множественные связи между понятиями, терминологическое описание понятий, логическая иерархия понятий, аксиоматизация предметной области, описание утверждений различного рода, а также доказательств, описание задач и способов их решений, когнитивные иллюстрации и др.), способность системы отвечать на большое число вопросов пользователя (при этом учитывается полнота ответа на вопрос), при отсутствии ответа в базе знаний система пытается ответить на него с помощью решателя задач.

Система по геометрии имеет две версии реализации. Первая версия системы реализуется в виде семантически структурированного гипертекста в среде MediaWiki на SCn-коде [OSTIS, 2011] с

гипертекстовой навигацией. SCn-код представляет собой способ гипертекстового представления исходных текстов базы знаний. Вторая версия системы представлена в виде семантической сети с интеллектуальным поиском, решателем задач и пользовательским интерфейсом.

Понятие базы знаний тесно связано с понятием предметной области. Семантика базы знаний интеллектуальной системы – это соотношение между базой знаний и описываемой ею предметной областью [Голенков, 2011].

Семантическая структура базы знаний интеллектуальной системы трактуется в рамках семантической технологии проектирования баз знаний интеллектуальных систем как иерархическая система взаимосвязанных между собой предметных областей, которые представляются в базе знаний.

Для выделения иерархической структуры базы знаний необходимо в рамках предметной области явно выделить *класс исследуемых объектов*, *класс вторичных объектов*, построенных на основе исследуемых, *класс вспомогательных объектов*, через связи с которыми описываются некоторые характеристики исследуемых объектов, *отношения*, *связки которых связывают только исследуемые объекты между собой*, а также *отношения*, *связки которых связывают исследуемые объекты со вспомогательными*.

При проектировании интеллектуальной справочной системы по геометрии рассматривалась предметная область геометрии Евклида, которая является стационарной предметной областью.

В предметной области геометрии Евклида исследуемыми объектами являются геометрические фигуры и пространственные отношения между ними. Исходя из этого, семантическая сеть, которая представляет собой информационную модель описываемой предметной области, включает следующие ключевые узлы, являющиеся классами объектов исследования геометрии: геометрическая фигура, точка, отрезок, луч, линия, плоскость, многоугольник, треугольник, четырехугольник и др. К ключевым узлам, являющимися отношениями и составляющими предмет исследования, относятся: параллельность, перпендикулярность, пересечение, конгруэнтность, сторона, внутренний угол, лежать между, лежать против, вписанность и др.

В рамках предметной области геометрии Евклида используются такие понятия, как число, сложение, вычитание, деление, которые являются объектами и предметом исследования предметной области числовых моделей и являются внешними понятиями, по отношению к Геометрии Евклида. Также в Геометрии Евклида используются понятия, являющиеся объектами и предметом исследования теории измерений, теории множеств, теории отношений, русского языка и других предметных областей.

При структуризации предметной области по

различным критериям в рамках основной предметной области выделяются различные классы предметных областей, описывающие основную предметную область с точки зрения одного признака. Фрагменты, описывающие каждую из предметных областей, будем называть *разделами базы знаний*. Разработка каждого такого раздела базы знаний соответствует этапу проектирования семантической модели базы знаний.

Основываясь на аспекте коллективного проектирования комплексной методики проектирования интеллектуальных справочных систем, 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.

Таким образом, структуризация базы знаний с учетом ее связи с предметной областью может быть использована для доведения процесса декомпозиции процесса проектирования на задания для различных разработчиков, при условии, что они будут согласовывать ключевые узлы описываемой предметной области (т.к. данный процесс автоматизировать не представляется возможным).

По такому принципу разрабатываются все компоненты системы, а основой для интеграции всех компонентов служит унифицированная модель представления знаний на основе SC-кода.

Рассмотрение процесса проектирования с учетом аспекта коллективного проектирования в рамках семантической технологии проектирования интеллектуальных систем дает новые возможности для организации и управления процессом проектирования интеллектуальных систем.

Организация процесса проектирования интеллектуальных систем осуществляется специализированной подсистемой управления коллективным проектированием интеллектуальной системы (данная подсистема является частью основной системы и разрабатывается на основе тех же методов и средств, что и основная система), в задачи которой входят:

- синхронизация ветвей параллельной разработки
- создание заданий для разработчика
- назначение статуса задания на текущий момент времени
- классификация заданий по приоритету
- управление сроками выполнения заданий
- интеграция со средствами разработки
- управление жизненным циклом системы.

В рамках семантической технологии проектирования баз знаний выделены следующие *этапы проектирования семантической модели базы знаний*:

- уточнение структуры описываемой предметной области – на данном этапе проводится уточнение объекта и предмета исследования описываемой предметной области, а также уточнение набора вспомогательных объектов, связь с которыми имеет существенное значение для рассмотрения исследуемых объектов;
- построение *предметной области, являющейся теоретико-множественной онтологией* рассматриваемой предметной области – на данном этапе все понятия описываемой предметной области рассматриваются с точки зрения теоретико-множественных отношений между ними (рис.2);
- построение *предметной области, являющейся логической онтологией* рассматриваемой предметной области – систематизация всех понятий по логическим уровням, выделяемых в рамках рассматриваемой предметной области, с точки зрения анализа их определений (что на основе чего определяется);
- построение *предметной области, являющейся терминологической онтологией* описываемой предметной области – описание идентификации терминов предметной области и их этимологии;
- построение *предметной области логического описания* рассматриваемой предметной области – описание множества логических формул (высказываний), интерпретируемых на рассматриваемой предметной области, а также их систематизация на основе их доказательств (рис. 3, 4);
- построение *предметной области вопросов и информационных задач* для заданной предметной области;
- построение *предметной области когнитивных мультимедийных иллюстраций и библиографических источников* для заданной предметной области [Зенкин, 1991].

Таким образом, проектирование базы знаний можно рассматривать как процесс построения некоторой исходной предметной области и процесс наращивания указанной предметной области целым рядом надобластей, у каждой из которых есть свой класс исследуемых объектов.

Так при проектировании базы знаний интеллектуальной справочной системы по геометрии предметная область геометрии Евклида рассматривалась как совокупность некоторых более частных предметных областей, каждая из которых представляет собой набор ключевых понятий и отношений между ними.

На множестве предметных областей могут быть заданы следующие отношения: включение, объединение, пересечение, декомпозиция, гомоморфизм, изоморфизм, теоретико-множественная онтология, логическое описание, логическая онтология. Таким образом, мы можем рассматривать некую *метаобласть*, объектами исследования которой являются всевозможные

геометрическая фигура

▷ геометрическое тело

§ Разбиение по признаку *размерности*:

- дискретная геометрическая фигура
- линия
- конфигурация линий
- поверхность
- конфигурация поверхностей
- геометрическое тело
- конфигурация геометрических тел
- геометрическая фигура смешанной размерности

§ Разбиение по признаку *ограниченности размера*:

- геометрическая фигура ограниченного размера
- геометрическая фигура неограниченного размера

§ Разбиение по признаку *существования граничных точек*:

- геометрическая фигура, имеющая граничные точки
- геометрическая фигура, не имеющая граничных точек

§ Разбиение по признаку *расположения в одной плоскости*:

- планарная фигура
- непланарная фигура

§ Разбиение по признаку *расположения на одной прямой*:

- прямолинейная фигура
- непрямолинейная фигура

Рисунок 2 – Фрагмент предметной области теоретико-множественного описания предметной области
Геометрии Евклида

треугольник

§ Утверждения:

- Текст Утв. (треугольник; тройка точек; вершина*; непрямолинейная фигура)
•• [Для каждого треугольника тройка точек, являющихся вершинами этого треугольника, принадлежит классу непрямолинейных фигур.]
- Текст Утв. (треугольник; внутренний угол*; мера угла*)
•• [Сумма мер углов треугольника равна 180 угловых градусов.]
•• [Для каждого треугольника выполняется следующее числовое соотношение
 $\alpha + \beta + \gamma = 180^\circ$ где
 α, β, γ - величины внутренних углов треугольника.]
- Текст Утв. (треугольник; внутренний угол*; острый угол, тупой угол, прямой угол)
•• [В любом треугольнике либо все углы острые, либо два угла острые, а третий тупой или прямой.]
- Текст Утв. (треугольник; окружность; вписанность*; центр*)
•• [Центр окружности, вписанной в треугольник, есть точка пересечения биссектрис треугольника.]
- Текст Утв. (треугольник; сторона*; внутренний угол*; конгруэнтность*)
= Признак равенства треугольников по 2-м сторонам и углу между ними
•• [Если две стороны и угол между ними одного треугольника равны соответственно двум сторонам и углу между ними другого треугольника, то такие треугольники равны (конгруэнтны).]
- Текст Утв. (треугольник; сторона*; внутренний угол*; прилежащий*; конгруэнтность*)
= Признак равенства треугольников по стороне и прилежащим к ней углам
•• [Если сторона и прилежащие к ней углы одного треугольника равны соответственно стороне и прилежащим к ней углам* другого треугольника, то такие треугольники равны (конгруэнтны).]

Рисунок 3 – Фрагмент предметной области логического описания Геометрии Евклида

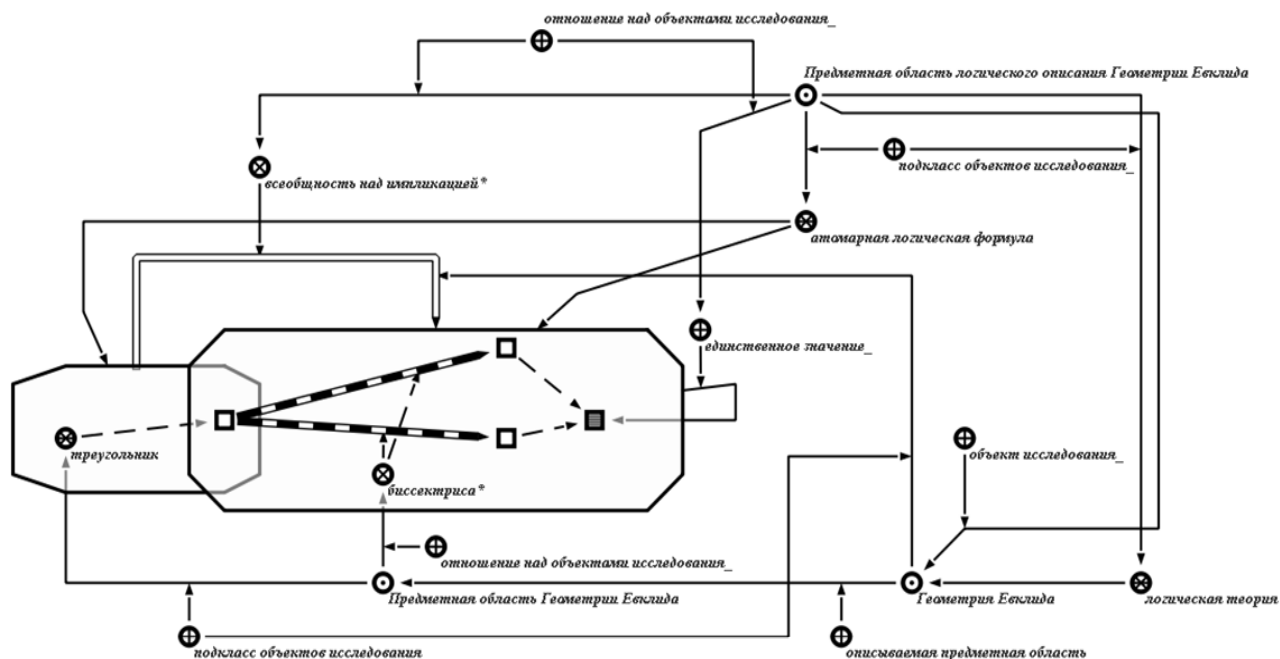


Рисунок 4 – Фрагмент предметной области логического описания Геометрии Евклида, описывающий теорему «У каждого треугольника любая пара его биссектрис имеет одну и ту же общую точку»

предметные области.

Таким образом, семантическая структура базы знаний представляет собой иерархическую систему описываемых ею предметных областей, надстраиваемых над заданной основной предметной областью.

В основе комплексной методики проектирования семантических моделей интеллектуальных систем лежит *задачно-ориентированный подход*. То есть каждый компонент системы разрабатывается исходя из определенного для него круга решаемых задач, которые определяются тестовым сборником. На основе этого тестового сборника задач проектируется первая версия компонентов системы.

Фрагмент тестового сборника задач для системы по геометрии представлен ниже:

- Запросы основных свойств заданного объекта
 - *Какими свойствами обладают прямоугольные треугольники*
- Сколько-вопросы
 - *Какова (чему равна) площадь Треугольника (TA, TB, TC)*
- Запросы минимального высказывания (минимального фрагмента базы знаний), описывающего семантически значимую связь между всеми объектами заданного множества объектов
 - *Как связаны между собой понятия луча и прямой*
- Запросы пар высказываний, описывающих сходные (похожие, аналогичные) свойства заданных двух объектов
 - *В чем заключается сходство Понятия отрезка и Понятия плоского угла*

- Запросы одноуровневой классификации заданного множества
 - *Как разбивается (классифицируется) Понятие треугольника*
- Запросы всех известных подмножеств заданного множества
 - *Какие классы геометрических фигур являются подклассами Класса планарных фигур (т.е. классами, которые являются подмножествами Множества всевозможных планарных фигур)*
- Запросы надмножеств заданного множества
 - *К каким классам геометрических фигур относится трапеция*

Неотъемлемым компонентом интеллектуальной справочной системы, определяющим функциональные возможности конкретной интеллектуальной справочной системы, являются информационно-поисковые и навигационные операции [Голенков и др., 2001], [OSTIS, 2010].

Семантическая технология компонентного проектирования средств навигации и поиска в семантических сетях [Житко, 2011] предоставляет технологию разработки, хранения и повторного использования информационно-поисковых sc-операций. Ключевым элементом технологии является библиотека совместимых ip-компонентов, использование которой позволит разрабатывать машину обработки знаний интеллектуальной системы из готовых функциональных элементов – sc-операций. Это позволяет снизить расходы на разработку и реализацию средств поиска и навигации в интеллектуальной системе.

В интеллектуальных системах, основанных на технологии OSTIS, информация представляется в

виде семантической сети, что позволяет оперировать не только фактографической информацией, но и осуществлять навигацию по установленным отношениям в рамках предметной области прикладной вопросно-ответной системы. Отметим также важность интеллектуальных справочных, т.к. они составляют основу интеллектуальных систем.

Проектирование информационно-поисковых основано также на задачно-ориентированном подходе, то есть начинается с составления тестового сборника вопросов системе.

Тестовый сборник вопросов для системы по геометрии приведен выше.

Для задания вопроса системе используется семантический язык вопросов.

Семантический язык вопросов предназначен для организации взаимодействия, как между пользователем и интеллектуальной системой, так и внутри системы между различными её компонентами.

Пример неатомарного запроса, формулируемого как «Приведите примеры каждого класса понятия *треугольник*», приведен на рисунке 5.

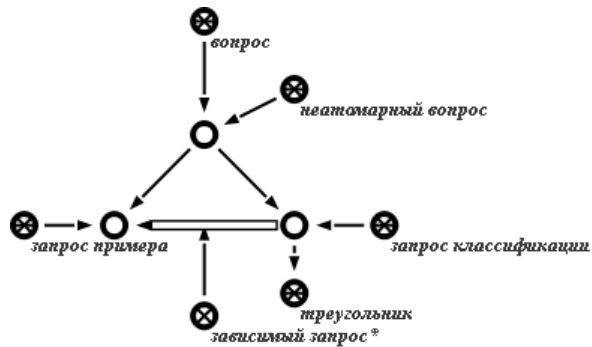


Рисунок 5 - Пример неатомарного запроса

В приведённом примере на первом шаге будет найдена полная классификация множества *треугольник*, и далее во второй части вопроса, каждому из найденных классов будет найден пример.

Этот же вопрос можно задать, используя шаблоны изоморфного поиска в формулировке вопроса, пример приведен на рисунке 6.

Интеллектуальность справочной системы определяется многообразием вопросов, на которые система может дать ответ [Федоров, 1994]. Из этого следует, что интеллектуальность справочных систем будет зависеть и от множества операций информационного поиска, т.к. именно они определяют возможности системы ответа на поставленные вопросы. При этом, при отсутствии ответа в базе знаний, система должна генерировать ответ на вопрос на основе имеющихся знаний, что обеспечивается интеллектуальным решателем задач.

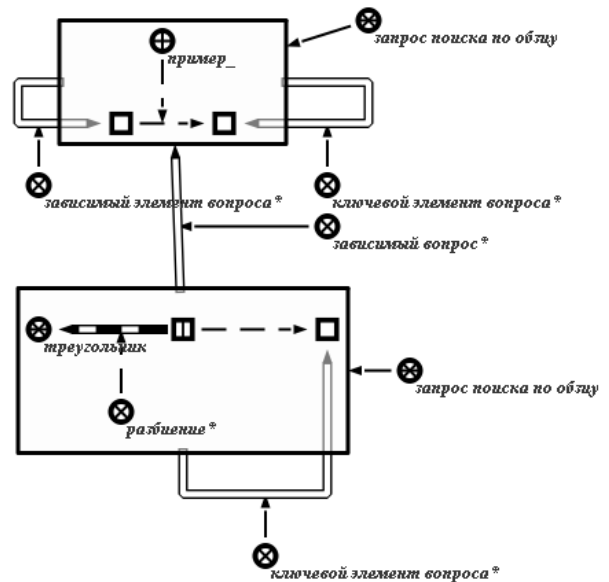


Рисунок 6 - Пример неатомарного вопроса с использованием шаблонов изоморфного поиска

Технология проектирования интеллектуальных решателей задач основана на задачно-ориентированной методологии [Заливако и др., 2012]. В связи с этим проектирование системы операций состоит из четырех основных этапов:

- создание тестового сборника задач, которые решаются в рамках исследуемой предметной области;
- определение набора предметно независимых операций, которые будут использоваться при решении задач из тестового сборника;
- уточнение семантической спецификации каждой из указанных операций;
- реализация и отладка операций.

В общем случае можно выделить следующие предметно независимые классы задач:

- задачи синтеза доказательства;
- задачи верификации;
- задачи синтеза способа (алгоритма) решения;
- задачи анализа;
- задачи классификации.

В качестве примера предметной области рассмотрим геометрию Евклида. Тогда классификация задач тестового сборника будет выглядеть следующим образом:

- по способу решения:
 - вычислительные задачи;
 - задачи на доказательство;
 - задачи на построение;
 - комбинированные задачи;
- по объекту решения:
 - задачи о точках;
 - задачи о прямых и отрезках;
 - задачи о треугольниках;
 - задачи о многоугольниках;
 - задачи об окружностях;
 - и другие;
- по размерности пространства:
 - задачи по планиметрии;

- задачи по стереометрии.

Подробнее методика проектирования интеллектуальных решателей задач рассмотрена в работе [Заливако и др., 2012].

При работе пользователя со справочной системой большое значение имеет организация диалога пользователя и системы, которая обеспечивается пользовательским интерфейсом.

В рамках семантической технологии OSTIS пользовательский интерфейс интеллектуальной справочной системы представляет собой специализированную интеллектуальную систему, которая разрабатывается по тем же принципам и с использованием такой же методики, как и любая другая интеллектуальная система, и состоит из базы знаний, машины обработки знаний [Корончик, 2012].

Более подробно методика проектирования пользовательских интерфейсов интеллектуальных систем рассмотрена в работе [Корончик, 2012].

ЗАКЛЮЧЕНИЕ

В работе приведена комплексная методика проектирования интеллектуальных справочных систем, основанная на массовой семантической технологии проектирования компьютерных систем различного уровня интеллекта OSTIS. Основными особенностями и достоинствами предлагаемой методики проектирования заключаются в следующем:

- расширенный семантический язык вопросов;
- семантическая модель информационного поиска;
- семантическая модель структуризации баз знаний, на основе которой обеспечивается разделение задач между разработчиками для параллельной разработки компонентов интеллектуальных справочных систем, что влечет за собой сокращение сроков разработки интеллектуальных справочных систем;
- интеллектуальная help-система для информационного обслуживания и обучения конечных пользователей интеллектуальной справочной системы;
- интеллектуальная help-система для информационного обслуживания и обучения разработчиков интеллектуальной справочной системы;
- семантическая модель интеллектуальной справочной системы;
- продолжение проектирования интеллектуальной справочной системы, направленного на постоянное ее совершенствование, непосредственно в ходе эксплуатации;
- поэтапное эволюционное развитие основных компонентов интеллектуальной справочной системы (база знаний, интеллектуальная информационно-поисковая машина, интеллектуальный решатель задач,

интеллектуальный пользовательский интерфейс);

- обеспечение модульной разработки интеллектуальных справочных систем;
- единый подход к проектированию всех компонентов интеллектуальных справочных систем на основе семантической модели интеллектуальной справочной системы.

Результаты, приведенные в работе, апробируются в рамках открытого проекта OSTIS [OSTIS, 2012].

Данная работа выполнялась при поддержке гранта БРФФИ-РФФИ №Ф10Р-149, а также гранта БРФФИ №Ф10М-085.

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

[Владимиров и др., 2008] Программный комплекс «УДАВ»: практическая реализация активного обучаемого логического вывода с линейной вычислительной сложностью на основе миварной сети правил / Владимиров А.Н., Варламов О.О., Носов В.В., Потапова Т.С. // Конгресс по интеллектуальным системам и технологиям. Труды конгресса; – М.: ФИЗМАТЛИТ, 2010.

[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 и др., 2001] Интеллектуальные обучающие системы и виртуальные учебные организации / Голенков В.В. [и др.]; под ред. В.В. Голенкова – Минск, 2001. – 488с.

[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, 2010] Грибова, В.В. Системы управления интеллектуальными Интернет-приложениями. / Грибова В.В., Клещев А.С., Шалфеева Е.А.// Владивосток: ИАПУ ДВО РАН, 2010. 31 с.

[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, 2011] Давыденко И. Т. Интеллектуальная справочная система по геометрии / И. Т. Давыденко, В. А. Житко, С. С. Заливако, Д. Н. Корончик, С. Г. Мошенко, О. Ю. Савельева, С. С. Старцев, Д. В. Шункевич //Материалы международной научно-технической конференции OSTIS-2011, стр. Минск БГУИР

[Житко, 2011] Житко В. А. Технология компонентного проектирования средств навигации и поиска в семантических

сетях / В. А. Житко //Материалы международной научно-технической конференции OSTIS-2011, стр. Минск БГУИР

[Загоруйко и др., 2008] Загоруйко, Ю.А., Боровикова, О.И. Подход к построению порталов научных знаний / Ю.А. Загоруйко [и др.] // Автометрия. – 2008 – № 1, Т. 44, – С. 100–110.

[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, 2012

[Зенкин, 1991] Зенкин, А.А. Когнитивная компьютерная графика/ А.А. Зенкин; - М. : Наука, 1991.

[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

[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, 2012

[Открытая математика, 2010] Открытая математика. Планиметрия [Электронный ресурс]. Минск, 2010. – Режим доступа: <http://mathematics.ru/courses/planimetry/design/index.htm> – Дата доступа: 17.11.2010.

[Федоров, 1994] Федоров, Б.И. Логика компьютерного диалога / Б.И. Федоров, З.О. Джалишвили. – Москва : Онега, 1994.

[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]. Минск, 2010. – Режим доступа: <http://ezop-project.ru/drupal5/>. – Дата доступа: 14.11.2011.

[Geogebra, 2011] Сайт системы динамической геометрии GeoGebra [Электронный ресурс]. – Режим доступа: <http://www.geogebra.org> – Дата доступа: 11.11.2011.

[Geometry Expressions, 2011] Geometry Expressions [Электронный ресурс]. – Режим доступа: <http://www.geometryexpressions.com/>. – Дата доступа: 11.11.2011.

[OSTIS, 2011] Проект OSTIS [Электронный ресурс]. Минск, 2011. – Режим доступа: <http://ostis.net/>. – Дата доступа: 11.12.2011.

INTEGRATED METHODOLOGY OF DESIGN OF THE INTELLIGENT REFERENCE SYSTEM SEMANTIC MODEL

Davydenko I.T.

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

ir.davydenko@gmail.com

INTRODUCTION

In this work we will examine intelligent question-answering systems design.

Let's consider intelligent question-answering system as system class, which is able to answer different freely

constructed user's questions, and also solve problems from the corresponding knowledge domain.

Providing an opportunity for user to ask as wide as possible range of questions to the system, within the limits of some knowledge domain, is a particular feature and intellectuality illustration of intelligent question-answering systems. Such system makes search and navigate in the knowledge base, also it generates answer, if it is not found in user's knowledge base in certain knowledge domain.

MAIN PART

Within the limits of OSTIS [Голенков, 2011] technology, intelligent question-answering system consists from knowledge base, intelligent search machine, program package for problem solving, intelligent problem solver, user interface.

Intelligent system design comes to design of its logic-semantic model, in other words formal description of intelligent system and all its components structure. The advantage of such system design method is independence from realizing platforms. In other words, if you designed just semantic model of intelligent question-answering system, you will be able to realize this model on different platforms, without changing greatly your system model.

At the heart of evolutionary collective design of intelligent systems semantic models complex principle there are two aspects: *evolutionary intelligent question-answering system design* and *collective intelligent question-answering system design*.

By this principle all system components are developed. Basis for all components integration is uniform knowledge representation model based on SC-code.

For the effective organization of intelligent question-answering system knowledge bases design, it is necessary to include the next components to the main system composition as subsystems:

- Intelligent design support and system maintenance subsystem;
- Help-system for knowledge bases makers informational servicing;
- Design automation intelligent system;
- Intelligent design control and system maintenance subsystem.

CONCLUSION

In the given paper short description of integrated method of design of integrated method of design of the intelligent reference system semantic model are present. The article describes the design of the geometry intelligent reference system, which is designed with open semantic technology of design intelligent systems.

Given results are tested in an open project OSTIS [OSTIS, 2011].

СЕКЦИЯ № 6

ПРИКЛАДНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ, ОСНОВАННЫЕ НА СЕМАНТИЧЕСКИХ СЕТЯХ

SECTION № 6

APPLICATION-ORIENTED INTELLECTUAL SYSTEMS BASED ON SEMANTIC NETWORKS

Данная секция преследует следующие цели:

- Показать многообразие сфер применения интеллектуальных систем (комплексная автоматизация интеллектуальных процессов в различных профессиональных областях деятельности; медицинская диагностика; компьютерное обучение; комплексная интеграция всех известных знаний в заданной области исследований);
- Показать преимущества использования семантических моделей при проектировании прикладных интеллектуальных систем;
- Показать то, что обеспечение семантической совместимости прикладных интеллектуальных систем переводит их на качественно новый уровень.

Работы, представленные в секции, затрагивают различные области применения семантических технологий. Как правило, новое появляется в тех областях, где традиционные подходы сталкиваются с неразрешимыми проблемами. К таким областям традиционно относят телекоммуникационные и Internet-технологии (где быстрые темпы развития ставят задачи, не решаемые с помощью традиционных технологий), а также области, связанные с решением различных прикладных медицинских задач, которые и представлены в данной секции.

В работе Майбороды Ю.И. представлен опыт применения миварных технологий для решения задачи частотно-территориального планирования телекоммуникационных сетей. Близка по области применения к работе Майбороды Ю.И. и работа Гладуна А.Я, Рогушиной Ю.В, в которой авторы делятся опытом повышения качества обслуживания за счет использования семантических технологий гетерогенной беспроводной среде.

В работе Вовк А.И., Рубан Ю.Я., Гирнык Д.А. представлены подходы, позволяющие ликвидировать последствия эффекта “языковых ножниц” за счет использования для представления различных математических текстов семантических технологий. Такое представление с одной стороны будет понятно инженерам по знаниям, а с другой стороны будет удобным для представления знаний экспертам в области математики. В работе Глоба Л.С., проф., Кузин И.А., Мочалкина К.С., Новогрудская Р.Л. представлен подход к организации информационно справочного портала, информационное ядро которого строится с использованием онтологий

Авторский коллектив Клещев А.С., Черняховская М.Ю., Шалфеева Е.А. в своем докладе исследует возможности автоматизации интеллектуальных процессов в профессиональной деятельности на примере медицины. Еще одна работа, посвященная медицинской тематике, авторов Сметанин Ю.М, Сметанин М.Ю.. В ней авторы сделали попытку адаптировать существующие методы логического вывода к задачам медицинской диагностики.

В работе Козлова О.А., Михайлова Ю.Ф. предложен подход по формированию индивидуальной траектории изучения студентом некоторой предметной области знаний с помощью искусственной нейронной сети теории адаптивного резонанса.

ВОПРОСЫ, ВЫНОСИМЫЕ НА ОБСУЖДЕНИЕ КРУГЛОГО СТОЛА СЕКЦИИ № 6:

- *Каковы перспективные направления применения интеллектуальных систем, основанных на семантических сетях.*
- *Зачем практически нужна логико-семантическая совместимость интеллектуальных систем.*
- *Как могут быть использованы семантические сети для обеспечения информационной безопасности интеллектуальных систем.*
- *Можно ли в перспективе говорить о порталах знаний как об основном виде оформления научных результатов.*
- *Какова перспектива превращения Wikipedia в интеллектуальную справочную систему.*



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8

МЕТОДИКА РЕШЕНИЯ ЗАДАЧИ ЧАСТОТНО-ТЕРРИТОРИАЛЬНОГО ПЛАНИРОВАНИЯ НА ОСНОВЕ ПРИМЕНЕНИЯ ПРОДУКЦИОННЫХ ЭКСПЕРТНЫХ СИСТЕМ И ЭВРИСТИЧЕСКИХ АЛГОРИТМОВ

Майборода Ю.И.

Московский физико-технический институт (государственный университет)

г. Долгопрудный, Россия

juramaiboroda@gmail.com

В работе представлена методика решения задачи частотно-территориального планирования сетей связи и вещания, которая заключается в следующем: создаются ограничения в пространстве возможных решений за счет миварного метода логического вывода по базе знаний, содержащей эвристики, после чего происходит оптимизация методом генетического поиска, реализация которого интегрируется в экспертную систему в виде отдельного сервиса.

Ключевые слова: мивар, экспертные системы, генетический поиск, частотно-территориальное планирование

ВВЕДЕНИЕ

На сегодняшний день методы искусственного интеллекта (ИИ) широко применяются для создания систем автоматизированного проектирования, автоматизированных систем управления и принятия решений. Так, например, на основе эволюционных алгоритмов создаются системы для проектирования СБИС, нейронные сети широко распространены в химических и биохимических исследованиях, а также применяются для предсказания финансовых временных рядов, онтологии используются для описания бизнес процессов. Кроме того сделаны большие продвижения в разработке методов распознавания текстов, изображений, звуков. Большой популярностью также пользуются и экспертные системы, основанные на знаниях, которые находят своё применение в различных предметных областях таких как: химия, электроника, медицина, техника, геология (разведка полезных ископаемых), компьютерная техника. [Джарратано и др., 2007]

Тем не менее, в большинстве случаев все эти методы лишь частично заменяют человека, т.к. ориентированы на решение задач в узкоспециализированных областях. Однако, на сегодняшний день представляется возможным для решения сложных проблем применять комплексный подход, основанный на использовании нескольких методов ИИ. Такой подход позволяют реализовать экспертные системы (ЭС), созданные на основе

миварных технологий, конструирующие логические цепочки из продукционных правил, процедур, сервисов, которые могут являться, в том числе, реализациями других методов ИИ. [Варламов, 2002] [Санду, 2010]

В данной работе рассмотрен подход совместного использования миварной экспертной системы и метода генетического поиска на примере решения задачи частотно территориального планирования сетей связи и вещания.

1. Постановка задачи частотно-территориального планирования (ЧТП) сетей связи и вещания

У компании есть K пунктов для установки передающих устройств (ПУ) с координатами (Latj, Longj). Необходимо установить N передатчиков ($1 \leq N \leq K$) в этих пунктах с такими параметрами (P_i – мощность, H_i – высота, D_i – диаграмма, Pol_i – поляризация), чтобы получить максимальное количество слушателей P_{max} . Заданы определённые денежные ограничения в размере S руб.

2. Методика решения задачи ЧТП

Задача ЧТП является важнейшей проблемой в области радио-телекоммуникаций. Для того чтобы получить частотно-территориальный план, человеку-эксперту необходимо перебрать, по сути, огромное число всевозможных вариантов

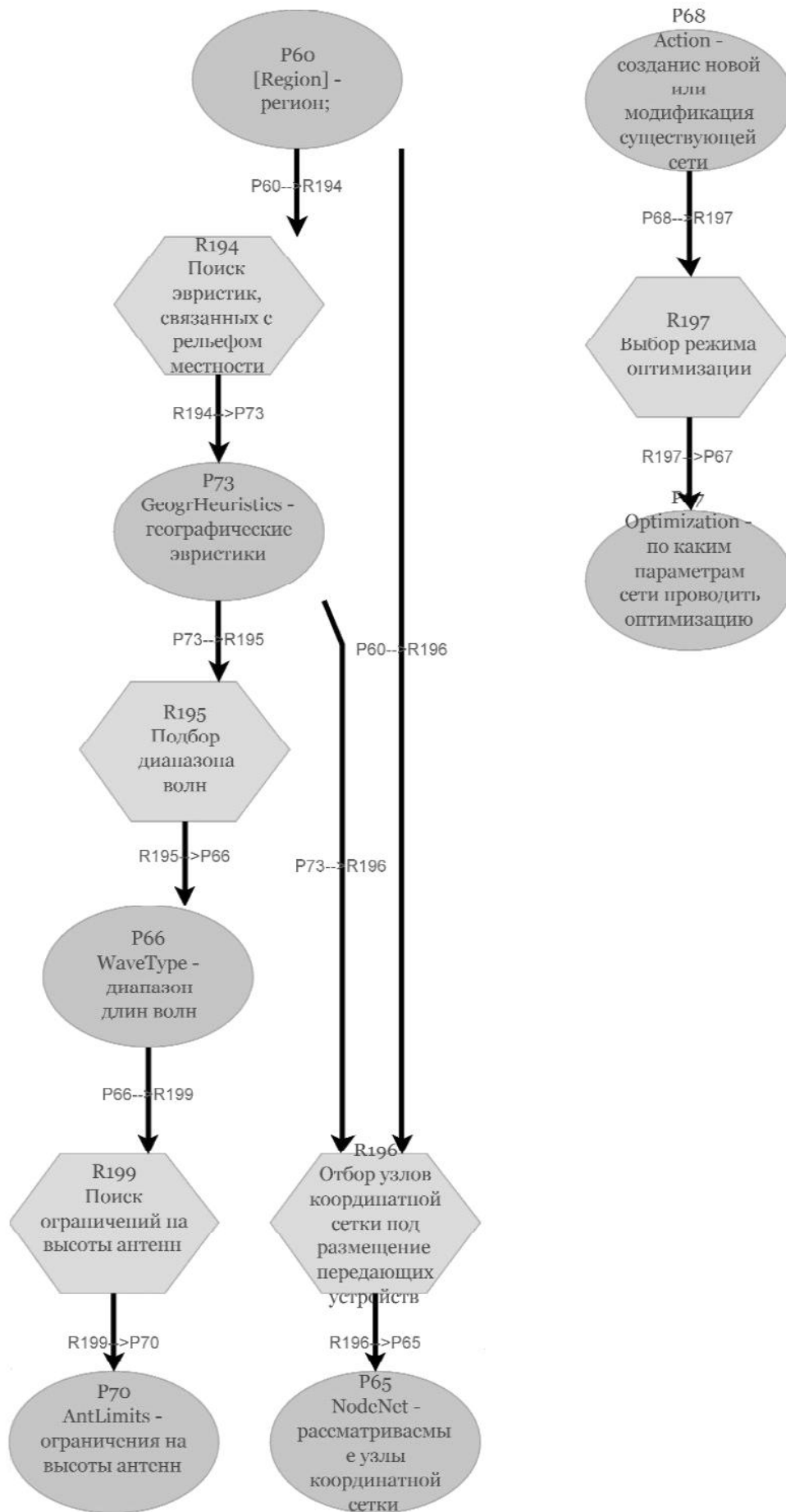


Рисунок 1 – Пример логической цепочки, построенной ПК "УДАВ", для создания ограничений в задаче ЧТП

конфигураций сети, каждый из которых зависит от количества передатчиков, их расположения, диаграмм направленности (ДН) антенн, высот и поляризаций антенн, помеховой обстановки и т.д. Например, если планируется сеть в СВ/ДВ диапазонах (оценочно количество номиналов мощностей порядка 20, номиналов высот 30) с $N = 6$, $K = 10$, и мы учитываем лишь такие параметры, как мощность передатчика и высоту антенны, количество возможных решений составляет:

$$\frac{K^N}{(K-N)!} 30^N 20^N \cong 7 * 10^{21}$$

а если рассматривать полное пространство решений, то его порядок составляет до $\sim 10^{40}$. Естественно человек не может перебрать все возможные варианты и в итоге выбирает одну конфигурацию из ряда допустимых, которая обычно приемлема, но она далеко не всегда является оптимальной.

Современное развитие вычислительных мощностей, а также математических инструментов позволяет уже сейчас решать сложные многопараметрические задачи, одной из которых и является задача оптимального ЧТП сетей связи и вещания.

Алгоритм решения этой задачи логически делится на 2 этапа:

1. Создание ограничений на параметры ПУ сети с помощью общепризнанных эвристик.
2. Решение оптимизационной задачи ЧТП сети на основе полученных ограничений на её параметры.

Первый этап непосредственно моделирует действия человека. К примеру, когда эксперт в области планирования сетей связи (вещания) видит картографическое изображение некоторого региона, он может определить примерное количество передающих станций, необходимое для размещения, диапазон используемых радиоволн (в зависимости от рельефа или "забитости спектра"), ограничения на высоты антенн и мощности передатчиков. Т.е. эксперт выстраивает некоторую логическую цепочку для получения ограничений на параметры конфигурации сети.

Моделирование этой ситуации делается с помощью ПК "УДАВ" [Владимиров и др., 2010], который создает ограничения на параметры ПУ из эвристик, находящихся в Базе Знаний (БЗ). Примеры таких эвристик:

«ЕСЛИ 'действие' = 'планирование новой сети', ТО 'вариативность' = 'по всем параметрам'»;

«ЕСЛИ 'действие' = 'улучшение существующей сети', ТО 'вариативность' = 'по высоте антенны и по мощности передатчика'»;

Если планируются сети радиовещания на СВ/ДВ или ОВЧ ЧМ, то к ним могут быть применимы

следующие правила:

*«ЕСЛИ 'тип местности' = 'гористая', ТО 'диапазон волн' = 'ДВ'»;**

*«ЕСЛИ 'размер региона' = 'маленький' И 'плотность населения' = 'большая', ТО 'диапазон волн' = 'ОВЧ'»;**

* это эвристики, полученные в ходе экспресс-анализа и предварительного ознакомления с предметной областью, и значениям 'гористая', 'большая', 'маленький' планируется дать количественные оценки на этапе полномасштабного сбора знаний с экспертов.

Таким образом, мы сразу отсекаем варианты, которые даже не стоит рассматривать, сокращая, таким образом, пространство возможных решений задачи. При составлении Базы Знаний используются общепризнанные среди экспертов эвристики, взятые, как из опыта экспертов, так и из нормативных документов, санитарных норм.

Если эксперт будет не согласен с логической цепочкой, предложенной ПК, то "УДАВ" даст ему возможность изменить значения параметров, вызывающих сомнения, и составит новую с учётом введённых изменений, т.е. будет реализована опция "что будет, если...?".

На рис. 1 представлен пример логической цепочки, построенной с помощью ПК "УДАВ". Овалами на рисунке обозначены параметры, а шестиугольниками – правила. На этом рисунке показано, что выбирая регион и действие (развертывание новой сети или модернизация существующей), мы получаем ограничения на расположения ПУ, ограничения на высоты антенн и режим оптимизации, т.е. какими параметрами ПУ можно варьировать. Например, если проводится модернизация существующей сети, то не нужно варьировать расположением уже существующих ПУ (в модернизацию сети также может входить процедура размещения дополнительных ПУ, для которых необходимо варьировать этим параметром), в отличие от случая, когда развертывается новая сеть. Стоит отметить, что на рисунке представлена лишь одна логическая цепочка из ряда возможных, коих в данной задаче существует очень большое количество.

После получения ограничений ПК приступает к выполнению этапа 2. На этом этапе решается задача оптимального частотно-территориального планирования сети связи. Оптимизация может выполняться рядом эвристических алгоритмов, например, алгоритмами генетического поиска, алгоритмом поиска с запретами (Tabu Search).

Предлагаемый метод процесса оптимизации с помощью алгоритма генетического поиска [Davis, 1991] [Курейчик, 2002]:

1. Инициализация популяции генотипов. Вычисляется приспособленность каждого генотипа, а затем средняя приспособленность популяции.

Начальная популяция может задаваться как произвольным образом с помощью генератора случайных чисел, так и с помощью человека. Таким образом, во втором случае программа будет улучшать варианты, предложенные человеком.

2. Оценка каждого генотипа в популяции.

3. Создание новых генотипов посредством скрещивания текущих генотипов, т.е. применение оператора кроссинговера, применение операторов мутации и рекомбинации.

4. Устранение генотипов из популяции, чтобы освободить место для новых.

5. Оценка новых генотипов и вставка их в популяцию.

6. Если время исчерпано, то остановка и возврат к наилучшему генотипу, если нет, то переход к пункту 3.

Следует отметить, что в данном случае отдельным возможным решением является генотип, а не хромосома, как это обычно бывает при решении задач с помощью алгоритмов генетического поиска. Это связано с тем, что в данном случае мы ищем решение в расширенном пространстве:

{расположение, мощность передатчика, высота антенны, ДН антенны, поляризация антенны}

Таким образом, каждый генотип состоит из N хромосом, где N – количество таких подпространств как {мощность}, {высота антенны}, и.т.д. Длина одной хромосомы равна n позициям (длина одной позиции может быть больше одного символа, однако одинакова для каждого ПУ в пределах одной хромосомы), где n – количество ПУ сети, а каждая такая позиция определяет конкретную характеристику соответствующего ПУ. Например, пусть в сети находятся два ПУ с мощностями 1 кВт и 5 кВт, и под каждую позицию выделено 3 символа. Тогда если мы рассмотрим хромосому, отвечающую за мощности передатчиков сети, то она будет выглядеть следующим образом 001101 (в данном случае выбрано двоичное представление чисел).

Целевой функцией в данном случае является покрытие сети по населению, и определяется следующим образом:

$$ЦФ = \sum_{i=1}^N \Pi_i - \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^N \Pi_{ij}$$

где Π_i – охват по населению i-ого ПУ, Π_{ij} – количество людей, живущих на территории, покрываемой как станцией i, так и станцией j. Что касается Π_i и Π_{ij} , то формулы для их расчета будут варьироваться в зависимости от типа службы, для которой планируется сеть.

ЗАКЛЮЧЕНИЕ

На сегодняшний день оказывается возможным решать задачи в сложных предметных областях с помощью применения комплекса методов искусственного интеллекта. В частности, для проблемы частотно-территориального планирования сетей связи и вещания выбран подход, состоящий из двух этапов: применение миварной экспертной системы на первом этапе, которая создает ограничения на пространство возможных решений, и эвристических методов оптимизации, например, алгоритмов генетического поиска на втором. Это связано с тем, что миварный подход создания экспертных систем предполагает использовать как продукционные правила, так и процедуры, функции, сервисы, которые могут быть, в том числе, реализациями методов ИИ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Джарратано и др., 2007] Экспертные системы: принципы разработки и программирование, 4е издание/ Джарратано Дж. [и др.]; – М.: Вильямс, 2007. 1152 с.
- [Варламов, 2002] Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство/ Варламов О.О.; – М.: Радио и связь, 2002. 282 с.
- [Санду, 2010] Миварный подход к созданию интеллектуальных систем и искусственного интеллекта. Результаты 25 лет развития и ближайшие перспективы/ Санду Р.А., Варламов О.О.; – М.: Стандартинформ, 2010. 339 с.
- [Владимиров и др., 2010] Программный комплекс “УДАВ”: практическая реализация активного обучаемого логического ввода с линейной вычислительной сложностью на основе миварной сети правил/ Владимиров А.Н. и [др.]/Труды научно-исследовательского института радио. - 2010.- №.1. С. 108-116.
- [Davis, 1991] Handbook of Genetic Algorithms/ ed. by Lawrence Davis; - New York: Van Nostrand Reinhold, 1991. 412 p.
- [Курейчик, 2002] Генетические алгоритмы и их применение/ Курейчик В.М.; - Таганрог: ТРТУ, 2002. 242 с.

TECHNIQUE OF THE FREQUENCY-SPATIAL PROBLEM SOLVING BASED ON USAGE OF PRODUCTIONAL SYSTEMS AND HEURISTIC ALGORITHMS

Maiboroda Y.I.

*Moscow Institute of Physics and Technology
(university), Dolgoprudny, Russia*

juramaiboroda@gmail.com

This work shows that with the aid of the mivar method of expert systems development it is possible to create software programs which use different realizations of artificial intelligence methods as services for solving problems in complicated subject areas. This paper presents a method of frequency-spatial planning problem solving in communication and broadcast networks: firstly, the mivar method of inference from knowledge base creates constraints in the space of probable solutions, then optimization with the aid of a genetic search algorithm is used, which realization is integrated into an expert system as a service.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89

АНАЛИЗ МЕДИЦИНСКОЙ ИНТЕЛЛЕКТУАЛЬНОЙ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ С ТОЧКИ ЗРЕНИЯ ВОЗМОЖНОСТЕЙ ЕЕ АВТОМАТИЗАЦИИ

Клещев А.С., Черняховская М.Ю., Шалфеева Е.А.

*Институт автоматизации и процессов управления,
г. Владивосток, Россия*

kleshev@iacp.dvo.ru

chernyah@iacp.dvo.ru

shalf@iacp.dvo.ru

В работе исследуется возможность единого подхода к автоматизации интеллектуальных процессов в отдельно взятой профессиональной области деятельности. Проводится системный анализ различных видов деятельности в медицине. Обсуждаются требования к онтологии всех информационных компонентов, используемых экспертными системами, средствами документирования, системами для обучения, для накопления знаний, интегрируемыми для единых целей.

Ключевые слова: знания, интеллектуальный процесс, профессиональная деятельность, онтологии информационных компонентов.

ВВЕДЕНИЕ

К настоящему времени достигнуты значительные успехи в автоматизации деятельности коммерческих предприятий, в разработке информационных и других систем для более эффективного управления бизнесом [Битек, 2007]. Однако до сих пор не создан инструментарий, позволяющий комплексно автоматизировать сферы экономики, процессы в которых связаны с интеллектуальной деятельностью и использованием постоянно обновляемых знаний. К таковым относятся многие области, финансируемые государством, - медицинское обслуживание, образование, наука, оборонная деятельность и др. К настоящему времени используются разрозненные подходы к автоматизации в таких сферах, как, например, медицина. При автоматизации лечебно-диагностических процессов отдельными средствами автоматизируют взаимодействие между участниками лечебно-диагностического процесса, с акцентом на документирование всех шагов этого взаимодействия. Лечебно-диагностический процесс рассматривают как бизнес-процесс, как технологию информационного взаимодействия, аналогичного любым другим управленческим процессам.

Задачи автоматизации постановки диагноза, назначения и прогнозирования лечения и т.п., если и решаются, то другими средствами. При этом есть проблемы с их полноценным использованием (часто приходится упрощать процессы), а также с их сопровождением. Однако наметилась тенденция к

стандартизации представления используемых документов: введено понятие *электронная история болезни* [ГОСТ, 2008].

К системам, разрабатываемым для автоматизации различных видов профессиональной интеллектуальной деятельности, финансируемых государством, предъявляются требования обеспечения всех специалистов адекватными средствами, повышающими их качество работы. Эти средства должны быть интегрированы друг с другом по управлению и по информации, должны быть адаптивными и управляемыми в силу влияния на деятельность специалистов внешних факторов (таких как изменения в законодательстве), и внутренних (непрерывного усовершенствования знаний).

Целью исследования является идентификация интеллектуальных видов деятельности в отдельно взятой достаточно сложной области профессиональной деятельности, системный анализ различных видов деятельности в медицине, разработка требований к онтологии всех информационных компонентов, необходимых для комплексной автоматизации.

1. Идентификация интеллектуальных видов деятельности (на примере медицины)

Особенность медицинской сферы состоит в особой ответственности врача за постановку

диагноза и назначение лечения пациенту. Поэтому медицина нуждается не в автоматизации принятия решений, а в автоматизации поддержки их принятия. Особую роль играют здесь консультации, к которым могут прибегать врачи, чтобы узнать аргументацию других специалистов при постановке диагноза или назначении лечебных мероприятий. Консультирование есть процесс поддержки принятия решения, результатом которого становится объяснение решения (т.е. анализ данных о состоянии пациента на основе экспертных знаний).

Виды профессиональных деятельностей, не затрагивающие обучение профессии и научную составляющую, обычно представляют тремя группами: *основные процессы, процессы управления, обеспечивающие процессы* [Битек, 2007].

Традиционно (в рамках автоматизации коммерческих предприятий медицинской сферы) к основным процессам относят те, исполнителями которых являются лечащий врач и регистратор [Битек, 2007, Каштальян, 2007].

При рассмотрении видов *интеллектуальной деятельности* (требующих принятия ответственных решений) целесообразно обратиться к известной классификации. При разработке систем, основанных на знаниях, принято выделять следующие классы интеллектуальных задач: *классификации, диагностики, интерпретации, мониторинга, управления, планирования, прогноза, ремонта, проектирования* [Уотермен, 1989]. Кроме того, сюда включают и такие задачи как: *поддержка принятия решения; обучение*.

К интеллектуальным (и тесно связанным с ними) задачам медицинской деятельности в рамках текущего исследования отнесены следующие.

К «основным, интеллектуальным» процессам относятся:

- *установление диагноза* пациента (это пример задачи диагностики);
- *назначение лечения* пациенту (это пример задачи ремонта);
- *опрос пациента* (это пример задачи получения недостающей информации – установить, значения каких признаков необходимы);
- *до-обследование* (это пример задачи получения недостающей информации с помощью лабораторных исследований – установить, значения каких признаков необходимы);
- *планирование дальнейших обследований и очередного посещения* врача (это пример задачи управления процессом обследования - за какими признаками надо наблюдать, и в какие моменты времени);
- *коррекция лечения*, которая включает в себя коррекцию назначения лечения, если прежняя схема лечения не дала прогнозируемых изменений

состояния, и коррекцию диагноза, если изменилось представление о состоянии пациента, (это пример задачи управления на основе результатов задачи мониторинга);

- *прогнозирование* результата лечения / развития болезни (это пример задачи прогноза) и

- *проведение инструментального обследования / функциональной диагностики* (это пример задачи интерпретации).

Автоматизация этих видов деятельности состоит в построении соответствующих экспертных систем. Программно-технические системы реального времени для проведения *инструментального обследования* остаются за рамками автоматизации интеллектуальной деятельности, однако вербальное представление их результатов врачами должно быть рассмотрено как часть автоматизируемого интеллектуального процесса – это вербально представление является входной информацией для других автоматизируемых видов деятельности.

К «обеспечивающим, интеллектуальным» процессам относятся:

- обучение проведению обследования (осмотра),
- обучение интерпретации результатов функциональной диагностики,
- обучение диагностике (и, возможно, необходимому для нее управлению до-обследованием),
- обучение назначению лечения.

Автоматизация деятельности состоит в построении тренажеров, проверяющих знания.

К числу «основных, неинтеллектуальных» процессов относится *ведение истории болезни* (это пример задачи документирования).

Задача выполнения лабораторного обследования (пример задачи интерпретации) рассматривается с точки зрения получения результатов в вербальном виде.

Автоматизация деятельности состоит в построении АРМов.

К числу научных (также «интеллектуальных») процессов относятся:

- формирование знаний о диагностике заболевания,
- формирование знаний о схеме лечения заболевания,
- формирование знаний о лекарствах и лечебных мероприятиях.

Автоматизация деятельности состоит в построении редакторов знаний и программ индуктивного формирования знаний [Клещев и др., 2011b] Использование результатов этих научных задач означает переход на качественно новый уровень в медицинской сфере, при этом

формируемые знания (вместе с их объяснением на конкретных фактах) должны проходить цикл критики и подтверждения в медицинском сообществе и только потом могут быть допущены к процессу обучения специалистов и к практике.

Сообщество создаваемых экспертных систем для консультирования и других вышеперечисленных программных средств обеспечивает систематическую поддержку принятия решений, направленных на выздоровление пациентов, и расширяет возможности обращения врачей к опыту и знаниям экспертов «от медицины».

2. Взаимосвязи подзадач в интеллектуальной профессиональной деятельности

Многие задачи **медицинской деятельности**, решаемые на протяжении периода взаимодействия пациента со специалистами медицинского учреждения, на сегодняшний день автоматизированы. Для них созданы поддерживающие принятие решений ЭС (или исследовательские прототипы) в соответствии с классами интеллектуальных задач (диагностики, ремонта и т.д.).

Однако автоматизация интеллектуальной профессиональной деятельности в комплексе требует интеграции поддержки принятия решений всех интеллектуальных и связанных с ними задач, а также задач обучения и формирование знаний.

Поддержка принятия решений специалистов-медиков на протяжении периода выздоровления пациента с точки зрения «автоматизации в комплексе» наиболее близка задачам теории управления сложными системами [Грибова 2010].

Если обратиться к терминологии, выработанной теорией управления, то *объектом управления* оказывается **пациент**, *целью управления* становится **отсутствие** (устранение) **заболевания**, она достигается за счет *управляющих воздействий* – **лечения** (которое пациенту назначено и которое он с большой вероятностью проведет). *Управление* – **выработка** этих **управляющих воздействий**, включающая мониторинг состояния (обследование и до-обследование), обработку информации (диагностика), принятие решений (назначение лечения).

Система управления состоит из управляющего объекта (медперсонала) и объекта управления (пациента). *Функцией управляющего объекта* является совокупность его действий (скорее, не-«однородная по некоторому признаку») – *сбор информации, установление диагноза пациента, назначение лечения, планирование, коррекция, прогнозирование лечения\развития, обследование, ведение истории болезни*, - подчиненных общей цели управления.

У пациента *протекают* процессы в организме, *проявлениями* которых являются значения

наблюдений признаков в различные моменты времени (они *составляют* дневник наблюдений – важную *часть* истории болезни). На основе значений наблюдений и собственных знаний о признаках заболеваний и их динамике врач ставит диагноз; далее на основе информации об особенностях пациента и собственных знаний о схеме лечения заболевания и знаний о лекарственных средствах и других лечебных мероприятиях готовит план лечебных мероприятий, которые должны воздействовать на больного – вести его к выздоровлению. В процессе выздоровления осуществляется слежение за изменением состояния пациента и, в случае отклонений от прогнозируемого изменения, могут быть скорректированы лечебные мероприятия или даже диагноз.

Примечание. Задачи обучения врачей и формирования новых знаний непосредственно к схеме управления состоянием пациента не относятся (скорее, к управлению медперсоналом). Но получаемые с их помощью знания являются важной информационной составляющей всего комплекса интеллектуальных профессиональных задач. Знания, приобретаемые в процессе обучения, являются основой правильных решений врачей. А наличие формализованных знаний повышает возможность каждого отдельного врача знать как можно больше (владеть более глубокими и/или широкими знаниями). Индуктивное формирование знаний приводит к новым знаниям о признаках заболеваний - по архивам историй болезни; к новым знаниям о схемах лечения - по протоколам корректировок планов лечебных мероприятий; к новым знаниям о лекарственных средствах\мероприятиях – по фрагментам дневников наблюдений от начала применения этих средств\мероприятий.

3. Структура и характеристики различных видов интеллектуальной деятельности (на примере медицинской)

Некоторые из видов деятельности могут рассматриваться как составные (состоящие из множества задач):

Основная деятельность = 1..*{задача},

где **задача = 1..*{[под]задача}**.

Любая интеллектуальная деятельность связана, как правило, с использованием существующих знаний и может рассматриваться как получение искомой\результатирующей информации на основе имеющейся входной. Особенность различных видов *интеллектуальной деятельности* медицинской сферы такова, что они подразумевают возможность, а иногда и необходимость консультирования, главным результатом которого является *объяснение (обоснование)* принимаемых решений. Это объяснение является дополнительной входной информацией для специалиста при принятии им

решения. Автоматизация консультирования связана с формализацией знаний (для выполняемой деятельности) и их использованием при построении объяснения.

Поэтому структура *основной интеллектуальной деятельности* (или каждой из ее подзадач) может быть представлена так, как на рис. 1.

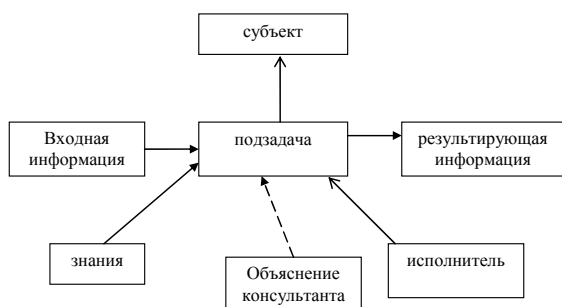


Рисунок 1 – Структура интеллектуальной деятельности.

Пунктирная стрелка означает, что «объяснение консультанта» может использоваться при принятии решения ответственным специалистом.

Для «основных» видов деятельности и подзадач медицинской сферы (*диагностика заболевания, назначение лечебных мероприятий, прогнозирование выздоровления, коррекция лечения*) основным субъектом является пациент, а информационным компонентом (входным и результирующим) является сложно устроенный документ *история болезни пациента*. Содержание этого документа расширяется в процессе выполнения различных подзадач медицинской деятельности.

Структура подзадачи «диагностика заболевания» до автоматизации такова (значком % помечены необязательные составляющие).

Субъект: **пациент;**

Входная информация:

история болезни (уточненная значениями лабораторных исследований);

Знания:

Персональные:

Теоретические знания о признаках заболеваний и их динамике;

%Собственная база прецедентов признаков заболеваний;

Общедоступные: методическая литература (книга, инструкция);

Исполнитель: **врач;**

%Объяснение консультанта: объяснение диагноза, выполненное консультантом (другим врачом, консилиумом);

Результирующая информация: **диагноз.**

При автоматизации этой деятельности

(разработке и внедрении справочных средств и диагностической ЭС) структура подзадачи «*диагностика заболевания пациента*» становится такой, как на рис. 2:

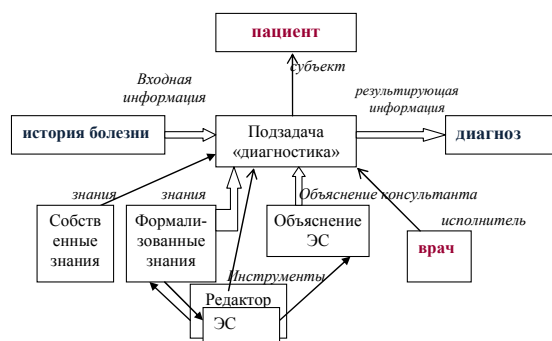


Рисунок 2 – Структура автоматизируемой интеллектуальной деятельности.

В том случае, когда разрабатывается «автоматизированный консультант» (ЭС) в помощь специалисту, «объяснение» формируется автоматически. Содержание некоторых элементов структуры подзадачи уточняется следующим образом.

Знания (общедоступные):

формализованные: Компьютерная база знаний о заболеваниях и норме;

текстовые: методическая литература.

Объяснение консультанта: **объяснение диагноза**, выполненное диагностической экспертной системой.

Инструмент или средство:

компьютерная база знаний о заболеваниях и норме,

диагностическая экспертная система.

Аналогично представляется структура подзадач *назначение лечебных мероприятий, прогнозирование выздоровления, коррекция лечения*, подобным образом - *до-обследование* и *выполнение функциональной диагностики*.

4 Требования к онтологии информационных компонентов деятельности

Как следует из анализа совокупности основных медицинских процессов, информационными компонентами различных видов интеллектуальной деятельности в медицине являются:

история болезни пациента,

формализованные знания о заболеваниях,

формализованные знания о лекарствах и других лечебных мероприятиях,

формализованные знания о схемах лечения заболеваний,

объяснение диагноза,
лечебные мероприятия (план лечения),
объяснение плана лечения,
объяснение коррекции лечения.

Комплексная автоматизация медицинской профессиональной деятельности требует разработки онтологии для каждого информационного компонента, поскольку все они используются во множестве решаемых задач.

«История болезни» должна включать в себя не только паспортную часть, особенности пациента, его жалобы на первичном приеме, результаты обследования врачом на первичном приеме, но и дневник наблюдений (жалобы и результаты обследования врачом на каждом следующем приеме), диагноз, а возможно еще и объяснение диагноза консультанта, план лечения, дневник лечения и план контрольных посещений врача, план дообследования.

Схематически *история болезни* может рассматриваться как:

паспортная часть +
особенности пациента +
жалобы +
дневник наблюдений +
диагноз

[+ дневник лечебных мероприятий].

Согласно национальному стандарту РФ «электронная история болезни» от 2008-01-01 [ГОСТ, 2008] каждая персональная медицинская запись включает в себя:

дату и время события (осмотра пациента, проведения манипуляции, забора биоматериала для анализа и др.),

автора данной медицинской записи,

медицинское содержание (результаты анализов или обследований, статус, эпикриз, назначение лекарств и т.д.) - текст или файл (медицинские изображения, графические материалы, тексты в различных форматах и т.д.) или построенные формализованные данные, позволяющие производить отбор и фильтрацию, проводить статистическую обработку, формировать отчеты.

Поэтому *дневник лечебных мероприятий* должен охватывать информацию о *назначенных лекарственных препаратах, о проведении манипуляций / процедур, а дневник наблюдений* – не только информацию об *инструментальных обследованиях (функциональной диагностике)*, но и о *результатах обследований узкими специалистами*.

Дневник наблюдений / лечения = * {запись}

Запись =

момент +
дата и время записи,
автор записи,
медицинское содержание.

Примеры области значений *медицинского содержания записи*: скалярное - *результат опроса (характер боли,...)*, *результат осмотра (цвет кожи,...)*, числовое значение - *результат объективного инструментального измерения (пульс, давление)*, таблица - *анализ крови*, изображение и текст - *результат функциональной диагностики (флюорография, УЗИ)*.

Для различных видов профессиональной деятельности (таких, как медицина), характерно использование единой терминологии при представлении данных и знаний. Явное представление ее в отдельном ресурсе, от которого зависят ресурсы-данные и ресурсы-знания, гарантирует и согласованность при сотрудничестве всех специалистов, и согласованность элементов данных, которыми оперирует прикладная логика программных компонентов, автоматизирующих или поддерживающих деятельность всех этих специалистов.

Онтология (структура) всех информационных ресурсов (документов) должна обеспечивать их эффективное использование экспертными системами, системами для обучения, средствами накопления знаний, средствами документирования, интегрируемыми для единых целей. Работа в этом направлении уже привела к получению *онтологии наблюдений* в медицине и *онтологии знаний о заболеваниях* [Клещев и др., 2011с, Клещев и др., 2011d]. Одним из языков, ориентированных на единообразное представление информационных ресурсов и их онтологий для создаваемых сообществ программных систем, является декларативный язык, разрабатываемый в ИАПУ ДВО РАН [Клещев и др., 2011а].

Онтологии вышеперечисленных информационных компонентов должны обеспечивать повторную используемость формируемых документов при создании сообществ экспертных систем, систем для обучения, средств накопления знаний и других инструментов при комплексной автоматизации медицинской профессиональной деятельности.

ЗАКЛЮЧЕНИЕ

Построенные модели интеллектуальной профессиональной деятельности и онтологии необходимых ей информационных ресурсов являются основой концепции комплексной автоматизации процессов в медицинской области, цель которой - систематическая поддержка принятия ответственных решений. Показано, что совокупность решаемых задач на протяжении периода достижения главной цели (выздоровления

пациента) близка теории управления сложными системами.

Модели видов деятельности и онтологии ресурсов дают основную схему для формирования методологии системного анализа и моделирования произвольных сфер деятельности с интеллектуальными процессами. В свою очередь методология позволяет построить единую онтологию профессиональной деятельности специалистов, на базе которой могут быть разработаны технология и инструментарий автоматизации интеллектуальной профессиональной деятельности.

Работа выполнена при поддержке грантов РФФИ № 10-07-00089-а и ДВО РАН № 12-III-A-01И-006.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Битек, 2007] «Процессно-организационная бизнес-модель стоматологической клиники, занимающейся лечением зубов, пародонта, зубопротезированием, имплантологией, исправлением прикуса и профилактикой.» / материалы открытого семинара-практикума «Совершенствование системы управления предприятием на основе описания и оптимизации бизнес-процессов», 5-7 ноября 2007 г., компания «БИТЕК», Москва // Информационный портал Betec.Ru. <http://www.betec.ru/secure/index.php?id=5&sid=14&tid=03>.

[ГОСТ, 2008] «Электронная история болезни. Общие положения» (). / ред. Гелемеева О.В. // ГОСТ Р 52636-2006. Дата введения 2008-01-01.

[Грибова, 2010] Грибова В.В., Клещев А.С., Шалфеева Е.А. Управление интеллектуальными системами // Известия РАН. Теории и системы управления. 2010. № 6. С. 122-137.

[Каштальян, 2007] Каштальян А.А. Анализ затрат труда врачей амбулаторно-поликлинического приема (по материалам хронометражного исследования) // Журнал «Медицинские новости», 2007, С. 71—74. - <http://www.mednovosti.by/journal.aspx?article=204>.

[Клещев и др., 2011а] Клещев А.С., Грибова В.В., Крылов Д.А. Контекстно-зависимые грамматики искусственных языков. - Владивосток: ИАПУ ДВО РАН, 2011. 30 с.

[Клещев и др., 2011б] Клещев А.С., Смагин С.В. Алгоритм формирования баз знаний по обучающим выборкам для онтологии медицинской диагностики, приближенной к реальной / Владивосток: ИАПУ ДВО РАН. - 2011. - 48 с. - <http://www.iacp.dvo.ru/is/publications/2011-Kleshev.Smagin-ExperThree.pdf>

[Клещев и др., 2011с] 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. - [http://www.iacp.dvo.ru/is/publications/Article1-Moskalenko-IACP\(NTI-2005\).rtf](http://www.iacp.dvo.ru/is/publications/Article1-Moskalenko-IACP(NTI-2005).rtf)

[Клещев и др., 2011д] 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.

[Уотермен, 1989] Уотермен Д. Руководство по экспертным системам: Пер. с англ. под ред. В. Л. Стефанюка. — М.: «Мир», 1989: — 388 с.

THE ANALYSIS OF MEDICAL INTELLECTUAL PROFESSIONAL ACTIVITY FROM THE VIEWPOINT OF AUTOMATION

Kleschev A. , Chernyahovskaya M., Shalfeeva E.

The Institute of Automation and Control Processes, Vladivostok, Russia

kleschev@iacp.dvo.ru

chernyah@iacp.dvo.ru

shalf@iacp.dvo.ru

The possibility of the uniform approach to automation of intellectual processes in some professional sphere is investigated in this work. The system analysis of various activities in medicine is carried out. The requirements to ontology of all information components used by expert systems, documentation programs, coursewares, knowledge-mining systems, integrated for the common purposes, are offered.

INTRODUCTION

Considerable successes in automation of commercial enterprises, in development of information systems for more efficient control business are reached by now. However there are no toolkits for a complex automation in professional spheres with intellectual activities and constantly updated knowledge.

The goal of this position paper is identification of intellectual activities in some complex area of professional work (in medicine), the system analysis of various activities, forming of requirements to ontology of all necessary information components.

MAIN PART (REPORT THESES)

The examples of intellectual medical activities (within current research) are that: diagnostics of the patient, appointment of treatment to the patient, additional inspection, planning of control inspections and the next visiting of the doctor, prediction of result of treatment evolution, functional diagnostics.

Automation of these activities consists in construction corresponding expert systems. Commonly expert systems in medicine are used in decision support: reminding information or options to an experienced decision maker.

Complex automation of intellectual professional activity demands the integration of supporting of all intellectual decision-making tasks and also training and formation of knowledge.

CONCLUSION

The intellectual professional activity models and ontology of necessary information resources are a basis of the medical processes complex automation conception which purpose is to systemetically support all of the critical decision making. It is shown that set of tasks solved during convalescence of the patient is close to the control theory of complex systems.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 510.63

МЕДИЦИНСКАЯ ДИАГНОСТИКА И ОРТОГОНАЛЬНЫЙ БАЗИС СИЛЛОГИСТИКИ

Сметанин Ю.М., Сметанин М.Ю.*

ГБОУ ВПО «Удмуртский государственный университет»,
г. Ижевск, Россия

gms1234gms@rambler.ru

* ГУЗ Диагностический центр УР, г.Ижевск, Россия

Misha1977@rambler.ru

В работе рассматривается одно из возможных приложений новой логической модели рассуждений на базе невырожденной булевой алгебры множеств. Предлагаемые методы поиска логических выводов из заданной системы посылок изначально ориентированы на применение компьютерной технологии решения и наглядное отображение логических соотношений между терминами задачи диагностики.

Ключевые слова: алгебраическая система, исчисление конститuentных множеств, медицинская диагностика, силлогистика.

1. Введение

Наблюдение, оценка наблюдаемых явлений и умозаключение - таковы три обязательных этапа на пути к распознаванию болезней и диагнозу. Соответственно этим трем этапам все содержание диагностики можно разделить на три до известной степени обособленных раздела:

- 1) раздел, охватывающий методы наблюдения или исследования, - врачебная методика или диагностика в узком смысле слова;
- 2) раздел, посвященный изучению симптомов, обнаруживаемых исследованием, - семиология или семиотика;
- 3) раздел, в котором выясняются особенности мышления при построении диагностических заключений на данных наблюдения, - врачебная или клиническая логика.

Первые два достаточно подробно разработаны и составляют главное содержание всех руководств и курсов по диагностике. Третий же раздел - врачебная логика - совершенно недостаточно подробно теоретически и практически разработан: обычно в учебниках, в главах, посвященных частной диагностике отдельных заболеваний, можно найти только простые сопоставления или перечисления симптомов, только внешние вехи врачебной логики в виде пожеланий использовать основные логические законы и отношения между суждениями. Усвоение процедур проведения логических рассуждений - этой существенной и необходимой стороны дела происходит в клинике, в самом процессе врачебной

деятельности и процесс этот во многом носит стихийный характер и слабоуправляем со стороны преподавателя. В сложившемся положении дел «виновата» не только медицина, но логика, которая до сих пор не может дать естественный для специалиста аппарат анализа рассуждений подобный силлогистике Аристотеля, которую логики до сих пор не могут органично встроить в математическую логику. Сама же эта силлогистика, являющаяся основой западного мировоззрения, лукава (многосмысловая) в самой своей сути. В работе указывается один из возможных путей выхода из данной ситуации, посредством принятия односмыслового базиса силлогистики взамен аристотелевского (*AXY, EXY, IXY, OXY*) [Сметанин 2009, 2010]. Предлагается практичный и наглядный аппарат для разработки приемов мышления и верификации гипотез, характерных для клинической логики. В статье анализируется актуальная проблема логики: интерпретация аристотелевской силлогистики в современных терминах. Количество работ в мировой науке, посвященных этому вопросу превышает 500, что указывает на сложность проблемы. Для ее решения предложен целый ряд расходящихся решений, предложенных самыми авторитетными логиками. В работе предложена точная и непротиворечивая модель полисиллогистики. Предлагаемый подход к решению полисиллогизмов является новым, на его основе построена программа для анализа силлогизмов и полисиллогизмов.

2. Способы описания состояния пациента и схема постановки диагноза.

2.1. Применяемые шкалы измерений

Для выявления ситуации, в которой пациент находится и находится и, предположительно, будет находиться, необходимо собрать и обработать объективную и субъективную информацию о состоянии его здоровья. Эта информация может быть получена в результате обследования состоящего из процедур: простого наблюдения, анамнеза, медицинских измерений, лабораторных исследований. Тип полученной в результате информации о пациенте, способ ее записи и возможные способы оперирования с ней зависят в основном от типа измерительной шкалы. Теория измерений оперирует понятием «эмпирическая система с отношениями» E , которая включает в себя множество измеряемых объектов A и набор интересующих исследователя отношений между этими объектами R ; $E = \{A, R\}$. Например, множество A — это множество физических тел пациентов, а набор R — отношения между ними по весу, росту, температуре, давлению и т. п. Для записи результатов наблюдений используется символьная система с отношениями N , состоящая из множества символов M , например множества всех действительных чисел, и конечного набора отношений P на этих символах: $N = \{M, P\}$.

Отношения P выбираются так, чтобы ими было удобно отображать наблюдаемые эмпирические отношения R . Если тело t тяжелее тела q , т. е. если имеет место отношение $R(t > q)$, то цифровая запись веса тел $t = 56$ и $q = 54$ позволяет наглядно увидеть это эмпирическое событие в записи $P(56 > 54)$. Договоренность использовать именно такое отображение системы E на систему N означает выбор некоторого определенного правила отображения g . Тройка элементов (E, N, g) называется шкалой измерений.

Шкалы классифицируются относительно возможных преобразований измерений в них. Пусть g_1 и g_2 различные отображения отношений между весами пациентов, выражающие веса в фунтах и килограммах. Тогда между результатами измерений в одной шкале и другой существует линейная зависимость $g_1(a) = k \cdot g_2(a), k > 0$. Шкалы, допускающие линейные преобразования относятся к типу рациональных шкал. Шкалы, переходы между которыми описываются линейными

$g_1(a) = k \cdot g_2(a) + b, k > 0$ преобразованиями, называются интервальными шкалами. Типичным примером интервальной шкалы является температурная шкала. Мы можем измерять температуру по шкале Цельсия или Фаренгейта. Переход от одной шкалы к другой осуществим при помощи линейного преобразования $C(x) = (5/9)F(x) - 160/9$. Линейное преобразование, в котором коэффициент $k=1$, называется сдвигом, а сама шкала называется шкалой разности. В медицине часто используются, так называемые, классификационные или перечислительные шкалы. По существу классификационная шкала есть не что иное, как

деление класса U на взаимно непересекающиеся и непустые подклассы, объединение которых равно универсуму. Классификационная шкала описывает операцию деления с точностью до наименования классов деления. Это означает, что мы можем классам деления приписывать какие угодно имена, но обязаны при этом соблюдать правило: одинаковые классы получают одинаковые имена, разные классы — разные. Как отмечено в работе [Смирнов, 2007] классификационные шкалы имеют принципиальное значение для применения силлогистических и полисиллогистических выводов в процессе диагностики. Цитируем:

«С некоторым огрублением любые шкалы можно свести к классификационным. Область значения функции f (т.е. N) мы можем разбить на взаимно непересекающиеся непустые классы так, чтобы их объединение совпадало с универсумом. Тогда информацию, что значение $f(x)=n$, мы огрубляем и полагаем, что значение $f(x)$ принадлежит классу, к которому принадлежит n . Например, мы имеем непрерывную интервальную температурную шкалу, принимающую значение от 33 до 42 градусов Цельсия. Область значений температуры мы можем разбить на классы:

резкая гипотермия	-от 34 до 35	включительно
гипотермия	-от 35 до 36	включительно
нормальная	-от 36 до 37	включительно
субфебрильная	-от 37 до 38	включительно
фебрильная	-от 38 до 40	включительно
предельная	-от 40 до 42	включительно.

Конечно, переход от интервальной (непрерывной) температурной шкалы к классификационной шкале означает потерю информации. Но этот переход дает нам возможность более просто производить логические операции. Отметим, что сделанная процедура сохраняет порядок между температурными классами. В общем же случае при переходе к классификационной шкале теряется и сам порядок.» конец цитаты.

2.2. Этапы диагностического процесса

Врачи со времен Гипократа понимают, что одни свойства организма непосредственно наблюдаемы, а другие нет. Однако на основании установленных связей между наблюдаемыми и ненаблюдаемыми свойствами посредством правильных рассуждений врач может сделать заключение о ненаблюдаемых свойствах, зная наблюдаемые свойства. Таким образом, в медицине мы имеем два типа знаний: данные о конкретном пациенте, полученные в результате наблюдения, и знания о связях между свойствами, полученные в результате индуктивных обобщений опыта и достижений теоретической медицины. Некоторые характеристики пациента свидетельствуют о его типических чертах (пол, возраст, тип конституции), другие — о нормальном или ненормальном состоянии его организма. Поэтому нам надо знать не только значения той или иной характеристики, но и знать, отклоняются или нет эти значения от нормы. Некоторые отклонения от нормы квалифициру-

ются как болезни. Болезнь проявляется через наблюдаемые свойства. Свойства, свидетельствующие о той или иной болезни, называются симптомами этой болезни.

В процессе клинической диагностики [Постовит 1991] выделяет две фазы и три этапа. Фазу анализа и дифференциации и фазу интеграции и синтеза, которые во времени протекают в трех периодах. Период сбора сведений – выявление всех симптомов заболеваний, период анализа и дифференциации – осмысление обнаруженных симптомов «сортировка» их по степени важности и характерности, период интеграции и синтеза – формулирование диагноза и его верификация. Три этапа разворачиваются во времени в форме циклов диагностической деятельности.

2.3. Используемая логическая модель.

Применяя логически обоснованные методы рассуждений к собранному данным и имеющимся знаниям, а также используя интуицию, врач выдвигает гипотезы и верифицирует их, осуществляя в конечном итоге постановку диагноза. Отличие от формальной логики здесь в том, что наличие симптома не обязательно свидетельствует о наличии, связанной с ним болезни и отсутствие симптома не позволяет утверждать об отсутствии, связанной с ним болезни. Предлагаемая в работе логическая модель описана Смирновым В.А. [Смирнов 2007], однако ее применение отлично от общепринятого в логике.

Пусть имеются знания о зависимостях между болезнями и симптомами, а также о наличии комбинаций симптомов, которые выражены суждениями в односмысловом ортогональном базисе силлогистики предложенным Сметаниным Ю.М. [Сметанин 2009]. Например, вербальные знания о связи болезнями D_i и S_k имеют вид четырех суждений ортогонального базиса силлогистики [Сметанин 2009, 2010] смотри (1):

$$\begin{aligned} &A(D_1, S_1 S_2 S_3 S_4), \\ &A(D_2, S_1 S_2 S_3 S_4), \\ &A(D_3, S_1 S_2 S_3 S_4) \\ &Eq(S_1 S_2 S_3, U) \end{aligned} \quad (1)$$

Первые три означают, что болезни D_1, D_2, D_3 являются причинами комплекса симптомов $S_1 \& \neg S_2 \& S_3 \& \neg S_4, S_1 \& S_2 \& \neg S_3 \& S_4$ и $S_1 \& S_2 \& \neg S_3 \& \neg S_4$ соответственно. Здесь через $\&$ и \neg обозначены логические операции «конъюнкция» и «отрицание». Кроме того у пациента выявлен комплекс симптомов $S_1 \& S_2 \& \neg S_3 \& \neg S_4$. необходимо логически обосновать выводы, которые можно сделать относительно наличия (отсутствия) болезнями D_1, D_2, D_3 у данного пациента. Тут математическая логика предлагает высказать предположение, а потом его доказать с помощью логического вывода, что безусловно не приемлемо для практикующего врача. В работе [Сметанин 2010] теоретические и компьютерные

средства интерпретации комплекса суждений в алгебре множеств, которые позволяют наглядно представить их логические связи в алгебраической системе носителем которой является система множеств, сопоставленных терминам $D_1, D_2, D_3, S_1, S_2, S_3, S_4$.

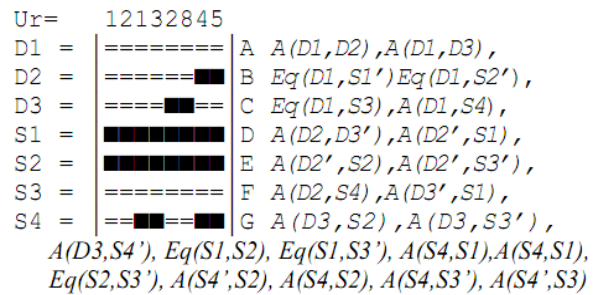


Рис. 1 Наглядная основа для логического анализа

Также можно проверить, в том числе и визуально любые гипотезы относительно наличия (отсутствия) конкретной болезни, если это возможно на данном уровне развития медицинской науки и практики.

Интерпретация изображена на машинограмме рис. 1 определения смысла обозначений комплекса условий (1) даны в разделе 3.

На Рис. 1 показана линейная диаграмма [Лобанов 2009] на которой универсумом является множество порожаемое непустыми конститuentами множеств $D_1, D_2, D_3, S_1 \& S_2 \& \neg S_3 \& \neg S_4$. Скажем другими словами понятными врачу. Каждая точка на горизонтальной оси ограниченной вертикальными линиями представляет собой пациента из множества пациентов, удовлетворяющих логическим условиям задачи постановки диагноза, то есть это пациент, который обладает комплексом симптомов $S_1 \& S_2 \& \neg S_3 \& \neg S_4$ и состояние которого мы пытаемся определить, обладая запасом знаний о связи болезни с симптомами. А именно, «болезни D_1, D_2, D_3 являются причинами комплекса симптомов $S_1 \& \neg S_2 \& S_3 \& \neg S_4, S_1 \& S_2 \& \neg S_3 \& S_4$ и $S_1 \& S_2 \& \neg S_3 \& \neg S_4$ соответственно».

Мы легко можем вывести следствие о том, что пациент (точка- элемент универсума) не может быть болен болезнью D1, эта болезнь изображена на диаграмме в виде пустого множества знаками (====), что означает, что при данном сочетании комплекса логических условий, описывающих состояние пациента и знания о связи болезней и симптомов, болезнь D1 можно не рассматривать как кандидат для диагноза. Но мы не можем этого сказать о болезнях D2 и D3, некоторые пациенты из универсума с данными свойствами могут ими страдать, а могут и не страдать. Единственно, что мы определенно можем добавить к выводу относительно D1 это то, что любой пациент из универсума, обладающего свойствами (1) не может страдать болезнями D2 и D3 одновременно. Он может страдать от D2 либо от D3 либо не страдать ни от одной из них. Если мы обладаем дополнительной информацией, что пациент болен одной из трех перечисленных болезней, то можем вывести, что пациент болен, либо болезнью D2, либо болезнью D3. Данное дополнительное предположение $Eq(D_2 + D_3, U)$ легко добавить к сис-

теме суждений (1) и новая диаграмма, которая его учитывает, изображена на рис. 2. Она не оставляет сомнений в том, что пациент болен болезнью D2 либо D3, но не той и другой одновременно.

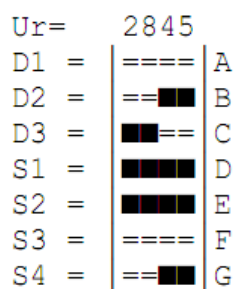


Рис.2 Окончательный диагноз при дополненном комплексе условий.

Полученное решение полностью совпадает с решением, приведенным в работе [Смирнов 2007], однако теперь его может получить и рядовой врач, а не только квалифицированный логик. При этом предлагаемый компьютерный метод решения полисиллогизмов одинаково легко решает значительно более сложные задачи.

Рассмотрим вероятностный подход к диагностике - учет вероятностных связей. Мы предполагали, что наши общие знания носят достоверный характер, то есть связь между болезнью и симптомом детерминирована. Однако в реальности она в большинстве случаев является стохастической. Цитируем [Смирнов 2007]

«Чаще мы имеем знания не вида "Если имеется болезнь D_i , то имеет место S_j ", а скорее вида "Если имеется болезнь D_i , то в 97% имеет место S_j ". Использование такого рода знаний позволяет нам дедуктивно выводить следствия из этих знаний и данных о пациенте только с определенной степенью вероятности. Отметим, что сама процедура рассуждений строго дедуктивна (а не правдоподобна). Но мы выводим в качестве следствий не утверждения типа "Имеет место болезнь D_i ", а утверждения "Имеет место болезнь D_i с такой-то вероятностью".

Вместо утверждения, что болезнью D_i болели три процента пациентов, мы говорим, что для произвольно взятого пациента вероятность того, что он болен болезнью D_i равна 0.03. Это утверждение мы записываем в виде $P(D_i)=0.03$. Тогда утверждение "Если имеется болезнь D_i , то будет наблюдаться набор симптомов S в 97% случаев" запишется в виде условной вероятности $P(S/D_i)=0.97$ (вероятность того, что при наличии болезни D_i будет иметь место набор симптомов S , равна 0.97).

Из теории вероятностей известно, каким образом вычислить условную вероятность $P(S/D_i)$ по вероятностям $P(S \& D_i)$ и $P(D_i)$. Последние две вероятности легко вычисляются на основании анализа статистических данных. Условная вероятность вычисляется по следующей формуле (2)

$P(S / D_i) = \frac{P(S \cdot D_i)}{P(D_i)}$	(2)
--	-----

Теперь мы можем использовать не только детерминистические связи между болезнями и симптомами, но и вероятностные связи. Задача вероятностной диагностики формулируется следующим образом: как по вероятности симптома относительно болезни установить вероятность болезни относительно симптома.

Пусть у нас имеется система связей между наборами симптомов и болезнями, имеющая вид $P(S^k/D_i)$, где k - это номер конъюнкции симптомов ($k=1, \dots, m$) и i - номер болезни ($i=1, \dots, n$). Теперь мы можем вычислить вероятность болезни относительно фиксированного набора свойств S^l , которые присущи исследуемому пациенту. То есть нам надо найти $P(D_i / S^l)$. А это осуществляется по известной формуле Байеса:

$P(D_i / S^l) = \frac{P(D_i) \cdot P(S^l / D_i)}{\sum_{j=1}^n P(D_j) \cdot P(S^l / D_j)}$	(2)
---	-----

Самый простой метод уменьшения числа рассматриваемых симптомов состоит в объединении некоторых из них в симптомокомплекс и рассмотрение этого симптомокомплекса как отдельного симптома. Например, слабость, сонливость, вялость, снижение работоспособности составляет малый симптом.

Однако, эта процедура не столь часта и не снижает значительно число подлежащих рассмотрению факторов, доступных наблюдению. Более существенным является понятие синдрома, которое подробно анализировалось в главе 6. А.С. Мелентьев резко возражает против отождествления понятий синдрома и симптомокомплекса. С логико-методологической точки зрения синдром есть новый фактор, не обязательно доступный непосредственному измерению. С синдромом связан некоторый симптомокомплекс, точнее, дизъюнкция симптомокомплексов. Наше знание о связи синдрома с симптомокомплексом выражается следующей импликацией:

$$C \rightarrow \varphi(A_1, \dots, A_n),$$

где C есть синдром, A_1, \dots, A_n - факторы, определенные значения которых рассматриваются как симптомы и φ - функция, устанавливающая взаимосвязь симптомов.

Синдром связывается также с патогенезом, типом патологического процесса (воспаление, опухоль, интоксикация и т.д.). Наконец, синдром связан с нозологиями или синдромами более высокого уровня (H)» конец цитаты из [Смирнов 2007]

3. Ортогональный базис как методологическая основа правильных рассуждений в предметных областях деятельности.

3.1. Роковые ля логики особенности базиса Аристотеля

Булева алгебра [Владимиров 1969] на основе множеств имеет непосредственное отношение к силлогистике, которое было замечено и исследовано Лейбницем, Жергонном, Венном, Эйлером и другими корифеями. В основе силлогистики Аристотеля лежат простые суждения, представленные четырьмя типами: **A** – общеутвердительное (все X есть Y); **E** – общеотрицательное (все X не есть Y); **I** – частноутвердительное (некоторые X есть Y); **O** – частноотрицательное (некоторые X не есть Y). Жергону рис.3 удалось представить все классы Аристотелевых простых суждений, с помощью соотношений между множествами. Соотношения $G_9, G_{11}, G_{13}, G_{14}, G_{15}$ получили в математике и логике название «жергонновых отношений». Их расширенный состав с учетом универсума показан на рис.3.

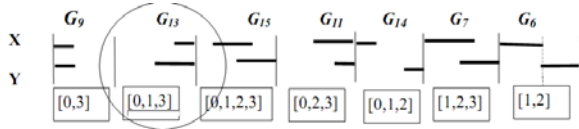


Рис. 3 Расширенные за счет G_6 и G_7 жергонновы отношения.

Кулик Б.А. [Кулик 1997] проиллюстрировал с помощью пяти классических жергонновых отношений многосмысловость базиса Аристотеля. Однако смысл простых суждений еще больше если рассматривать расширенные жергонновы отношения смотри рис. 4. Ортогональный базис силлогистики

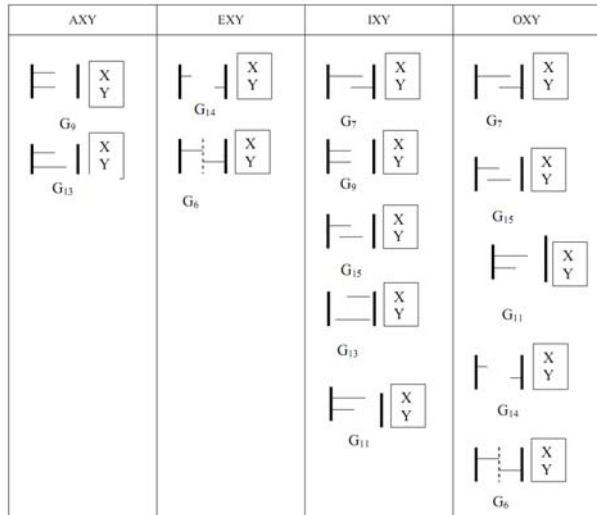


Рис. 4 Многосмысловость простых суждений из базиса Аристотеля

На рисунке 3 овалом обведено соотношение являющееся прообразом материальной импликации в формальной математической логике и элементом ортогонального базиса (ОБ) силлогистики предложенного Сметаниным Ю.М. [Сметанин 2009, 2010].

Пусть X и Y не пустые, собственные подмножества универсума. Тогда суждения 1 - 3 называются ортогональным базисом силлогистики (ОБ).

1. $A(X,Y)$ – Все X суть Y в смысле $X \subset Y$ (общеутвердительное суждение по другому - левостороннее включение);

2. $Eq(X,Y)$ – множество X совпадает с множеством Y по другому равносильность ;

3. $IO(X,Y)$ –(некоторые X есть Y) и (некоторые X не есть Y) и (X объединенное с Y не есть универсум) по другому независимое пересечение;

Покажем, как через суждения ОБ можно выразить расширенные жергонновы отношения (рис. 2).

1. $G_9(X,Y) = Eq(X,Y); U = XY + X'Y'$ – множество X совпадает (равно) с множеством Y ;

2. $G_{13}(X,Y) = A(X,Y)$ - множество X есть собственное подмножество Y или событие X влечет наступление события Y либо $U = XY + X'Y + X'Y'$. Другими словами все X есть Y тогда и только тогда, когда произвольный элемент универсума e удовлетворяет только одному из соотношений $e \notin X$ и $e \notin Y$, либо $e \notin X$ и $e \in Y$, либо $e \in X$ и $e \in Y$;

3. $G_{15}(X,Y) = IO(X,Y)$ либо $U = XY + XY' + X'Y + X'Y'$ существуют разбиение универсума на 4 непустые подмножества $XY, XY', X'Y, X'Y'$. По другому можно сказать что (некоторые X есть Y) и (некоторые X не есть Y) и (некоторые Y не есть X) и (некоторые не X не есть Y);

4. $G_{11} = A(X',Y')$ либо $U = XY + XY' + X'Y'$ - правостороннее включение;

5. $G_{14} = A(X,Y')$ все X не есть Y либо $U = XY' + X'Y + X'Y'$ - неполная несовместимость;

6. $G_7 = A(X',Y)$ либо $U = XY' + XY + X'Y'$ – зависимое пересечение;

7. $G_6 = Eq(X,Y')$ либо $U = XY' + X'Y$ – полная несовместимость.

Все семь расширенных жергонновых отношения выражаются одним из простых суждений ($A(X,Y), Eq(X,Y), IO(X,Y)$) и поэтому в качестве базиса можно использовать только три функтора (простых суждения о множествах X, Y) это $A(X,Y), Eq(X,Y), IO(X,Y)$.

С помощью введенного базиса ликвидируется многосмысловость базиса Аристотеля для расширенных жергонновых отношений как отношений между непустыми подмножествами универсума. Ортогональный базис можно применять как совокупности суждений о равенстве (неравенстве) пустому множеству некоторых конституентных множеств системы порождающих множеств смотри теоремы из работы [Сметанин 2010]:

$$1. G_3 : A(X,Y) \equiv X \subset Y \equiv (XY' = \emptyset) (X'Y \neq \emptyset)$$

$$2. G_5 : Eq(X,Y) \equiv X = Y \equiv (XY' = \emptyset) (X'Y = \emptyset)$$

$$3. G_5 : IO(X,Y) \equiv (X'Y' \neq \emptyset) (X'Y \neq \emptyset) (XY' \neq \emptyset) (XY \neq \emptyset)$$

Рассматривая жергонновы отношения в форме отношений между непустыми множествами, выраженные в форме равенства (неравенства) пустому множеству (либо его изнанке – универсуму) в работах [Сметанин 2009, 2010] удалось получить более адекватную по сравнению с моделью классической

логики модель в форме невырожденной булевой алгебры на основе множеств [Владимиров 1969]. В данной модели вместо высказывательных переменных и логических функций от них рассматриваются отношения между множествами прообразами этих высказывательных переменных в форме утверждений о равенстве (неравенстве) пустому множеству конститuent сравниваемых множеств.

3.2. Невырожденная булева алгебра на основе множеств – наиболее приемлемая модель для классической логики.

Рассмотрим критические недостатки классической модели, являющейся вырожденной булевой алгеброй [Владимиров 1969, стр. 25].

Во первых - не различение в AXU отношения строгого и нестрогого включения множеств сопоставляет отношениям $X \subset Y$ и $X \subseteq Y$ (в случае когда $(X \neq \emptyset)(Y \neq \emptyset)(X \subset U)(Y \subset U)$) одну и ту же операцию материальной импликации $x \Rightarrow y$. Обратим здесь внимание уважаемого читателя на очень важный момент касающийся высказывательных булевых переменных x и y . Это, отнюдь не какие-то произвольные неизвестно откуда взявшиеся переменные. Они непосредственно связаны с объективной реальностью, в которой имеются множество всех рассматриваемых предметов и, два непустые подмножества, всех предметов, которые имеют имена X и Y и находятся между собой в отношении ($X \subset Y$ либо $X \subseteq Y$). Таким образом, эти переменные x и y мы нашли не на улице, они являются характеристическими функциями множеств X и Y и их значение (истина или ложь) имеют не абстрактное как у Гильберта значение. Итак, характеристическая функция (индикатор) множества X ставит в соответствие любому элементу универсума $e \in U$, булеву переменную x , которая равна 1, если e принадлежит множеству X либо равна 0, если e не принадлежит X . Таким образом, смысл «истины» и «лжи» проявляется только на фоне инвариантной неопределенности – универсума. [Вальков 1985]. Соотношению $X \subseteq Y$ соответствует таблица 1, задающая соотношения между индикаторами этих множеств.

Таблица 1 - истинности для соотношения нестрогого включения.

x	y	
$e \in X$	$e \in Y$	$x \rightarrow y$
0	0	1
0	1	i
1	0	0
1	1	1

Этой таблице соответствует формула трехзначной логики $xy + x'y' + ix'y$, равносильная утверждению $(x < y) \oplus (x = y)$. Здесь через i обозначено – третье истинностное значение «может быть», то есть это не материальная импликация.

Соотношению $X \subset Y$ соответствует таблица 2, задающая отношение между индикаторами X и Y .

Таблица 2 - истинности для соотношения строгого включения.

	x	y	
Нп п	$e \in X$	$e \in Y$	$x \Rightarrow y$
1	0	0	1
2	0	1	1
3	1	0	0
4	1	1	1

Этой таблице соответствует формула $x'y' + x'y + xy$ - материальная импликация.

Во вторых трактовка строгого и нестрогого включения как импликации, а самой импликации как булевой функции, для случая вырожденных жергонновых отношений, напрямую приводит к парадоксам материальной импликации. Это показано в работе [Сметанин 2011].

В работе [Владимиров 1996] рассматриваются булевы алгебры - алгебраические системы, которые в зависимости от обстоятельств могут интерпретироваться как системы событий, либо как системы высказываний, допуская и другие интерпретации. В частности там отмечено [Владимиров 1996 с.8]

« Буль в своей обширной монографии «Исследование законов мышления, на которых основаны математические теории логики и вероятностей» отчетливо указал на связь построенного им исчисления с основаниями теории вероятностей. Эта связь основывается на аналогии между «событиями» и «высказываниями», позволяющей обслуживать логику и теорию вероятностей одним формальным аппаратом. Грубо говоря, «событие» — это то, что может произойти или не произойти; «высказывание» же — это то, что может быть истинно или ложно. Среди событий есть достоверные и невозможные; высказывания могут оказаться тождественно истинными или тождественно ложными. Между событиями возможна причинно-следственная связь: одно событие бывает иногда следствием другого. Точно так же между высказываниями возможна логическая связь; они могут вытекать одно из другого. Каждому событию может быть сопоставлено некоторое высказывание, утверждающее, что это событие произошло. С другой стороны, всегда можно истолковать высказывание как утверждение об осуществлении некоторого события. Сказанное сейчас убеждает в возможности построения единого «исчисления», которое могло бы, смотря по обстоятельствам, служить то «исчислением высказываний», то «исчислением событий». Такое исчисление и было создано Дж. Булем. В течение полувека, однако, оно развивалось в чисто «логическом» русле. Первое значительное исследование по аксиома-

тике теории вероятностей появилось лишь в 1917 г.; его автором был С. Н. Бернштейн). Последующие исследования в этой области, связанные в первую очередь с работами А. Н. Колмогорова), окончательно поставили теорию вероятностей на твердую почву и оказали большое влияние на смежные разделы математики, в особенности — на теорию меры». Конец цитаты.

Поскольку событие имеет интерпретацию в форме множества, а высказывание интерпретируется как пропозициональная переменная с двумя возможными значениями “истина” и ”ложь”, то изоморфизм между алгеброй событий и алгеброй их индикаторов возможен только в случае если множества - события в универсуме (достоверном событии), упорядочены отношением нестрогого порядка, покажем это. Пусть Σ и S два частично упорядоченных (отношением строгого порядка $<$) множества. Будем говорить, что отображение множества Σ в множество S есть строгий изоморфизм, (антирефлексивный изоморфизм) если оно взаимно однозначно и сохраняет строгий порядок ($<$), то есть неравенства $x < y$, $x, y \in \Sigma$ и

$\varphi(x) < \varphi(y)$ $\varphi(x), \varphi(y) \in S$ равносильны. Ясно,

что обратное отображение φ^{-1} есть также строгий

изоморфизм. Множества Σ и S в случае надобности можно отождествлять. Для обозначения таких

множеств можно использовать отношение равенства ($=$).

Пусть характеристическая функция, или индикатор

множества X , имеющая область определения элементы универсума, определяется, равенством

$x(e) = 1$, если $e \in X$ и $x(e) = 0$, если $e \notin X$,

где e произвольный элемент универсума. Рассмотрим систему $\Sigma = \{X_i\}$ произвольных собственных

подмножеств непустого множества U . Пусть на множествах из Σ определен нестрогий частичный

порядок посредством отношения \subseteq нестрогого

включения. Возьмем в качестве S систему всех характеристических функций множеств из Σ .

Обозначим через φ отображение, сопоставляющее

каждому X_i его характеристическую функцию

x_i . Ясно, что φ устанавливает взаимно однозначное

соответствие между Σ и S . Нестрогое включение

$X_i \subseteq X_j$ означает, что $x_i \leq x_j$ для любого

$e \in U$ поэтому неравенства $X_i \subseteq X_j$ и

$x_i \leq x_j$ равносильны. Таким образом, φ представляет собой

изоморфизм между Σ и S . Если рассматривать строгий

частичный порядок на $\{X_i\}$, то неравенство

$X_i \subset X_j$ посредством φ в случае

($e \in X_i$ и $e \in X_j$) указывает на $x_i = x_j$, а в случае

($e \notin X_i$ и $e \in X_j$) указывает на $x_i < x_j$. Очевидно,

что строгого изоморфизма нет. В работе [Сметанин

2011] показано, что нет и изоморфизма, а есть гомоморфизм.

С точки зрения частичного порядка безразлично, что рассматривать – алгебраическую

систему множеств или изоморфную ей алгебраическую

систему характеристических функций этих множеств. Это

отражено в теореме Стона [Горбатов 1976]. Справедлива таким образом

Теорема 1. Алгебраическая система $\Sigma = \{X_i\}$

задаваемая системой множеств с определенным на них

отношением строгого частичного порядка (строгое

включение) не изоморфна алгебраической системе S

их индикаторов, на которой строгий частичный

порядок из $\Sigma = \{X_i\}$ отражается в частичный

порядок.

В случае рассмотрения строгого частичного

порядка строгого изоморфизма между этими системами

нет, а есть гомоморфное отображение первой во

вторую. При котором, теряется разделение

отношений строгого включения и равенства между

множествами. Проекционная модель [Вальков 1985] на

основе индикаторов настолько грубая, что любое

непустое подмножество универсума алгебраической

системы множеств отражается в ней как ноль либо

единица. Это в свою очередь является одной из

причин парадоксов материальной импликации. Таким

образом, мы не можем утверждать, что существует

взаимно однозначное соответствие между отношением

нестрогого включения $X \subseteq Y \equiv (X \subset Y) \oplus (X = Y)$ и

материальной импликацией их индикаторов $x \Rightarrow y$.

Смотри таблицу 1 и таблицу 2. Поэтому при моделировании

рассуждений необходимо отказаться от использования

нестрогого включения.

Далее будем рассматривать частично упорядоченные

отношением строгого порядка системы подмножеств

множества U , называемого далее универсумом. При

этом отдельные его элементы в случае надобности

будем считать его одноэлементными подмножествами.

Частично упорядоченное отношение строгого

порядка множество U называется строгой структурой

(антирефлексивной структурой), если в нем при любых

его подмножествах $X \subset U$, $Y \subset U$ система множеств

$\{X, Y\}$ имеет точную верхнюю и нижнюю границы. Антирефлексивная

структура U обладает свойством дистрибутивности, если

для ее элементов выполняется соотношение $(X \cup Y) \cap Z = (X \cap Z) \cup (Y \cap Z)$.

Нулем (0) и (1) единицей антирефлексивной структуры U

называются ее наибольший и наименьший элемент. В нашем

случае это U и пустое множество.

Булевой алгеброй называется дистрибутивная структура с

неравными друг другу нулем 0 и единицей 1, в которой всякий

элемент имеет дополнение. Таким образом булева алгебра

всегда содержит не менее двух элементов. Алгебра, содержащая только

0 и 1, называется вырожденной [Владимиров с.19]. Классическая логика построена на основе вырожденной булевой алгебры, в которой 0 отождествлен с абстрактной ложью, а 1 с абстрактной истиной. То есть, она отражает объективную реальность как систему событий (минуя моделирование событий множествами) даже не в систему характеристических функции этих множеств, а в абстрактные по Гильберту пропозициональные переменные.

Всякая алгебра множеств является булевой алгеброй относительно (строгого и нестрогого включения - как естественного упорядочения.) С каждой такой алгеброй автоматически связывается (в случае частичного упорядочения на основе нестрогого включения) изоморфная ей булева алгебра соответствующих характеристических функций. В случае частичного упорядочения на основе строгого включения между алгеброй множеств и алгеброй характеристических функций устанавливается неизоморфное отображение смотри Теорему 1, которое является причиной парадоксов материальной импликации. Любое суждение может быть выражено как соотношение между множествами, поэтому можно в принципе отказаться от вырожденной булевой алгебры и развивать классическую логику на основе невырожденной булевой алгебры множеств, хотя бы уже потому, что в ней нет парадоксов материальной импликации.

Для практических приложений невырожденных булевых алгебр построенных на основе конечных систем множеств важную роль играет

Теорема 2. [Кулик 1997] Если S_1, S_2, \dots, S_n , - конечная система S множеств со свойствами кольца или полукольца, то существует и может быть построена конечная система E различных множеств

E_1, E_2, \dots, E_m , ($m \geq n$) со следующими свойствами:

(1) для любой пары (E_i, E_k) при $i \neq k$ $E_i \cap E_k = \emptyset$;

(2) любое множество системы S в точности равно объединению некоторых множеств системы E . Система множеств E называется системой конститuent системы S . Количество конститuentных множеств $m \leq 2^n$.

3.3. Решение полисиллогизмов посредством исчисления системы конститuentных множеств.

При исследовании вопроса о том, какие множества можно построить посредством операций (объединения «+», пересечения «·», дополнения до универсума «'») - $X' = U \setminus X$ из порождающих n произвольных множеств X_1, X_2, \dots, X_n , вводится важное понятие конститuent. Обозначим

$$X_i^{\sigma_i} = \begin{cases} X_i, & \text{если } \sigma_i = 1 \\ \bar{X}_i, & \text{если } \sigma_i = 0 \end{cases}$$

Множество вида, $\prod_{i=1}^n X_i^{\sigma_i} = X_1^{\sigma_1} X_2^{\sigma_2} \dots X_n^{\sigma_n}$

где $\sigma_i = 0$ или $\sigma_i = 1$ назовем конститuentой.

Общее число не пустых конститuent не превосходит 2^n .

Существует важная связь множества пустых (непустых) конститuent с бинарными отношениями между множествами X_1, X_2, \dots, X_n . Эта связь заключается в том, что множество пустых (непустых) конститuent полностью определяется набором отношений между всевозможными парами порождающих множеств и их дополнений $X_1, X_2, \dots, X_n, X'_1, X'_2, \dots, X'_n$. Набор этих отношений, выраженных суждениями ортогонального базиса назван инвариантом алгебраической системы с образующими X_1, X_2, \dots, X_n .

Например, алгебраическая система на основе трех множеств изображена на рис. 5.

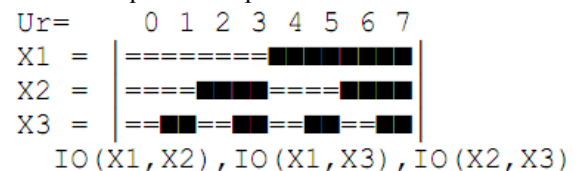


Рис.5 Алгебраическая система и ее инвариант.

Универсум задан как $Ur = \{0, 1, 2, 3, 4, 5, 6, 7\}$, где элементы есть номера, присвоенные конститuentам системы образующих множеств по порядку их нумерации. Номер 5 в двоичной форме 101 соответствует конститuentе $X_1 X_2' X_3$. Отношения IO - независимого пересечения между множествами определяют наиболее полную систему непустых конститuent количество которых в данном случае $8 = 2^3$. Каждой алгебраической системе с перенумерованными множествами носителями X_1, X_2, \dots, X_n можно сопоставить множество номеров непустых конститuent, которое названо базовым множеством номеров - БМН. Например, БМН алгебраической системы из 7 множеств $X1=D1, X2=D2, X3=D3, X4=S1, X5=S2, X6=S3$ и $X7=S4$ есть $Ur = \{12, 13, 28, 45\}$, все остальные конститuentы, определяемые номерами из множества $\{0..128\}$ являются пустыми множествами. Это предопределено отношениями, задаваемыми в (1) смотри раздел 2.3. В работе [Сметанин 2010], предлагается алгоритм, позволяющий строить БМН для системы перенумерованных образующих алгебраической системы и любой ее подсистемы, включая каждое из множеств X_i . Машинограммы являющиеся результатом работы компьютерной программы, реализующей данный алгоритм, показаны на рис. 1, 2, 5, 6. Предлагаемая интерпретация суждений как множеств образующих алгебраическую систему (булеву

алгебру множеств) позволяет теоретически решать любые задачи полисиллогистики.

Определение терминов. В конкретном рассуждении каждое множество обозначается каким-то именем (символом или набором символов). Эти имена мы будем называть терминами рассуждения. Условимся, что терминами являются обозначения пустого множества и универсума, а также обозначения дополнений множеств. Если из множества всех терминов исключить термины \emptyset и U , то оставшиеся термины мы назовем базовыми терминами суждения. В отношении $Rel(X, Y)$ будем называть множество X левой, а множество Y правой частью.

Определение суждения. Суждением называется выраженное с помощью базовых терминов отношение включения, равенства или IO , в левой части и правой части которого содержится множество, выраженное как пересечение, объединение, дополнения множеств, представленных некоторыми базовыми терминами. Множество в левой части называется **субъектом суждения**, множество базовых терминов, представленных в правой части и само отношение $A(., Y)$, $Eq(., Y)$, $IO(., Y)$, – **предикатами суждения**.

Рассуждением или полисиллогизмом назовем список суждений, разделенный на два класса. Первый класс называется посылками, второй заключениями. Будем говорить, что рассуждение (полисиллогизм) правильное (ный), если из множества посылок логически следует каждый элемент заключения.

Очевидно, список посылок задает отношения между порождающими множествами в алгебраической системе Кантора. Эти отношения могут противоречить или нет отношениям задаваемым на порождающих множествах множеством заключений.

Задача 1. Доказать логическое следование всех заключений силлогизма из данных посылок.

Задача 2. Рассматривая посылки силлогизма как отношения на порождающей системе терминов, найти максимальное множество следствий, каждое из которых логически следует из множества посылок. Это множество естественно назвать **решением полисиллогизма**. Рассмотрим пример решения Задачи Порецкого [Порецкий 1884]. Являющейся задачей 1.

Между птицами зоосада существует 5 соотношений, выраженных в форме суждений:

Птицы певчие – крупные или обладающие качеством Y . Птицы, не имеющие качества Y – или не крупные, или не имеют качества X . Птицы певчие, в соединении с крупными, объединяют всех птиц с качеством X . Каждая некрупная птица есть и певчая, или обладающая качеством X . Между птиц с качеством X совсем нет таких птиц с качеством Y , которые, не будучи певчими, были бы крупные.

Определить, **были ли птицы качества X певчие или нет** - $A(X, S)$. Узнать, содержится ли множество птиц качества X в множестве птиц качества Y - $A(X, Y)$. Найти, были ли среди качества X птицы качества Y и наоборот – $IO(X, Y)$.

Пусть X – множество птиц качества X , Y – множество птиц качества Y , S – множество певчих птиц, G – множество крупных птиц. Здесь заданы 4 термина G, S, X, Y и дополнительно указаны соотношения между элементами данных множеств, которые указывают на наличие дополнительных свойств каждого термина, наличествующих у других терминов. Универсум U это все птицы зоосада, при этом неявно подразумевается, что U есть сумма конstituент множеств-терминов G, S, X, Y . В ортогональном базисе силлогистики суждения задачи Порецкого будут (при замене AXY на $A(X, Y)$) выглядеть так.

$$\begin{aligned} P1. A(S, Y \cup G) & \quad P2. A(\bar{Y}, \bar{X} \cup \bar{G}) \\ P3. A(X, S \cup G) & \quad P4. A(\bar{G}, S \cup X) \\ P5. E(X, Y \cap \bar{S} \cap G) & = A(X, \overline{Y \cap \bar{S} \cap G}) \end{aligned} \quad (2)$$

На рис. 6 показана интерпретация суждений (2).

Ur=	5 7 8 9 12 13 15	
X1 =	====■	G A(G^, S), IO(G, X)
X2 =	■■■■====■	S A(G^, Y), A(S^, X^)
X3 =	==■■■■■■■■■■■■■	X IO(S, Y) A(X, Y)
X4 =	■■■■==■■■■■■■■■	Y

Рис. 6 Решение Задачи Порецкого и ее инвариант.

Справа от диаграммы приведен инвариант. Теперь можно определить, были ли птицы качества X певчие или нет (ответ – да). Можно узнать, то же в отношении птиц качества Y (ответ нет). Найти, были ли среди качества X птицы качества Y и наоборот (ответ да). Что касается, так называемых, логических уравнения и систем, то для данного подхода это прошлый и позапрошлый век. Например, результаты полученные Порецким [8, стр. 281] $x=xs$; $y=gy+g's$; $y=y+x$, тривиальным образом следуют из диаграммы, как и множество других, например $y=(s+gs'x'y)|gsx'y'$. Попробуем ответить на вопрос какой должен быть инвариант системы порождающих множеств, чтобы $Y \subset S$. Для этого мы должны либо добавить шестую посылку $A(Y, S)$ в постановку задачи, что приведет к пустоте конstituенты $GS'X'Y$, либо минимально «перестроить» Ur убрав из него ту же конституенту в результате мы получим новый инвариант. Ниже на диаграмме рис. 7 иллюстрируется решение модифицированной задачи с добавленной посылкой и его инвариант, отличие – отсутствие $GS'X'Y$ и замена $IO(X, Y)$ на $A(S', Y)$:

Ur=	5 7 8 12 13 15	
X1 =	====■	G A(G^, S) IO(G, X)
X2 =	■■■■====■	S A(G^, Y) A(S^, X^)
X3 =	==■■■■■■■■■■■■■	X A(S^, Y^) A(X, Y)
X4 =	■■■■====■	Y

Рис. 7 Интерпретация модифицированной задачи.

Идти по пути предначертанном Аристотелем, то есть формулировать правильные мыслительные формы (силлогизмы) заведомо бесперспективно (есть еще сориты, полисиллогизмы и другие формы умозаключений). Выход предложен в [Сметанин

2010], смысл его в том, чтобы строить интерпретацию умозаключения на базе подмножеств универсума - терминов рассуждения. Решение Задачи 2 проиллюстрировано на рис 1 и 2. Таким образом становится возможным верифицировать любую гипотезу на фоне данного множества посылок (суждений).

ЗАКЛЮЧЕНИЕ

Диагностика - предмет чисто методический; ее содержание составляют различные методы исследования. Никакое самое детальное и самое ясное изложение методов исследования с кафедры не может до конца научить диагностике. Все методы покоятся на восприятиях того или другого из органов чувств, а в диагностике, как уже говорилось, почти всех чувств одновременно. Это обстоятельство объясняет трудности, которые представляет диагностика. Только путем повторных, длительных и самостоятельных упражнений можно соответственным образом воспитать свои органы чувств, можно овладеть умением наблюдать и исследовать. Этим объясняется, почему опытный врач видит, слышит и осязает то, чего совершенно не замечает малоопытный. Но то же самое справедливо и для врачебного мышления, которое также вырабатывается посредством постоянного упражнения, путем активной самостоятельной работы. Закон, согласно которому развитие индивидуума повторяет развитие вида, имеет общее значение: он приложим и для образования. Чтобы стать ученым или врачом, нужно в сокращенном виде и ускоренным шагом пройти весь путь человеческой мысли и опыта в этом отношении: нужно научиться наблюдать, подмечать в частном общее, в общем - схватывать индивидуальное, видеть закономерность и логику в смене явлений. Активная и самостоятельная работа в одной области и с одним методом, как всякая тренировка в известном направлении, чрезвычайно облегчает в дальнейшем усвоение других методов и работу в других областях. Итак, практическая медицина вообще, и диагностика как ее методическая основа в частности, ввиду присущих им особенностей, требуют и особо тщательного и продуманного подхода к их изучению и усвоению. Здесь более чем где бы то ни было справедливо положение, что сущность образования всегда заключается в самообразовании. Только путем действительно самостоятельной работы, путем постоянного воспитания своих органов восприятия, путем настойчивого активного мышления можно овладеть методикой, в том числе, и **методикой применения логики**, но зато, владея методикой, можно без лишних проб и ошибок эффективно приобретать необходимые знания и опыт.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Бочаров 2010] Бочаров В.А., Маркин В.И. Силлогистические теории. – М.: Прогресс – Традиция, 2010. 336 с.
- [Вальков 1985] Вальков К.И. Проекционное моделирование и автоматизация. Учебное пособие для факультета повышения квалификации. Л.: ЛИСИ, 1985, 86 с.
- [Смирнов В.А. 2007] Смирнов В.А. Логико – методологическая модель диагноза// <http://logic.ru/ru/node/535>
- [Постовит 1991] Постовит В.А. Диагноз и диагностика в клинической медицине – Л., изд. ЛПМИ, 1991 г. С. 96
- [Владимиров 1969] Владимиров Д.А. Булевы алгебры. М.: Наука 1969. С. 320, л.
- [Лобанов 2009] Лобанов В.И. Русская вероятностная логика – М.: «Русская правда» 2009.- 320 стр.
- [Сметанин 2009] Сметанин Ю. М. Ортогональный базис силлогистики // Вестник Удмуртского университета. Серия математика, механика. Компьютерные науки. Вып. 4 . 2009 г. С .155-166
- [Сметанин 2010] Сметанин Ю. М. Алгоритм решения полисиллогизмов в ортогональном базисе посредством исчисления конституентных множеств // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2010. Вып. 4.-С. 172-185.
- [Сметанин 2011а] Анализ парадоксов материальной импликации в ортогональном базисе силлогистики // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2011. Вып. 4.- с. 144-162
- [Сметанин 2011б] Вероятностная логика и ортогональный базис силлогистики.// В настоящее собрание.
- [Горбатов 1976] Горбатов В.А. Теория частично упорядоченных систем. – М.: «Советское радио», 1976. – 336 с.
- [Кулик 1997] Кулик Б.А. Логические основы здравого смысла. Под редакцией Поспелова Д.А. – СПб.: Политехника, 1997. – 131 с.
- [Порецкий 1884] Порецкий П.С. О способах решения логических равенств и об одном обратном способе математической логики.// Собрание протоколов заседаний секции физико - математических наук общества естествоиспытателей при Казанском университете, т. 2, Каз., 1884.

MEDICAL DIAGNOSTICS AND ORTHOGONAL BASIS OF SYLLOGISTIC

Smetanin Yu.M., Smetanin M. Yu. *

Udmurt State University, Izhevsk
gms1234gms@rambler.ru

* *Republican Clinic and Diagnostic Center, Izhevsk*

Misha1977@rambler.ru

RESUME

In this paper we consider one of the possible application of a new logical model of reasoning based on a non – degenerative Boolean algebra of sets. Suggested methods for finding logical conclusions from the given system of premises initially focused on the use of computer technology solutions and visual display of logical relationships between the terms of the problem of medical diagnostics.

Key words: algebraic system, calculation of the constituent sets, medical diagnostics, syllogistic.



УДК 004.9:510

РАЗРАБОТКА СИСТЕМ ТЕСТИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ WEB-ОРИЕНТИРОВАННОЙ НОТАЦИИ МАТЕМАТИЧЕСКИХ ТЕКСТОВ, СОХРАНЯЮЩЕЙ СЕМАНТИКУ

Вовк А.И.* , Рубан Ю.Я.* , Гирнык Д.А.**

* Государственный научно-исследовательский институт
автоматизированных систем в строительстве, г.Киев, Украина

vovk@ndiasb.kiev.ua

ruban@ndiasb.kiev.ua

** Международный научно-учебный Центр информационных технологий и систем
Национальной Академии наук Украины, г.Киев, Украина

den@girnyk.com

В работе рассмотрена система тестирования, использующая WEB-ориентированную нотацию, реализующую естественный язык представления математических текстов и сохраняющую их семантику.

Ключевые слова: WEB-математика, математическая нотация, дистанционное обучение, тестирование.

ВВЕДЕНИЕ

Особенностью существенной части многих научных и образовательных ресурсов, содержащих математику, физику, экономические науки и т.п., является наличие математических формул. В частности, при разработке систем тестирования по техническим дисциплинам приходится сталкиваться с необходимостью использования математических выражений как при формулировании заданий, так и при подготовке ответов.

1. Математические тексты в системах тестирования

ГНИИ автоматизированных систем в строительстве проводит аттестацию ведущих работников отрасли. Одним из обязательных условий оценки знаний является прохождение специальных тестов. Используя обычные схемы тестовых заданий, например, выбор одного или нескольких правильных ответов из заданного списка предлагаемых, не всегда удается эффективно решать задачу оценки знаний субъектов тестирования. Поскольку проблемы в различных отраслях знаний аналогичные, достаточно рассмотреть этот вопрос на примере тестирования по математике. Сначала рассмотрим традиционный вариант тестового задания, который представлен на рис. 1.

Пример тестового вопроса "Выбор многих (одного) из многих"

Укажите значение наибольшего среди действительных корней алгебраического уравнения 4-го порядка

$$x^4 - 4x^2 + 8x - 4 = 0$$

- $1 + \sqrt{7}$
- $3 + \sqrt{3}$
- $1 - \sqrt{7}$
- 2
- $-1 + \sqrt{3}$
- $2 + \frac{\sqrt{3}}{2}$

ОТΟΣЛАТЬ ОТВЕТ

Рисунок 1 - Скриншот страницы сайта <http://asdev.com.ua/do/>

При таком подходе к тестированию участник тестирования не обязан знать, как решаются подобные уравнения и что такое комплексные числа. Для подстановки представленных вариантов решений в исходное уравнение ему достаточно знать математику в пределах алгебраических преобразований, выполнив которые, он найдет нужный ответ.

Теперь предложим более сложное задание, представленное на рис. 2.

Пример тестового вопроса "Ввод математического выражения"

Наберите (в желтом окне) с помощью клавиатуры значение наибольшего среди действительных корней алгебраического уравнения 4-го порядка

$$x^4 - 4x^2 + 8x - 4 = 0$$

$-1 + \sqrt{3}$

$-1 + \sqrt{3}$

ОТΟΣЛАТЬ ОТВЕТ

примеры оформления ответа
ВНИМАНИЕ: это не варианты ответа

как писать	результат
$4 + \sqrt{7}$	$4 + \sqrt{7}$
$-1/4 + \sqrt{5}/4$	$\frac{-1 + \sqrt{5}}{4}$

Рисунок 2 – Скриншот страницы сайта <http://asdev.com.ua/do/>.

В этом случае для нахождения ответа необходимо уметь решать подобные уравнения, в том числе знать, что такое комплексные решения. Для ввода, корректировки и проверки правильности ответа создан плагин, в основе которого используется нотация редактора математических формул MathTextView ([MathTextView, 2000], [Вовк и др., 2000], [Вовк и др., 2007], [Вовк и др., 2011]), которая сохраняет семантику формулы. Для чего это необходимо при создании тестовых заданий по математике, физике, экономике? Ведь существует широко распространенный язык TeX и ряд

родственных ему языков (см. Д. Кнут [Knuth, 1984]). Однако этот язык не обладает информацией о семантике формулы. Это, во-первых, затрудняет создание разумного WISIWIG-редактора (What You See Is What You Get, что видишь, то и получишь). Например, в рассматриваемом плагине все промежуточные операции завершаются "на лету". Редактор добавит отсутствующую скобку, потребует ввода недостающего операнда и т.д. Во-вторых, наличие семантики дает возможность корректно проверить правильность решения. Ведь даже в таком простом примере, представленном нами, может быть, по крайней мере, два варианта правильного ответа: $-1 + \sqrt{3}$ и $\sqrt{3} - 1$, если не учитывать, что $\sqrt{3} = 3^{1/2}$ или даже $\sqrt{3} = 3^{0.5}$ (хотя последние два случая можно и исключить, заранее оговорив форму ответа или расширив соответствующим образом список правильных ответов). Если же попросить субъекта тестирования для рассматриваемого случая выписать все действительные решения уравнения, например, в виде массива $(-1 + \sqrt{3}, -1 - \sqrt{3})$, то в этом случае добавится еще одна неоднозначность, связанная с порядком записи решений в массиве. Таким образом, приходим к выводу, что без сохранения семантики, невозможно или очень затруднительно корректно оценить правильность ответа. Предлагаемый плагин по мощности символьных преобразований, конечно, уступает такому продукту, как Maple [Maple], но для тестовых заданий вполне может быть пригоден. Что касается известного редактора, использующего язык математической разметки MathML, то он отвергается нами по причине сложной нотации, которая вряд ли когда-нибудь будет реализована в системах тестирования. Язык MathML – это язык, который целесообразен при машинной обработке математических текстов. Подчеркнем, что язык разметки MathTextView основан на том, что математические выражения сами обладают строгой структурой, являющейся носителем семантики.

Для демонстрации возможностей языка MathTextView, применяемого для выполнения целей тестирования с использованием математических текстов, приведем еще пару примеров. Отметим, что нотация MathTextView позволяет отобразить около 250 математических объектов [MTV]:

- арифметические операции;
- отношения;
- элементарные функции;
- пределы;
- интервалы;
- скобки;
- логические операции;
- представление множеств;

- операции над множествами;
- произвольные функции, индексы;
- векторы, матрицы;
- производные;
- интегралы, ряды, произведения;
- кванторы;
- разное;
- греческие буквы;
- готические буквы;
- спецсимволы;
- схематические рисунки;
- графики;
- размерности.

Пример нотации логической конструкции представлен на рис. 3.

Пример тестового вопроса "Ввод математического выражения"

Запишите (в желтом окне) решение нелинейного уравнения

примеры оформления ответа
ВНИМАНИЕ: это не вариант ответа

как писать	результат
$(x=1<if-(a=0))\&e(x=2/\sqrt{4a-3}<if-(a>=0))$	$\begin{cases} x=1, & \text{если } a=0 \\ x=\frac{2}{\sqrt{4a-3}}, & \text{если } a\geq 0 \end{cases}$

Рисунок 3 – Скриншот страницы сайта <http://asdev.com.ua/do/>.

Особенность этого примера – использование логических операторов для отображения обычных ситуаций, возникающих при решении школьных примеров повышенной сложности.

Следующий пример, представленный на рис. 4, посвящен решению систем уравнений,

представленных в матричном виде.

Пример тестового вопроса "Ввод математического выражения"

Запишите (в желтом окне) решение нелинейного уравнения

примеры оформления ответа
ВНИМАНИЕ: это не вариант ответа

как писать	результат
$(x(_1);x(_2);x(_3))=(14;32;50)$	$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 14 \\ 32 \\ 50 \end{pmatrix}$

Рисунок 4 – Скриншот страницы сайта <http://asdev.com.ua/do/>.

Отметим здесь, что с учетом семантики, например, x^2 – означает возведение в квадрат в отличие от $x^{(2)}$, где $(^2)$ – верхний индекс.

Система тестирования, реализующая предложенные методы, построена в формате IMS QTI (Instructional Management Systems Question and Test Interoperability) на языке XML и основана на IMS ([García, 2010], [Harchay, 2010], [Gutiérrez, 2010]), одном из ключевых отраслевых стандартов в области e-Learning, который поддерживается IMS Global Learning Consortium [IMS GLC].

2. Плагин для тестирования по математике

Плагин имеет следующую структуру входных параметров.

```
<OBJECT classid=
"clsid:459E7323-B9AB-4887-8A6E-6318AE4F3C07">
  <param NAME="TextWIDTH" VALUE="254">
  <param NAME="TextHEIGHT" VALUE="157">
  <param NAME="TestNum" VALUE="6">
  <param NAME="Answer" VALUE="-1+sqrt(3)">
  <param NAME="CorrectAnswer"
    VALUE="a-213+4s5q6r7t8(931)">
```


</OBJECT>

Отметим, что здесь представлен “мгновенный снимок” HTML-кода, поскольку значения параметров TestNum, Answer и CorrectAnswer формируются “на лету” с помощью программного обеспечения, расположенного на сервере. Кроме того, этот плагин обладает еще одной возможностью – он отправляет вариант ответа на сервер. Значение параметра CorrectAnswer закодировано во избежание “подглядывания” правильного ответа. Конечно, здесь необходимо дополнительно также использовать системные методы защиты HTML-кода от чтения. Значение параметра Answer необходимо для предоставления возможности корректировки ответа, если есть резерв отпущенного времени. Параметры TextWIDTH и TextHEIGHT задают соответственно ширину и высоту формы, являющейся контейнером плагина.

ЗАКЛЮЧЕНИЕ

В статье рассмотрены вопросы разработки систем тестирования, использующих WEB-ориентированную нотацию, реализующую натуральный язык представления математических текстов и сохраняющую их семантику.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [MathTextView, 2000] Математика в Интернете [Электронный ресурс]. – 2000. -Режим доступа: <http://math.accent.kiev.ua>. – Дата доступа: 30.11.2011.
- [Вовк и др., 2000] Вовк А.И. и др. Язык представления математических текстов в интернете. В кн.: Теорія та методика навчання фундаментальних дисциплін у вищій школі. Збірник наукових праць. Кривий Ріг, Вид. відділ НМетАУ, 2004.
- [Вовк и др., 2007] Вовк А.И., Гирнык Д.А. Язык общения математиков в Интернете. В кн.: New Information Technologies in Education for all: State of the art and Prospects (ITEA-2007), Kiev, Ukraine, IRTC, 21-23 November 2007, p.p. 96 – 103.
- [Вовк и др., 2011] Вовк А.И., Гирнык Д.А. Web-ориентированная нотация математических текстов, сохраняющая семантику. В кн.: Открытые семантические технологии проектирования интеллектуальных систем. Минск, 2011, с.431-434.
- [Knuth, 1984] Donald E. Knuth, The TeXbook (Reading, Massachusetts: Addison-Wesley), 1984.
- [García, 2010] García, A.; Barchino, R.; de Marcos, L.; García, E.; Hilerá, J.-R.; Gutiérrez, J.-M.; Otón, S.; Martínez, J.-J.; Gutiérrez, J.-A. Tool for Generation IMS-QTI v2.1 Files with Java Server Faces // IEEE 10th International Conference on Advanced Learning Technologies (ICALT), 2010, pp. 627 – 628.
- [Harchay, 2010] Harchay, A.; Cheniti-Belcadhi, L.; Braham, R.; An Investigation of the Enhancement and the Formal Description of IMS/QTI Specification for Programming Courses // IEEE 10th International Conference on Advanced Learning Technologies (ICALT), 2010, pp. 113 – 115.
- [Gutiérrez, 2010] Gutiérrez, I.; Kloos, C.D.; Crespo, R.M.; Assessing assessment formats: The current picture: Special session: Assessing assessment formats // IEEE 10th International Conference on Education Engineering (EDUCON), 2010, pp. 1233 – 1238.
- [IMS GLC] IMS Global Learning Consortium [Электронный ресурс]. –1997. -Режим доступа: <http://www.imsglobal.org/>. – Дата доступа: 05.12.2011.
- [MTV] Редактор математических текстов.help [Электронный ресурс]. – 2000. -Режим доступа: http://math.accent.kiev.ua/article/04/png_htm/04_00_png.htm. – Дата доступа: 05.12.2011.
- [Maple] MapleSoft – [Электронный ресурс]. – 1990 <http://www.maplesoft.com/>. - Дата доступа: 05.12.2011.

DEVELOPMENT OF TEST SYSTEMS USING WEB-ORIENTED NOTATION, MATHEMATICAL TEXTS, WHICH PRESERVES THE SEMANTICS

Vovk A.I. *, Ruban Y.Y. *, Girnyk D.A. **

* *State Scientific-Research Institute automated systems in the building, Kiev, Ukraine*

vovk@ndiasb.kiev.ua

ruban@ndiasb.kiev.ua

** *International Research and Training Center of Information Technologies and Systems*

National Academy of Sciences of Ukraine, Kiev, Ukraine

den@girnyk.com

The paper considers the testing system that uses a WEB-oriented notation, which realizes natural language presentation of mathematical texts and preserving their semantics.

INTRODUCTION

Feature of a substantial part of many scientific and educational resources that contain mathematics, physics, economics, etc., is the existence of mathematical formulas. In particular, the development of test systems in technical disciplines faced with the need to use mathematical expressions, as in the formulation of tasks and the preparation of responses.

MAIN PART

In the systems tested in mathematics using schemes with a choice of one or more correct answers from a list of possible answers is often does not lead to effective evaluation of the knowledge test subjects.

The paper proposes to use for this purpose mathematical text editor mathtextview. notation mathtextview, in contrast to the notation of tex, the semantics of a mathematical expression.

We propose specifically designed for this purpose, plug-in. we consider a number of examples of the test section of the school mathematics with the use of this plugin.

CONCLUSION

The questions of development of test systems that use WEB-oriented notation, which realizes natural language presentation of mathematical texts and sohranyayuschnyu their semantics.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

КОГНИТИВНЫЕ СЕТИ И ОНТОЛОГИЧЕСКИЙ АНАЛИЗ В ПОВЫШЕНИИ АДАПТИВНОСТИ И КАЧЕСТВА ОБСЛУЖИВАНИЯ В ГЕТЕРОГЕННОЙ БЕСПРОВОДНОЙ СРЕДЕ

Гладун А.Я.* , Рогушина Ю.В.**

* *Международный научно-учебный центр информационных технологий и систем НАНУ и МОНУ, г.Киев, Украина*
glanat@yahoo.com

** *Институт программных систем НАНУ, г.Киев, Украина*
ladamandraka2010@gmail.com

В работе рассматривается использование онтологий для представления базы знаний когнитивной сети, обеспечивающей повышение качества обслуживания в адаптивных коммуникационных системах, а также методы пополнения этой базы знаний.

Ключевые слова: онтология, когнитивная сеть, адаптивные коммуникационные системы.

ВВЕДЕНИЕ

Развитие телекоммуникационных сетей является элементом общего процесса развития общества и тесно взаимосвязано практически со всеми процессами, происходящими в общественных отношениях, науке и технике, промышленности и других областях деятельности человека. Рост потребности в информационном обмене и его значимости в жизни обеспечивает внедрение новейших технологий в сетях телекоммуникаций. Последние, в свою очередь, также оказывают влияние на процессы, происходящие в жизни и деятельности человека.

Когнитивная организация взаимодействия сетей сегодня является ключом к успеху в обслуживании клиентов в сложных динамических конфигурациях сетей и эффективной их эксплуатации. Основные цели создания когнитивных сетей заключаются в том, чтобы на основании знаний о предметной области повысить эффективность использования коммуникационных ресурсов и увеличить производительность, как линий связи, так и сети в целом.

К когнитивным процессам относят деятельность, направленную на приобретение знаний: восприятие, рассуждение, принятие решения, разрешение возникших проблем и, возможно, интуицию. Использование таких процессов в управлении беспроводными сетями позволяет адаптировать систему к современной динамической

и гетерогенной беспроводной среде, к которой относятся Интернет, мобильная телефония и ряд других беспроводных систем.

Эта среда обеспечивает пользователю целый спектр новых программных приложений, которые поддерживаются разработчиками Web-сервисов. Но при этом существенно возрастает сложность, как пользовательских устройств, так и самой сети (в частности, на базе Интернет формируется новая концепция развития сетей NGN - сети следующего поколения). Усложняются при этом и задачи управления такими сетями [Bari, 2006]. Развитие новых технологий в телекоммуникациях вызывают в свою очередь потребность поддержки качества сервисов, защиты и мобильности для новых сервисов, которые могут быть доступны беспроводной гетерогенной сети. Поэтому, как и в других направлениях информационных технологий (ИТ), несмотря на существующие трудности, многими мировыми исследователями ведутся работы по внедрению семантических, ориентированных на знания методов и технологий в сетях.

Термин «когнитивные беспроводные сети» используется для обозначения сетей, способных учитывать текущий контекст выполняемых операций, осуществлять анализ, логический вывод и планирование, принимать решения и действовать в соответствии с принятым решением, а также учиться на собственном опыте [Thomas, 2007]. Когнитивная сеть включает также элементы, обеспечивающие ее функциональность (элементы ментальной деятельности, функции мониторинга,

сбора информации, исполнительные устройства и др.). Когнитивные сети могут динамически адаптировать свои рабочие параметры в соответствии с потребностями пользователей или изменения условий окружающей среды. Они могут учиться на основе предыдущего опыта и использовать знания, чтобы предопределить свои будущие решения. Когнитивные сети относят к сетям будущего и «они нужны просто потому, что они позволяют пользователям сосредоточиться на вещах, кроме настройки и управления сетями» [Kliazovich, 2010].

1. Когнитивные сети – основные определения и проблематика

1.1 Определение когнитивной сети

В последние годы термин "когнитивный" ("smart") часто применяют к различным конвергентным сетям и коммуникационным системам. В литературе встречаются упоминания о когнитивном радио, «смарт» радио [Mitola, 2005], интеллектуальной антенне [Alexiou, 2004], когнитивных пакетах, «смарт»-пакетах [Gelenbe, 2001] и когнитивных сетях [Ramming, 2004], [Thomas, 2007]. Объединяет эти технологии способность к самообучению и адаптации к новым условиям.

Анализ концепции когнитивных сетей показывает, что сейчас в интеллектуальных сетях четко обозначилось смещение интереса от управления ресурсами к пониманию и анализу и обеспечению пользовательских потребностей и сетевой «интеллект» перемещается все далее к «краям сети» [Balamuralidhar, 2008]. Однако ни одна из проанализированных работ, к сожалению, не определяет точно, что такое когнитивная сеть и не регламентирует, как она должна работать.

Когнитивная сеть – это тип сетей передачи данных, в которых обеспечивается возможность для семантической обработки текущего контекста операций, анализа, логического вывода и планирования своих действий, что позволяет принимать решения и действовать в соответствии с достигнутым решением с учетом предыдущего опыта [Ramming, 2004]. Когнитивные сети имеют способность думать, обучаться, запоминать и адаптироваться к непостоянным условиям для того, чтобы достигнуть своих целей и задач и, таким образом, должны владеть самосознанием. Архитектура когнитивных сетей базируется на технологиях принятия решений и технологиях управления знаниями о предметной области.

Когнитивные сети, прежде всего, будут использоваться для межуровневой оптимизации сети и управлять динамикой действий, одновременно используя параметры, принадлежащие множественным уровням в стеке протоколов сети.

В коллективной работе авторов из университета штата Вирджиния, США [Thomas, 2005] сформулированы требования к когнитивной сети, где

она призвана обеспечить в заданный промежуток времени, лучшую непрерывную производительность, чем некогнитивная сеть. Когнитивность (познание) должно использоваться для того, чтобы улучшить управление ресурсом, качеством сервиса (QoS), безопасностью, управлением доступом и много других сетевых целей. Наиболее актуальными кандидатами для реализации когнитивных сетей авторы считают: специализированные (ad-hoc) сети; беспроводные сети в режиме инфраструктуры; наземные проводные сети и гибридные сети. Идеально, когнитивная сеть должна быть дальновидной и предусмотрительной, а не реактивной, немедленно реагирующей на действия и должна пытаться корректировать проблемы прежде, чем они произойдут.

В [Комашинский, 2011] рассматриваются пять уровней когнитивной системы (КС). В частности, выделено несколько взаимосвязанных доменов когнитивной системы: физический, информационный и когнитивный. Каждая из подсистем КС потенциально имеет в своем составе элементы, относящиеся к каждому из этих доменов. В физическом домене происходят энергетические процессы и взаимодействуют технические системы в процессе переноса информации между источниками и получателями. Информационный домен – это область, в которой находятся данные, информация, знания, методы. В когнитивном домене происходит анализ ситуаций и интеллектуальная деятельность, продуктом которой являются оценки и принятия решений. С учетом этого элементы КС классифицируются по отношению к подсистемам и доменам.

Уровень когнитивной подсистемы выполняет функции прикладного искусственного интеллекта и интеллектуального управления в реальном масштабе времени объектами, располагающимися на первом уровне архитектуры. Для этого КС постоянно взаимодействует с сетями сенсоров, исполнительных устройств и информационной подсистемой. КС содержит также комплекс инструментов (цифровых процессоров, нейропроцессоров, нечеткой логики, соответствующего математического, логического и программного обеспечения), который обеспечивает реализацию когнитивных прикладных процессов и сервисов.

Когнитивные сети могут быть централизованными или распределенными. Централизованный подход обеспечивает существенную обработку данных, коммуникацию, время и преимущества памяти, но обладают классическими недостатками централизованной архитектуры (например, узкие места производительности и единственные пункты отказа). Распределенные когнитивные сети могут быть сформированы как коллекция когнитивных объектов, которые включают интеллектуальные функциональные возможности, имеют возможности

логического вывода, характеризуются автономностью, социальной способностью, обучением из опыта, и адаптивностью, в то время когда они взаимодействуют с другими компонентами и действуют реактивным или проактивным способом для достижения своих целей [Thomas, 2006].

В когнитивных сетях явно не указывается способ представления знаний о сети, но при этом наличие таких знаний необходимо для обеспечения когнитивного цикла и выполнения целей сети. В частности, для представления таких знаний могут использоваться онтологии предметной области, пригодные как для машинной обработки, так и для понимания человеком-пользователем [Balamuralidhar, 2008].

2.1 Архитектура когнитивной сети

Расширяя последовательность шагов (цикл) реализации интеллектуального управления автономной системы, состоящего из последовательности “мониторинга, анализа, планирования и выполнения”, универсальная архитектурная платформа когнитивной системы может включать следующие модули: 1) модуль мониторинга и сбора данных; 2) модуль логического вывода (рассуждения); 3) модуль обучения; 4) модуль принятия решения; 5) исполнительный модуль; 6) модуль политики и 7) модуль управления знаниями как показано на Рис.1. Когнитивная система непрерывно зондирует свою среду, чтобы идентифицировать потенциальные условия, которые могли бы оказывать воздействие на ее функционирование.



Рисунок 1 – Архитектура платформы когнитивной системы

Рассмотрим назначение основных модулей когнитивной сети. Модуль мониторинга и сбора данных объединяет, коррелирует и фильтрует данные до тех пор, пока условие, которое должно быть далее проанализировано, не будет идентифицировано. Наблюдения, зафиксированные модулем мониторинга и сбора данных, будут обработаны и проанализированы модулем логического вывода и в то же время они направляются в модуль обучения, который в

состоянии изучить и запомнить полезные наблюдения, которые могут помочь процессу принятия решения в будущем.

Модуль логического вывода определяет потенциальные действия, которые будут базироваться на наблюдениях, знаниях, приобретенном посредством модуля обучения и политик, хранящихся в модуле политики. Модуль принятия решений определяет действия, которые будут взяты из эксплуатации системы, а также из обучения на опыте. Исполнительный модуль реализует результаты обработки, полученные модулем принятия решения.

Модуль обучения может обучаться на основе нескольких источников, например, из собранной информации, стратегий, решений, и полученной обратной связи; он может также коррелироваться и выводить из этого знания.

Модуль управления знаниями доступен и используется другими модулями системы и предназначен: для создания и хранения онтологий и тезаурусов о предметной области (ПрО); для однозначного использования терминологии ПрО; для семантической обработки данных в «сырое» знание из неструктурированных и необработанных данных (все потоки данных должны быть связаны с онтологическими понятиями и количественными данными); для приобретения знаний модулем обучения (используется ID3, как индуктивный метод для создания дерева решений). Мы использовали оригинальный алгоритм для этой цели, который использован в некоторых различных экспертных системах.

Одним из важных элементов в любой беспроводной системе, который существенно влияет на качество передаваемых данных, является механизм хэндовера. Поэтому, концентрируясь на процессе управления хэндовером, модуль мониторинга и сбора данных идентифицирует подходящие сети доступа (среди множества сетей присутствующих на данный момент в эфире), которые могли бы обеспечить решение ABC-проблемы (автоматического выбора сети), измеряет и агрегирует QoS-связанную информацию, наблюдает за текущим контекстом (например, скорость пользователя и местоположение пользователя, состояние зарядки его батареи) и направляет эти метаданные к модулю логического вывода. Модуль логического вывода решает, нужно ли инициализировать процесс хэндовера (переключение на другую базовую станцию), принимая во внимание накладные расходы передачи данных и потенциальные эффекты зашумленности, основываясь на информации, полученной от модуля мониторинга и сбора данных и знания, приобретенного от модуля обучения и модуля управления знаниями. Кроме того, на этом шаге отбрасываются сети-кандидаты доступа в случае, если они не удовлетворяют определенной политике, принятой в системе. Например, сеть, возможно, не в состоянии оказать требуемую услугу или

удовлетворить минимальные требования QoS, ожидаемые пользователем (например, минимальная скорость передачи данных). В дополнение, в случае, если пользователь движется с высокой скоростью, сети с маленьким ареалом покрытия могут быть изъяты из списка кандидатов (например, WLAN исключены в случае, если скорость пользователя превышает порог). Кроме того, модуль логического вывода может ограничивать список кандидатов, в случае, если репутация оператора сети-кандидата, управляющего определенной сетью доступа, слишком низкая.

2. Методы многокритериального выбора сетей обслуживания

2.1. Обзор традиционных методов выбора сетей обслуживания

Предоставление услуг в базирующейся на IP гетерогенной беспроводной сетевой среде требует выбора оптимальной сети доступа, чтобы снизить стоимость или повысить качество сервиса. На выбор сети в такой среде влияет множество различных факторов, и в настоящее время не существует общепринятого подхода для решения этой проблемы.

Проблема выбора сети среди гетерогенных беспроводных сетей получила недавно большое внимание из-за развития конвергентных коммуникационных систем. В [Song, 2004] предлагается объединить две математические методики в алгоритме для выбора сети между Универсальной Системой Мобильной Связи (UMTS) и беспроводными локальными сетями (WLANs). В этой работе аналитический процесс иерархии и теория систем Грея используются для того, чтобы оценить персональные настройки пользователя и требования сервиса и скомбинировать приоритетные установки атрибутов QoS с производительностью альтернативных сетей для того, чтобы принять решение о выборе сети.

Динамическая система, выбирающая сеть для предоставления сервиса, базирующегося на текущих рыночных условиях, таких как QoS и, стоимости атрибутов, был описана в [Le Vodice, 2000]. Используя предложенную структуру, пользователь может выбрать сеть доставки по телефону. Выбор сети, основанный на стратегии распределения ресурсов для наиболее эффективного использования ресурса в гетерогенной сетевой среде, был предложен в [Bari, 2007]. Базирующаяся на нечеткой логике многокритериальная система принятия решения, производящая выбор сети доступа в гетерогенной сетевой среде, описана в [Hongvan, 2003]. Простая система, поддерживающая политику хэндовера, в рамках гетерогенных беспроводных сетей представлена в [Гладун, 2009], которая позволяет пользователям выражать свою политику согласно с тем, какая из них является лучшей беспроводной системой в любой момент, и

выбирать компромисс между сетевыми характеристиками и динамическим поведением, таким как стоимость, производительность, и потребляемая мощность. У механизмов, описанных в вышеупомянутых работах, есть существенные недостатки.

Факторы, рассматриваемые в решающем процессе, недостаточны; например, информация о поддерживаемых типах доступа, поддерживаемые типы аутентификации, и поддерживаемые партнеры по роумингу, не рассматриваются в решающем процессе, которые описанные в этих трудах.

Использование примеров описанных сценариев являются ограниченными и не реалистичные для перспективы развертывания или бизнес-моделей, используемых в промышленности. В результате они не обеспечивают законченных и развертываемых решений проблемы.

Решение о выборе сети в гетерогенной беспроводной среде зависит от различных факторов. Проблема выбора сети может быть решена, используя MADM-алгоритмы. MADM был активной областью исследования с 1970-ых. Из-за их детерминированной природы и несложной реализации, алгоритмы MADM нашли применение в решении большого количества проблем, от общественных наук до исследования операций. Они могут использоваться в комбинации с нечеткой логикой, где входные значения атрибутов ясно не определены.

Этапы процесса выбора сети на основе рассмотренных выше методов представлены на рис.2.



Рисунок 2 – Процесс принятия решения при выборе сети.

Следует отметить, что на всех этих этапах не используются формализованные знания об анализируемых сетях и их свойствах, а лишь на основе анализа их формальных параметров.

2.2. Принятие решения по выбору сети

В механизме, который осуществляется при помощи сети, предложенном в [Bari, 2007], сеть

помогает терминалу в процессе выбора, выполняя сбор данных и их анализ, обеспечивая ранжирование сетей. Архитектура предлагает использование трех базирующихся на сети функциональных объектов: 1) узел сбора данных (DCN) для поиска сетевых характеристик; 2) узел объявления сервиса (SAN) для обеспечения данных обслуживающих сервисы; 3) узел аутентификации, авторизации и учета (AAA) для AAA-информации.

Вместе они обеспечивают входные данные для базирующегося на сети логического объекта оценки, который вычисляет ранжирование сети для использования терминалом при выборе сети. Кроме того, терминал обеспечивает свое местоположение и любую другую информацию, которую могла бы рассматривать сеть для анализа (сервис/QoS по запросу, просмотры PLMN, SSIDs, и т.д.).

Процесс начинается с терминала, пытающегося получить связь с сетью, несмотря на то, какой бы ни была эта сеть (с или без аутентификации). Этот начальный доступ к сети подобен получению информации у справочного стола по сегодняшней телефонной сети с коммутацией каналов. Начальная сеть, возможно, не оптимальная сеть для требуемого сервиса и может быть заменена, как только будет собрана достаточная информация, чтобы принять решение. Используя информацию, собранную от узлов DCN, SAN, и AAA, и ту которую обеспечивает терминал, сеть вычисляет ранжирование подходящих сетей доступа и предоставляет их терминалу. В завершении этой работы мы представляем комплексный алгоритмический подход, который будет использоваться базирующимся на сети логическим объектом оценки в предложенной архитектурной платформе.

Процесс решения выбора сети в гетерогенной сетевой среде можно рассматривать как MADM-проблему, которая оценивает ряд альтернатив, используя их атрибуты. Решающий процесс ранжирует альтернативы в порядке предпочтения, используя набор атрибутов, которые обеспечивают различные аспекты, посредством которых альтернативы могут быть рассмотрены.

Решение о выборе сети зависит от нескольких факторов, которые можно разделить на две категории: 1) атрибуты, включающие параметры, не связанные, которые редко изменяются и поэтому могут обычно обеспечиваться сетью; 2) атрибуты, связанные с QoS, как динамические, так и в значительной степени статические.

В Таблице 1 представлен список атрибутов, которые могут быть учтены при выборе сети, используя алгоритм, предложенный в этой статье, с разделением их на эти две категории.

Таблица 1. Атрибуты, учитываемые при выборе сети.

Атрибут	Описание атрибута	Категория
Имя оператора ON	идентификация оператора (например, партнер по роумингу SSID или идентификатор PLMN ID), для которого нужна остальная информации.	1
Механизм аутентификации AM	механизм аутентификации, используемый партнером роуминга (например, SIM или пароль пользователя).	1
Технология доступа AT	Технология беспроводного доступа для использования (например, UMTS, WiMAX, WLAN).	1
Географическое положение GL	информация о географическом положении базовой станции.	1
Область покрытия CA	измерение зоны покрытия, например, hotspot физического адреса	1
Стоимость байта CB	транспортные расходы оператора для отдельной сети доступа с учетом использования лицензированного спектра частот и соглашения о роуминге.	2
Суммарная полоса пропускания ТВ	диапазон полосы пропускания, доступной для всеобщего использования для линий беспроводного доступа	2
Разрешенная полоса пропускания АВ	полоса пропускания, разрешенную для пользовательского использования	2
Коэффициент использования U	измерение текущего использования беспроводного канала	2
Задержка пакета D	средняя задержка пакета в пределах системы доступа	2
Дрожание пакета J	средние изменения задержки в пределах системы доступа	2
Потеря пакета L	средняя норма потери пакета в пределах системы доступа.	2

3. Семантические технологии в выборе сети обслуживания

3.1. Технологии Semantic Web как основа представления знаний

Проект Semantic Web предлагает мощный

практический подход к получению средств управления большим количеством информации и информационных услуг. Основными компонентами Semantic Web являются онтологии, сервисы и программные агенты [Хорошевский, 2008]. Semantic Web базируется на наборе открытых стандартов [Андон, 2008], в частности, на языках описания метаданных RDF [RDF, 2002]; представления онтологий OWL [OWL, 2004]; запросов к RDF SPARQL [SPARQL, 2008]; определения правил SWRL [SWRL, 2004].

Более детальный анализ технологий Semantic Web и их использование в разработке интеллектуальных приложений приведен в [Гладун, 2008, Gladun, 2009], а проблемы управления знаниями в Semantic Web - в [Гладун, 2007].

Модель знаний Semantic Web можно интегрироваться с прикладными программами и использоваться для отдельных доменно-ориентированных бизнес-логик из самой программы. Таким образом, Semantic Web – это мощное направление для повышения эффективности распределенного и совместного доступа к информации и ее использование прикладными программами. Технологии Semantic Web целесообразно применять и для управления адаптивными коммуникационными системами.

3.2. Формализация понятия логического вывода на онтологиях на базе DL

Важной особенностью онтологий и языков их представления, разработанных в рамках Semantic Web, является то, что они базируются на дескриптивных логиках. Дескриптивные логики (DL) возникли как расширения фреймов и семантических сетей механизмами формальной логики. DL позволяют описывать понятия предметной области в формализованном виде. Каждая DL является логикой первого порядка. Базовые элементы DL – это множество классов NC; Множество индивидуумов NI; Множество отношений NR. DL объединяют в себе, с одной стороны, богатые выразительные возможности, а с другой – удовлетворительные вычислительные свойства (разрешимость и относительно невысокую вычислительную сложность основных логических проблем), что позволяет применять их на практике.

Каждая конкретная DL характеризуется набором конструкторов и индуктивным правилом, с помощью которого составленные концепты данной логики строятся из атомарных концептов и атомарных ролей, используя эти конструкторы. В частности, OWL-DL является синтаксической вариацией высоко выразительной дескриптивной логики SHOIN(D).

Ключевое отличие логик (например, логики предикатов первого порядка FOL, DL, а также OWL) от других формальных языков - это наличие формальной семантики, для которой используется язык, отличной от языка описания синтаксиса. Этот язык лишенный неоднозначностей естественного языка, на котором описывают семантику, например,

языков программирование, и потому значения логических утверждений всегда строго определены (в отличие от неформального описания семантики). В связи с этим представляется целесообразным использовать OWL-DL для представления базы знаний когнитивной сети.

3.3. Обучение когнитивной сети

Обучение представляет собой пополнение или изменение базы знаний (БЗ) на основе накопленного опыта. В нашем случае предполагается, что БЗ когнитивной сети представлена в виде онтологии. Таким образом, процесс обучения КС представляет собой пополнение и усовершенствование такой онтологии. В процессе обучения онтология может быть пополнена новыми терминами, могут быть обнаружены новые связи между уже существующими терминами. Кроме того, онтология может быть пополнена новыми экземплярами уже существующих классов, а содержащиеся в ней экземпляры могут быть модифицированы или вообще удалены из онтологии.

Для пополнения онтологии существует несколько основных способов:

- ручное внесение изменений на основе знаний и опыта эксперта: осуществляется в каком-либо редакторе онтологий экспертом предметной области, однако этот процесс занимает много времени, сложен и дорог;
- автоматизированное пополнение онтологий: этот процесс также требует участия человека-эксперта, однако значительная часть работы может быть выполнена соответствующим программным обеспечением.

Автоматизированное пополнение онтологий может быть осуществлено на основе различных методов: 1) логических операций над онтологиями (интеграция, сопоставление, выравнивание и т.д.); 2) извлечения знаний из внешних информационных ресурсов (ИР) – естественноречевых документов, например, документация на новый тип предложенных услуг, сведения о провайдерах), структурированных метаданных, мультимедиа (например, рекламные ролики разных провайдеров); 3) индуктивного обобщения слабоструктурированных данных для построения правил предметной области (например, обработка потоков данных о месторасположении клиентов и провайдеров, качестве сигнала, объемах передаваемой информации).

Чтобы на основе онтологии улучшить работу системы, нужно извлечь из онтологии, содержащиеся в ней знания. Для выполнения логических запросов используют язык SPARQL, а для определения истинности того или иного утверждения – ризонеры (reasoners).

В целом, существует два подхода к реализации логического вывода: на базе правил (rule-based) – с использованием алгоритмов forward-chaining или backward-chaining и на основе семантического табло (semantic tableau). На базе правил реализованы,

например, Semantics-SDK и OwlIm, а на базе семантического табло – Pellet, FaCT++ и Hermi. Все эти ризонеры можно интегрировать в OWL API – фреймворк для работы с онтологиями (представление онтологии в памяти, возможность ее модификации на уровне объектов и т.п.), который уже де-факто стал стандартом, и в Jena. Базовые возможности Pellet следует рассматривать в свете: загрузки онтологии; логического вывода и выполнения SPARQL-запросов.

3.4. Информационная поддержка формирования базы знаний КС

Для того, чтобы система способна была обеспечить принятие эффективных решений, необходимо обеспечить поступление актуальных сведений об имеющихся ресурсах коммуникативных систем – сервисах, предлагаемых провайдерами, их качестве, доступности, стоимости и т.д., и об отзывах клиентов о реальном состоянии предлагаемых сервисов. Очевидно, что такие сведения регулярно меняются и требуют постоянного обновления. Источником такой информации могут быть ресурсы Web, доступ к которым и преобразование в онтологическую БЗ обеспечивает семантическая поисковая система. В качестве формального описания интересующей при поиске предметной области используется онтологическая БЗ когнитивной сети.

Мультиагентная информационно-поисковая система (МАИПС) с развитыми средствами интеллектуализации ее поведения, которая детально описана в [Рогущина, 2010] и [Гришанова, 2009], предоставляет пользователю высоко релевантные результаты поиска, которые достигаются благодаря:

- ориентации системы на пользователей, которые имеют в сети постоянные информационные интересы и требуют постоянного поступления соответствующей информации (поисковая система направлена на выполнение сложных многообразных запросов в довольно узких профессиональных областях);
- применению программных агентов, которые способны действовать в динамической среде, учась на собственном опыте;
- использованию БЗ предметной области, которая представлена в виде онтологии.

В созданной поисковой системе интегрированно используются:

- онтологии и тезаурусы ПрО;
- теоретико-множественные операции над тезаурусами;
- генерация тезаурусов по естественно-языковыми текстами;
- технологии Web 2.0 (облака тэгов – для визуализации поисковых тезаурусов; социальных сервисов – для взаимодействия между пользователями);

- технологии Semantic Web;
- оригинальные алгоритмы упорядочения информационных ресурсов, найденных системой, с учетом веса онтологических терминов;
- критерии оценки читабельности текста для поиска информации, которая отвечает персональным потребностям пользователя;
- методы индуктивного вывода для обобщения опыта работы поисковой системы;
- мультиагентный подход к созданию модели интеллектуальной поисковой системы;

Пользователь может обращаться к онтологиям, созданным другими пользователями – пересматривать их, задавать за ними контекст поиска, копировать из них нужные фрагменты, но не имеет права изменять их. ИПС может предусматривать поиск онтологий, которые содержат введенные пользователем термины, а также поиск онтологий, похожих на выбранную пользователем онтологию. Это позволяет создавать группы пользователей с общими информационными интересами и предотвратить дублирование в выполнении одинаковых многообразных запросов разных пользователей. Адекватным средством представления таких онтологий есть язык OWL.

ЗАКЛЮЧЕНИЕ

В данной работе представлены исследования, касающиеся когнитивных сетей как перспективной технологии для улучшения производительности современных гетерогенных беспроводных систем. Результатом исследования явилась модель управления когнитивной сети, основанной на управлении знаниями и использовании технологий Semantic Web. Предложенные в данной работе подходы к использованию семантических методов для улучшения обслуживания клиентов гетерогенных беспроводных сетей требуют дальнейшей доработки и развития.

Внедрение знание-ориентированной системы управления когнитивной сети сегодня вполне реально, поскольку базируется на ряде разработок авторов в области онтологического анализа, информационного поиска в неструктурированном распределенном пространстве и опытом реализации ряда прикладных информационных систем и приложений на основе семантических технологий.

Когнитивная сеть, состоящая из коллекции элементов (потенциальных ресурсов), посредством обучения, логического вывода и принятия решений, динамически адаптируется к изменяющимся сетевым условиям, с целью оптимизации непрерывной производительности в сети.

Однако проблема когнитивных сетей имеет множество нерешенных вопросов, которые требуют своего дальнейшего развития. В проанализированных нами работах решались часто только некоторые частные проблемы когнитивных

сетей, поэтому у нас была попытка интегрированного подхода с использованием знание-ориентированных технологий.

Мы выделяем проблемы и требования, которые мотивируют исследования и реализацию когнитивных сетей: сетевая сложность, беспроводная среда, цели конечного пользователя (переносного терминала) не удовлетворяют текущим подходам к архитектуре сети и недостаточно проработанными.

Сложность беспроводных сетей приводит к проблемам, которые нельзя решить путем использования локальных и реагирующих на какие-либо действия протоколов, а многоуровневый подход к сетевому стеку иногда препятствует тому, чтобы когнитивная сеть смогла достигнуть непрерывных целей. В связи с этим модель управления когнитивной сетью должна задействовать различные уровни протоколов и не должна быть жестко привязана к семиуровневой эталонной сетевой модели, но способна учитывать и обрабатывать семантику ПрО.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Alexiou, 2004] Alexiou Angeliki and Haardt Martin. Smart antenna technologies for future wireless systems: trends and challenges. IEEE Communications Magazine, 42(9):90–97, 2004.
- [Balamuralidhar, 2008] Balamuralidhar P. and Prasad R., "A context driven architecture for cognitive nodes", Wireless Personal Communications 45 (2008), pp. 423–434.
- [Bari, 2007] Bari F., Leung V.C.M., "Automated network selection in a heterogeneous wireless network environment", IEEE Network Magazine Volume: 21 Issue:1 2007, pp.34-40.
- [Bari,2006] Bari F. and Leung V. C. M., "Service Delivery over Heterogeneous Wireless Networks: Network Selection Aspects," Proc. ACM IWCMC, Vancouver, Canada, July 2006.
- [Gelenbe, 2001] Gelenbe E., Lent R., and Xu Z. Design and performance of cognitive packet networks. Performance Evaluation, 46(2-3):155–176, 2001.
- [Gladun, 2009] Gladun A., Rogushina J. Use of Semantic Web technologies in design of informational retrieval systems // in Book "Building and Environment", 2009 Nova Scientific Publishing, New-York, USA. – P.89-103.
- [Hongvan, 2003] Hongvan S., Chen H. and Lingge J., "Intelligent Signal Processing of Mobility Management for Heterogeneous Networks," Proc. Int'l Conf. Neural Networks and Signal Processing, Nanjing, China, Dec. 2003.
- [Kliazovich, 2010] Kliazovich D., Granelli F., Nelson Luis Saldanha da Fonseca N.L. Which Architectures for the Cognitive Networks of the Future?: ICST'S Global Community Magazine 09/25/2010, University Trento, Italy.
- [Le Bodic, 2000] Le Bodic G. et al., "Dynamic 3G Network Selection for Increasing the Competition in the Mobile Communications Market," Proc. IEEE VTS-Fall, 2000, Boston, MA, Sept. 2000.
- [Mitola, 2005] Mitola Joseph. Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio. PhD thesis, Royal Institute of Technology (KTH), 2005.
- [OWL, 2004] OWL Web Ontology Language, Reference, W3C Recommendation 10 February 2004. – <http://www.w3.org/TR/owl-ref/>
- [Ramming, 2004] Ramming C. Cognitive networks. In DARPA Tech, 2004.
- [RDF, 2002] RDF Model Theory, W3C Working Draft 29 April 2002. – <http://www.w3.org/TR/rdf-mt/>
- [Song, 2004] Song Q. and Jamalipour A., "Quality of Service Provisioning in Wireless LAN/UMTS Integrated Systems Using Analytic Hierarchy Process and Grey Relational Analysis," Proc. IEEE GLOBECOM, Dallas, TX, Nov./Dec. 2004.
- [SPARQL, 2008] SPARQL Query Language for RDF, W3C Recommendation, 2008, <http://www.w3.org/TR/rdf-sparql-query/>

[SWRL, 2004] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Member Submission 21 May 2004. – <http://www.w3.org/Submission/SWRL/>.

[Thomas, 2005] Thomas R.W., DaSilva L.A., MacKenzie A.B., "Cognitive Networks," Proc. IEEE DySPAN 2005, Nov. 2005, pp. 352–60.

[Thomas, 2006] Thomas R.W., Friend D.H., DaSilva L.A., MacKenzie A.B., Cognitive Networks: Adaptation and Learning to Achieve End-to-End Performance Objectives // IEEE Communications Magazine, December 2006, pp.234-242.

[Thomas, 2007] Thomas R.W. Cognitive Networks. – http://scholar.lib.vt.edu/theses/available/etd-07172007-150149/unrestricted/Thomas_CognitiveNetworksDiss4.pdf.

[Андон, 2008] Андон Ф.И., Гришанова И.Ю., Резниченко В.А. Semantic Web как новая модель информационного пространства интернет, Проблеми програмування. 2008. № 2-3, 417-430.

[Гладун, 2009] Гладун А.Я., Рогушина Ю.В. Использование технологии Semantic Web для интеллектуального управления в динамических распределенных системах // International Book Series "Information Sciences and Computing", 2009, Varna, Bulgarien. – P.143-153.

[Гладун, 2007] Гладун А.Я., Рогушина Ю.В., Штонда В.М. Онтологический анализ Web-сервисов в интеллектуальных сетях // Proc. of The XIII-th International Conf. "Knowledge-Dialogue-Solution", ITNEA, Sofia, V.2, 2007. – С.451-459.

[Гладун, 2008] Гладун А.Я., Рогушина Ю.В. Технологии Semantic Web и их использование при разработке интеллектуальных приложений // Проблеми програмування, № 2-3, 2008. – С.385-394.

[Гришанова, 2009] Гришанова И.Ю., Рогушина Ю.В. Компьютерная программа "Мультиагентная информационно-пошуковая система "МАПС"("МАПС") // Свідчення про реєстрацію авторського права на твір №32015

[Комашинский, 2011] Комашинский В., Мардер Н., Парамонов А. От телекоммуникационной к когнитивной инфокоммуникационной системе. – Технологии и средства связи №4, 2011.С.-67-75.

[Рогушина, 2010] Рогушина Ю.В., Гришанова И.Ю. Літературний твір наукового характеру "Модель мультиагентної інформаційно-пошукової системи "МАПС"("Модель МАПС"). – Свідчення про реєстрацію авторського права на твір №32068.

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

COGNITIVE NETWORKS AND THE ONTOLOGICAL ANALYSIS IN ADAPTIBILITY AND SERVICE REFINEMENT OF HETEROGENEOUS WIRELESS ENVIRONMENT

Gladun A. *, Rogushina J. **

* *International Research and Training Center of Information Technologies and Systems of National Academy of Sciences of Ukraine, Kiev, Ukraine*
glanat@yahoo.com

** *Institute of software systems of National Academy of Sciences of Ukraine, Kiev, Ukraine*
ladamandraka2010@gmail.com

This paper discusses the use of ontologies for knowledge representation of cognitive network that provides refinement of services in adaptive communication systems. Methods of this knowledge base completion are proposed.

Keywords: ontology, cognitive network, adaptive communication systems.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

МОДЕЛЬ ИНТЕРНЕТ-ПОРТАЛА НАЦИОНАЛЬНОГО ЦЕНТРА АНТАРКТИЧЕСКИХ ДАННЫХ

Глоба Л.С., д.т.н., проф., Кузин И.А., Мочалкина К.С., Новогрудская Р.Л.

*Национальный Технический Университет Украины «Киевский Политехнический Институт»,
г. Киев, Украина*

lgloba@its.kpi.ua

ksu_mochalkina@yahoo.com

ihkuzin@gmail.com

rinan@ukr.net

В статье представлен подход к построению интернет-портала знаний в области антарктических исследований. В качестве информационной модели портала используются онтологии.

Ключевые слова: интернет-портал, представление знаний, модель, онтология.

Введение

В настоящее время осуществляется большое количество исследований в Антарктике в различных научных направлениях. Однако существует проблема доступа к полученным результатам и их дальнейшей обработке. Важные результаты, как теоретические, так и практические, которые получают исследователи, остаются сосредоточенными лишь на уровне конкретных исследовательских институтов. Происходит процесс сосредоточения знаний, они становятся доступными узкому кругу ученых и специалистов. Поэтому актуальной задачей становится предоставление возможности использования этих знаний большому количеству исследователей, для которых они предназначены.

Согласно требованиям Резолюции 4.1 XXII АТСМ 1998 года все Стороны Договора об Антарктике, осуществляющих антарктические исследования, обязаны создать Национальные центры Антарктических Данных (НЦАД). Исходя из этих требований, пока актуальна задача по созданию такого Центра в Украине. Эффективное функционирование НЦАД поможет решить ряд проблем, связанных со сбором, хранением, обработкой и передачей, как первичных данных так и обработанных результатов антарктических исследований широкому кругу ученых и научным учреждениям в Украине, а также заинтересованным зарубежным научным организациям. Кроме этого, создание такого информационного пространства позволит Украине

полноценно войти в мировой антарктического научного сообщества, что повысит эффективность работы наших ученых и поднимет международный авторитет нашего государства.[Национальный Центр Антарктических Данных и конструкций, 2005]

Из всего вышесказанного следует необходимость построения качественной модели представления знаний на портале. В качестве модели представления знаний на портале использовались онтологии.

1. Онтология портала

1.1. Представление знаний в области антарктических исследований

Рассмотрим построение Интернетпортала в области антарктических исследований. Информационную основу портала 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 портала задает структуры для представления реальных данных и связей между ними. Использование онтологий для построения информационной основы портала 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 и данных по выбранной тематике, их семантическую связанность.

Онтологии состоят из следующих элементов: экземпляров, классов, атрибутов и отношений.

- Атрибуты – описывают классы по определенным характеристикам. Каждый атрибут имеет, по крайней мере, имя и значение, и используется для хранения информации, которая специфична для объекта и привязана к нему.

- Отношения – устанавливают связи (описывающие зависимости) между элементами онтологии. Обычно отношением является атрибут, значением которого является другой объект.

- Экземпляры – это основные, нижнеуровневые компоненты онтологии. Экземпляры могут представлять собой как физические объекты (люди, дома, планеты), так и абстрактные (числа, слова).

- Онтология представляет собой иерархию классов, связанных отношениями. Для построения онтологии обычно используются следующие 4 вида отношений:

- Ассоциативное отношение – это отношения задаваемые пользователем. Наличие таких отношений позволяет осуществлять содержательный поиск.

- Отношение часть-целое – это отношение позволяет устанавливать связи между классами на уровне иерархии. При поиске информации, связанной отношениями такого типа, осуществляется транзитивное замыкание.

- Отношение наследования – это отношение позволяющее осуществить передачу всех отношений и атрибутов от родительского класса дочернему.

- Отношение вида класс-данные – это отношение позволяющее связывать конкретные экземпляры понятий с классом.

Базируясь на описанных выше понятиях, построена онтология портала знаний в области антарктических исследований. Формально онтология может быть задана как: $O = \{C, A, R, T, F, D\}$, где C – множество классов, описывающих понятия предметной области; A – множество атрибутов, описывающих свойства понятий и отношений; R – множество отношений, заданных на классах: $R = \{R_{AS}, R_{IA}, R_n, R_{CD}\}$ – отношения следующих типов: R_{AS} – ассоциативное, R_{IA} – «часть-целое», R_n – наследования, R_{CD} – «класс-данные»; T – множество стандартных типов значений атрибутов; F – множество ограничений на значения атрибутов понятий и отношений; D – множество экземпляров классов [Загоруйко Ю.А. и др., 2007].

Определенная таким образом онтология может служить для представления понятий, необходимых

как для описания знаний в области антарктических исследований, так и для выполняемой в ее рамках обработки данных.

1.2. Онтология портала «Национального Центра Антарктических Данных»

Онтология портала в области антарктических исследований состоит из четырех онтологий: онтология внешнего пользователя, онтология антарктического знания исследователя и онтология предметной области (рис. 1). Онтология внешнего пользователя (ОВП) включает классы, относящиеся к организации доступа внешнего пользователя к материалам портала. Онтология антарктического знания (ОАЗ) содержит следующие метапонятия, задающие структуры для описания рассматриваемой предметной области. Онтология исследователя (ОИ) описывает возможности, предоставляемые исследователям на портале по доступу и обработке размещенных на портале материалов. Онтология предметной области (ОПО) отражает общие знания о предметной области, такие как иерархия классов понятий, семантические отношения на этих классах. [Глобал.С., и др., 2011]



Рисунок 1 – Онтология портала «Национальный Центр Антарктических Данных»

На (рис. 2) приведены классы ОВП, ОАЗ и ОИ, а также отношения заданные на этих классах. Классы отображены заштрихованной овальной областью которая расположена в определенном прямоугольнике, который соответствует той онтологии, к которой принадлежит данный класс.

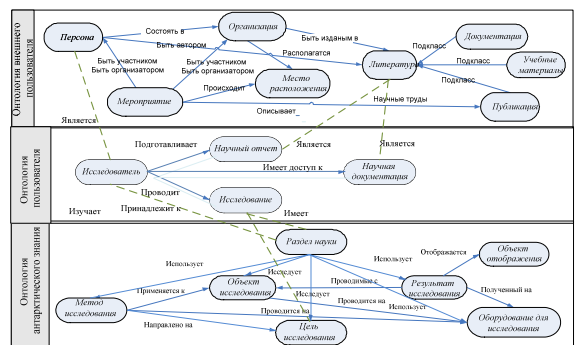


Рисунок 2 – Элементы онтологий портала

Для примера рассмотрим формальное описание онтологии внешнего пользователя, которая включает восемь общих классов понятий, относящиеся к организации научной деятельности, которые связаны отношениями различных типов. Различные свойства каждого понятия описываются на основе атрибутов понятий и ограничений, наложенных на область их значений. Исходя из этого ОВП может быть описана следующим образом:

$$O_1 = \{C_{o_1}, A_{o_1}, R_{o_1}, T_{o_1}, F_{o_1}, D_{o_1}\}, \text{ где} \quad (1)$$

$$C_{o_1} = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9\},$$

$$A_q = (A_{C_1}, A_{C_2}, A_{C_3}, A_{C_4}, A_{C_5}, A_{C_6}, A_{C_7}, A_{C_8}, A_{C_9}),$$

$$D_q = (D_{C_1}, D_{C_2}, D_{C_3}, D_{C_4}, D_{C_5}, D_{C_6}, D_{C_7}, D_{C_8}, D_{C_9}),$$

$$R_{o_1} = (R_{AS_1}(O_1), \dots, R_{AS_{10}}(O_1), R_{L_4}(O_1), \dots, R_{L_6}(O_1),$$

$$R_{n_1}(O_1), \dots, R_{n_6}(O_1), R_{CD_1}(O_1), \dots, R_{CD_3}(O_1)).$$

ОВП включает следующие классы понятий:

Персона - $C_1(O_1)$. К этому классу относятся понятия, связанные с субъектами научной деятельности: исследователями, сотрудниками и членами организаций. Атрибутами персоны являются: персональные данные, ученая степень, звание, направления научной деятельности, место проживания.

$$C_1(O_1) = (A_{C_1}, D_{C_1}). \quad (2)$$

Организация - C_2 . Понятия этого класса описывают различные организации, научные сообщества и ассоциации, институты, исследовательские группы и другие объединения. Атрибутами организации являются: название и место расположения.

$$C_2(O_1) = (A_{C_2}, D_{C_2}). \quad (3)$$

Мероприятие - C_3 . В этот класс входят понятия, описывающие научно-организационную или научно-исследовательскую деятельность – научные мероприятия, конференции, исследовательские поездки, проекты, программы и т.п. К атрибутам события относятся: название, место проведения, дата начала, дата окончания, степень завершенности.

$$C_3(O_1) = (A_{C_3}, D_{C_3}). \quad (4)$$

Публикация - C_4 . Этот класс служит для описания различного рода публикаций, (в периодических изданиях и тех, которые издаются в результате проведения конференций, научных мероприятий и т. д.). К атрибутам публикации относятся: название, описание, дата публикации и язык публикации.

$$C_4(O_1) = (A_{C_4}, D_{C_4}). \quad (5)$$

Местоположение - C_5 . Этот класс понятий позволяет описывать географическую и административно-территориальную локализацию объектов исследования, организаций и т.п. Атрибутами этого класса являются название местоположения и географический тип.

$$C_5(O_1) = (A_{C_5}, D_{C_5}). \quad (6)$$

Литература - C_6 . Этот класс служит для описания литературы (представленных в печатном или электронном форматах), которая используется в деятельности исследователей (монографии, статьи, отчеты, труды конференций, периодические издания, фото- и видеоматериалы и др.). К атрибутам публикации относятся: название, описание, дата публикации и язык публикации.

$$C_6(O_1) = (A_{C_6}, D_{C_6}). \quad (7)$$

Документация - C_7 . Этот класс служит для описания различного рода документации, такой как ГОСТЫ, ISO и т. д. К атрибутам документация относятся: название, описание, дата публикации и язык.

$$C_7(O_1) = (A_{C_7}, D_{C_7}). \quad (8)$$

Учебные материалы - C_8 . Этот класс содержит существующие учебники, справочники, учебные пособия, мануалы... К атрибутам класса относятся: название, описание, дата публикации и язык.

$$C_8(O_1) = (A_{C_8}, D_{C_8}). \quad (9)$$

1.3. Онтология предметной области

Онтология предметной области отражает общие знания о предметной области, такие как иерархия классов понятий, семантические отношения на этих классах. Основой онтологии предметной области для портала антарктических знаний в области антарктических исследований послужило постановление Кабинета Министров Украины.

Онтология предметной области (ОПО) описывает классификацию направлений исследований, которые проводятся на антарктической станции (рис. 3), в различных разделах науки и включает формальное и неформальное описание понятий и отношений между ними.

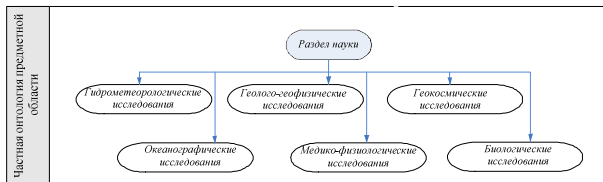


Рисунок 3 – Классификация направлений исследований НЦАД.

Эти понятия являются реализациями метапонятий онтологии Антарктического знания и могут быть упорядочены в иерархию общее-частное и часть-целое. Так, например, *Объектами* исследования соответствуют природные явления Антарктики такие как: рельеф дна и суши, наземные и подводные экосистемы, электромагнитное поле и др., а в качестве *Методов* исследования выступают такие понятия, как: сейсмографические исследования, зондирование почвы, прогнозирование погоды и др. Пример предметной области для геолого-геофизических исследований приведен на (рис. 4). [ГлобалС., и др., 2010]

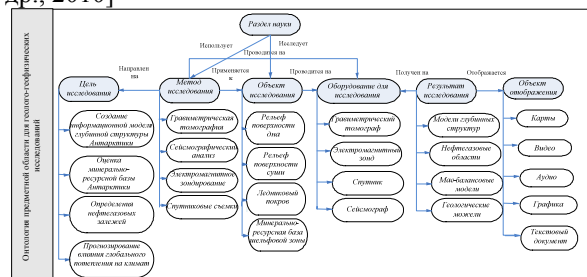


Рисунок 4 – Фрагмент модели предметной области.

2. Функции портала

Согласно рассмотренной выше онтологии портал НЦАД должен решать ряд функциональных задач:

- Качественное представление знаний на портале
- Систематизация и структурирование информации
- Формализация антарктических знаний
- Эффективный и содержательный поиск

Портал обеспечит доступ к базам данных, справочной литературе, мануалам, экспертной информации, ресурсам сети и т. пр. На портале возможна реализация просмотра блоков новостей, информации о конференциях (предстоящих и прошедших), конкурсах и грантах, а также другая информация о событиях, имеющих отношение к данной области знаний. Важным этапом построения портала является структуризация и систематизация информации и знаний в области антарктических исследований, что позволит конечному пользователю осуществлять просмотр и поиск конкретных сведений в рассматриваемой сфере. Структуризация и размещение на портале организуется удобным для конечного пользователя образом, реализуя проблемно-ориентированные

средства навигации и поиска по информационному пространству портала. При этом поиск информации организован так, что пользователь имеет возможность задавать запрос не только и не столько по ключевым словам, сколько в знакомых ему терминах предметной области портала (рис. 5).

Функционально Национальный Центр Антарктических Данных:

- создает Единое информационное пространство для представления результатов антарктических исследований украинских ученых в среде Интернет;
- обеспечивает единую точку доступа ко всем видам научных ресурсов;
- использует такие базовые подходы, как:
 - одновременность системной разработки, реализации и использования данных при обеспечении одновременной работы всех ученых;
 - стандартизацию всех типовых компонентов информационных процессов;
 - интеграцию как метод организации взаимодействия отдельных компонентов;
 - интеллектуализацию как метод вычислений необходимых аналитических и статистических данных.

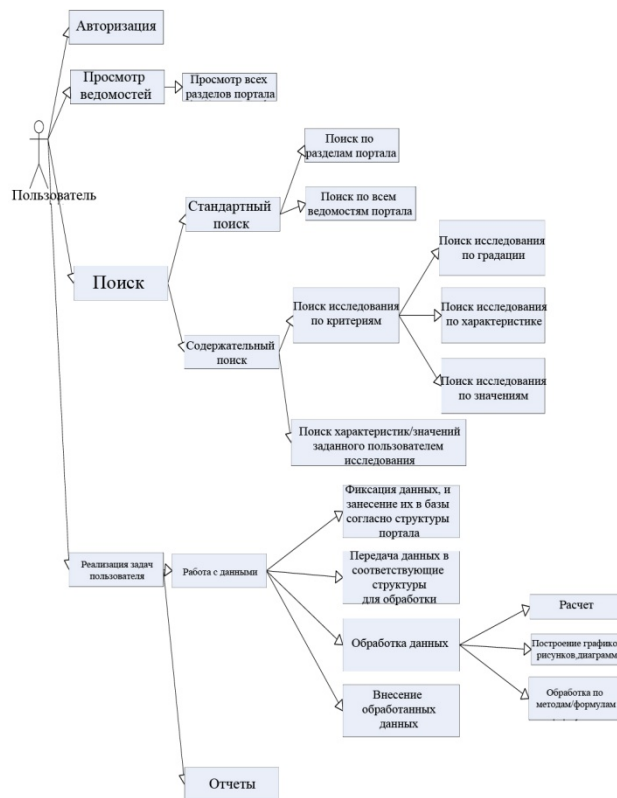


Рисунок 5 – Общая функциональная схема

Для поддержки процессов сбора, хранения, обработки и передачи данных необходимо учесть наличие таких составных частей единой информационной среды:

- необходимо учесть наличие таких составных частей единой информационной среды:
- портала, который используется как единая точка входа;
- единой сервисной шины для объединения всех сервисов хранения, обработки и представления информационных ресурсов антарктических исследований;
- базы метаданных и онтологий, которые задают структуры для представления реальных данных и связей между ними, позволяют не только целостно представить всю необходимую информацию о данных антарктических исследований, но и обеспечить их семантическую связанность;
- базы актуальных данных антарктических исследований; технической документации; мониторинга состояния оборудования; сохранения и анализа статистических данных;
- базы геоданных для отображения информации в картографическом виде;
- программы для сохранения, резервного копирования и восстановления информационных ресурсов;
- хранилища архивных ресурсов для сохранения исторических информационных ресурсов антарктических исследований в разрезе времени;
- средств и технологий интерактивного взаимодействия.

В процессе разработки программного и информационного обеспечения необходимо предусмотреть работу системы в следующих основных режимах:

- Режим обычного пользователя;
- Режим научного исследователя;
- Режим администрирования банка данных.

Режим обычного пользователя предусматривает доступ к уровню портала включающий общедоступную информацию, такую как описание деятельности НЦАД, направлений исследований, проводимых на станции Академик Вернадский и другие. Главное требование к режиму обычного пользователя - возможность доступа к portalу с любой рабочей станции.

Режим научного исследователя предполагает доступ к рабочим корпоративным материалам, научным результатам и другим необходимым данным только специалистам, работающим с данной информацией, в том числе и на Украинской антарктической станции Академик Вернадский. Главное требование к режимам научного исследователя в процессе функционирования системы - асинхронность, то есть, независимость одного от другого. Этот режим должен предоставлять пользователю возможность выполнять необходимые работы с учетом распределения доступа к базе данных и ее отдельных элементов. Вместе с тем предусматривается возможность доступа к

конкретным элементам базы данных, к более детализированной информации.

Режим администрирования банка данных должен включать в себя все виды работ, выполняемых обслуживающим персоналом для поддержки всех составных частей системы в работоспособном состоянии. Функционирование режима администрирования не должно мешать пользователям портала. Если этого избежать невозможно, необходимо предусмотреть переход пользователей в автономный режим работы или предупредить о временном прекращении работы пользователей с системой. Как правило, режим администрирования должен выполняться в нерабочее для большинства пользователей время. Режим администрирования системы должен обеспечивать выполнение необходимых действий связанных с хранением и защитой информации от возможного разрушения по техническим причинам, подключение новых пользователей, установка паролей и уровня доступа пользователей к базам данных и их составных частей и отдельных элементов и другие функции.

Система должна обеспечивать защиту информации от несанкционированного доступа с помощью системы паролей, определяющих уровень доступа к базе данных и ее составных частей

ЗАКЛЮЧЕНИЕ

Рассмотрена возможность построения портала «Национальный Центр Антарктических Данных». Информационную основу портала составляют онтологии, с помощью которой осуществляется систематизация и структуризация информации, организовывается эффективный поиск и навигация по информационному пространству портала Национального Центра Антарктических Данных. Также рассмотрены функции портала реализуемые на основании созданной онтологии.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Национальный Центр Антарктических Данных и конструкций, 2005] Национальный Центр Антарктических Данных и конструкций / Редкол.: В. Т. Троценко (отв. ред.) и др. – К.: Академперіодика, 2005. – 1088с.
- [Загорулько Ю.А. и др., 2007] Загорулько Ю.А., Боровикова О.И. Технология построения онтологий для порталов научных знаний / Загорулько Ю.А., Боровикова О.И. // Весник НГУ. Серия: Информационные технологии. - Том 5, выпуск 2. – С.12–15.
- [Глоба Л.С., и др., 2011] Глоба Л.С., д.т.н., проф., Новогрудская Р.Л. Модель интернет-портала «Прочность материалов» / Глоба Л.С., д.т.н., проф., Новогрудская Р.Л. // Международный научно-технический сборник «Надежность и долговечность машин и сооружений». - Київ; - 2011, стр. 183-189
- [Глоба Л.С., 2010] Глоба Л.С., д.т.н., проф., Новогрудская Р.Л. Подход к построению портала инженерных знаний / Глоба Л.С., д.т.н., проф., Новогрудская Р.Л. // ИАИ – 2010. Труды конференции. С53-62.

NATIONAL ANTARCTIC DATA CENTRE INTERNET-PORTAL MODEL

Globa L.S., Kuzin I.A., Mochalkina K.S.,
Novogradskaya R.L.

*National Technical University of Ukraine "Kiev
Polytechnic Institute", c. Kiev, Ukraine*

lgloba@its.kpi.ua

ksu_mochalkina@yahoo.com

ihkuzin@gmail.com

rinan@ukr.net

The article presents the approach for the designing of knowledge Internet portal in the field Antarctic researches. Ontology is used as portal informational model.

INTRODUCTION

There are currently a large number of studies in the Antarctic in various scientific fields. However, there is a problem of access to the results obtained and their further processing. Important results, both theoretical and practical remain focused only on the level of individual research institutes. Therefore, an urgent task is to provide opportunities to use this knowledge to more researchers, for whom they are intended.

According to the requirements of Resolution 4.1 XXII ATCM 1998 all Antarctic Treaty Parties engaged in Antarctic research have to establish National Antarctic Data Centre (NTSAD). From all of these points there is a need to build a qualitative model of knowledge representation on the portal. As a model of knowledge representation portal ontology is used.

MAIN PART

Recently, for the construction of scientific knowledge portals ontological model are being used. Ontology is a conceptual model, which helps in certain area's formalization. Introducing a formal description of domain concepts as classes of objects and their relationships, the ontology defines the structure of the portal to provide real data and relationships between them.

Ontologies consist of the following elements: instances, classes, attributes and relationships. Classes - these are abstract groups, sets or collections of objects, which include instances other classes, or the same combination and order of the two. Attributes - describe the classes of certain characteristics. Each attribute has at least the name and value, and is used to store information that is specific to the object and attached to it. Relationships - establish a connection (describing the dependence) between the elements of the ontology.

Ontology of National Antarctic Data Centre (NADC)portal consists of four ontologies: the ontology of the external user the ontology of Antarctic knowledge, the ontology of a researcher and the ontology of a subject area. The ontology of the external user (OEU) includes classes related to providing access to materials for an external user. Ontology of the Antarctic knowledge (OAK) contains the following metadata defining a framework for the description of the subject area. Ontology of the researcher (OR) describes the possibilities provided to researchers at the portal to access and processing of materials posted on the portal. Ontology of a subject area (OSA) reflects the general domain knowledge, such as class hierarchy of concepts, semantic relationships for these classes.

According to the discussed above ontology portal NADC must solve a number of functional problems:

- Quality of knowledge representation in the portal
- Organizing and structuring of information
- The formalization of Antarctic knowledge
- Effective and meaningful search

The portal will provide access to databases, reference books, manuals, express information, resources, networks, etc., etc. In the portal can be implemented blocks viewing of news, information about conferences (upcoming and past), competitions and grants, as well as other information about events relevant to this area of expertise. An important step in the construction of the portal is the structuring and systematization of information and knowledge in the field of Antarctic research, which will allow end users to browse and search for specific information in this field. Structuring and placement on the site is organized for the end user friendly way by implementing problem-oriented navigation tools and search the information space portal. The search information is organized so that the user can specify the request is not only and not by words, but in terms familiar to him the domain portal.

CONCLUSION

The possibility of building a portal "National Centre for Antarctic Data" was discussed. Portal's information base are ontologies, which helps in systematizing and structuring of information, helps to organize an effective search and navigation in information space of the National Antarctic Data Centre portal. The portal's functions implemented on the basis of described ontology are presented.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.032.26

**ПОСТРОЕНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ
ФОРМИРОВАНИЯ ИНДИВИДУАЛЬНОЙ ТРАЕКТОРИИ
ИЗУЧЕНИЯ СТУДЕНТОМ
НЕКОТОРОЙ ПРЕДМЕТНОЙ ОБЛАСТИ ЗНАНИЙ
НА ОСНОВЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ ТЕОРИИ АДАПТИВНОГО РЕЗОНАНСА**

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 индивидуальной траектории изучения студентом некоторой предметной области знаний с помощью искусственной нейронной сети ART.

Ключевые слова: вектор параметров оценки качества усвоения знаний, декартовое произведение векторов, нейронный слой сравнения, нейронный слой распознавания, интеллектуальная обучающая система, модуль адаптации.

ВВЕДЕНИЕ

Актуальным с точки зрения личностно ориентированного обучения является понятие индивидуальной траектории обучения и изучения. Формирование индивидуальной траектории изучения является задачей со множеством неопределенных параметров, поэтому предлагается для ее решения применять искусственную нейронную сеть. Предлагается схема организации процесса обучения, с применением искусственных нейронных сетей, построенных на основе теории адаптивного резонанса (сети ART), описывается процесс функционирования системы. Ядром предлагаемой информационной обучающей системы, влияющим на построение траектории обучения, является индивидуальная траектория изучения студентом некоторой предметной области знаний, которую можно представить в виде декартового произведения векторов. Введение в структуру информационной обучающей системы модели индивидуальной траектории изучения студентом некоторой предметной области знаний, которая включает личностные и профессиональные качества студента, а также нейронных сетей, которые классифицируют обучаемых и определяют очередной этап траектории обучения, позволяют индивидуализировать и дифференцировать процесс обучения.

Интеллектуализация индивидуальной траектории изучения предметной области знаний на основе искусственной нейронной сети теории адаптивного резонанса

Реализации парадигмы индивидуально-личностного обучения предполагает применение двух основных подходов: педагогического и информационного. Педагогический подход основан на необходимости реализации в учебном процессе различных дидактических целей (характер представления окружающей действительности, организация разнообразных видов учебно-познавательной деятельности, осуществление мотивационных, учебно-воспитательных и контрольно-корректирующих функций и т.п.). Информационный подход направлен на создание своеобразной обучающей среды, в которой при использовании определенных педагогических технологий происходит процесс познания и интеллектуального развития. Информационный подход предполагает существенную перестройку образовательной технологии, направленную на нейтрализацию таких отрицательных последствий обучения в условиях классно-урочной системы, как недостаточно развитая вариативность образования, слабый учет индивидуальных способностей, творческого потенциала и личных интересов

обучаемых.

Наиболее конструктивной альтернативой, на наш взгляд, является интеграция этих подходов в педагогическую технологию, основанную на применении информационно-коммуникационной предметной среды (по Роберт И.В.), обеспечивающей индивидуальный выбор обучаемыми интерактивного режима работы с учебной информацией, ее изучение и закрепление в индивидуальных и групповых формах.

Средству обучения, функционирующему на базе информационных и коммуникационных технологий, при необходимости (по Роберт И.В.) можно частично передать функции обучающего: контроль результатов обучения; предоставления заданий, адекватных уровню обучающегося; тренировки на формирование умений, навыков; сбор, обработку, хранение, передачу информации; тиражирование, управление учебной деятельностью; обеспечение коммуникационных процессов; организацию разнообразных форм деятельности по самостоятельному извлечению и представлению 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.

На основе анализа научной литературы по рассматриваемой проблеме [Козлов, 1999], [Роберт, 2009], [Ширшов, 2002], [Brusilovsky, 1993], [Corbett и др., 1992] индивидуальную траекторию изучения студентом некоторой предметной области знаний можно представить в виде декартового произведения векторов

$$M = VDE \times VSU \times VPS \times VSO \times VAD \times VPD \times VPU \quad (1)$$

В этой модели

$VDE = (de_1, de_2, de_3, \dots, de_k)$ – вектор предъявляемых дидактических единиц $(de)_t$, из базы знаний учебной дисциплины, $(t = 1, 2, \dots, k) \in K$,

где K – число дидактических единиц для изучаемого модуля знаний.

$VSU = (s_1, s_2, s_3, \dots, s_n)$ – вектор состояний обучающегося, ученика s_j в процессе изучения дидактической единицы, $(j = 1, 2, \dots, n) \in N$,

где N – число возможных состояний обучающегося в процессе изучения дидактической единицы.

$VPS = (ps_1, ps_2, ps_3, \dots, ps_z)$ – вектор элементов педагогического сценария обучения ученика $(ps)_i$; усвоению дидактического элемента,

$$(i = 1, 2, \dots, z) \in Z,$$

где Z – число возможных элементов - форм, видов организации занятия ученика с

дидактическим элементом.

$VSO = (o_1, o_2, o_3, \dots, o_b)$ – вектор параметров оценки качества усвоения знаний o_l дидактического элемента обучающимся, $(l = 1, 2, \dots, b) \in B$,

где B – число оцениваемых параметров (знаний, умений, навыков, представлений) по данному дидактическому элементу.

$VAD = (ad_1, ad_2, ad_3, \dots, ad_c)$ – вектор адаптационных действий $(ad)_f$, предпринятых средствами ИКТ для адаптации педагогического сценария к текущему состоянию обучаемого, $(f = 1, 2, \dots, c) \in C$,

где C – число допустимых операций адаптации педагогического сценария для дидактического элемента.

$VPD = (pd_1, pd_2, pd_3, \dots, pd_h)$ – вектор процедур управления $(pd)_h$ процессом усвоения дидактического элемента учеником, сформированный средствами ИКТ, $(h = 1, 2, \dots, q) \in Q$,

где Q – число допустимых процедур управления процессом усвоения ученика (предъявление дидактического элемента; контроль усвоения дидактического элемента; контроль состояния ученика; оценка отклонения от педагогического сценария процедур управления, сформированных учеником), сформированных средствами ИКТ.

$VPU = (pu_1, pu_2, pu_3, \dots, pu_r)$ – вектор процедур управления $(pu)_v$ средствами ИКТ, сформированный учеником в процессе изучения дидактического элемента, $(v = 1, 2, \dots, r) \in R$,

где R – число допустимых процедур управления (просмотр, самоконтроль, контроль – тестирование, поиск, пропуск), предусмотренных педагогическим сценарием на средствах ИКТ.

В этом случае индивидуальную траекторию изучения студентом t -го дидактического элемента модуля знаний можно представить как 7-арное отношение множества векторов

$$m_{de} = ((de)_t, s_j, (ps)_i, o_l, (ad)_f, (pd)_h, (pu)_v) \quad (2)$$

В условиях реализации средств ИКТ на базе достижений нанотехнологии, nanoиндустрии (многоядерные процессоры, нейропроцессоры) можно говорить о распараллеливании процессов в информационно-коммуникационной предметной среде, и как следствие такого подхода, о применении искусственных нейронных сетей в построении информационно-коммуникационной предметной среды и построении учебного информационного взаимодействия на основе интеллектуальных обучающих систем.

Индивидуальная траектория изучения студентом, в этом случае, должна отражать учебное информационное взаимодействие, описанное в терминах искусственных нейронных сетей.

Все большее понимание получает мысль о том, что эффективное обучение невозможно без систематического применения методов извлечения, обработки и систематизации знаний, что характерно для области использования технологий искусственного интеллекта.

Рассматривая, каким образом человек в процессе обучения с помощью компьютера может

анализировать поступающую информацию, в терминологии нейрогенетики введено ключевое понятие — нейросеть. Именно совокупность нейросетей образует отделы нервной системы человека, которые в свою очередь определяют всю деятельность, придают существу разум, интеллект. Простейший нейрон – это преобразователь данных, у которого на входе один сигнал, а на выходе – значение функции. Искусственные нейронные сети демонстрируют удивительное число свойств, присущих мозгу. Они «обучаются» на основе опыта, «обобщают» предыдущие прецеденты при анализе новых данных и извлекают существенные свойства из поступающей информации, содержащей излишние данные.

Индивидуальная траектория изучения студента выражает цель обучения и содержит информацию о состоянии знаний обучаемого, которые разделяются на общие (интегрированные) и частные, отражающие усвоение текущего материала.

В интегрированные характеристики обучаемого выделяют рейтинг ученика, среднее время решения задачи, оценку по пройденному разделу, ошибки по разделам с частотой их возникновения, а также ряд других.

Для оптимизации ведения диалога в траекторию изучения добавлен текущий сценарий диалога, содержащий имена и тип предлагаемых учебных фрагментов, а также протокол выдачи информации, представляющий собой список пройденного материала с указанием времени его выдачи. Последний введен чтобы избежать дублирования при добавлении новых фрагментов в сценарий диалога.

Единая база данных (БД) учебного материала ведется на машине преподавателя. Приложения же обучаемых предоставляют сервис по ведению данных о текущих характеристиках ученика. Запрос новых данных от БД учебного материала является инициатором изменения интегрированных характеристик, которое, в свою очередь, требует запроса к серверу, ведущему данные об ученике, т.е. приложению обучаемого. Как становится очевидным, данные предоставляются клиентскому приложению по явному запросу. Интегрированные характеристики обучаемого участвуют только при формировании новых учебных воздействий и, также ведутся приложением преподавателя.

С целью оценки начальных знаний по выбранному или заданному учебному материалу применяют 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.

Дилемма стабильности - пластичность является важной особенностью обучения, необходимо

обучать сеть, моделирующую студента, 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.

Недостатком сети ART является чувствительность к порядку предъявления входных векторов. Данный недостаток устраняется в модели обучения на основе элементов педагогического сценария обучения студента усвоению дидактического элемента. Педагогический сценарий обучения студента определяет порядок предъявления входных векторов.

Другой недостаток сети ART – невозможность классификации зашумленных векторов, то есть векторов с недостоверной информацией, устраняется в модели обучения с помощью адапционных действий, предпринятых средствами ИКТ для адаптации педагогического сценария к текущему состоянию обучаемого, и параметров оценки качества усвоения, компонентов дидактического элемента, обучающимся.

Таким образом, нейросетевая модель процесса изучения общеобразовательного (учебного) предмета (предметной области) может быть реализована на архитектуре искусственной нейронной сети ART – сети адаптивного резонанса. Структура сети представлена на рисунке 1.

Она включает два слоя нейронов: слой сравнения и слой распознавания.

Модуль сброса обеспечивает управляющие функции, необходимые для обучения и классификации.

Входной вектор X , выходной вектор C , модифицирующий вектор R , приемники 1 и 2 обеспечивают формирование векторов рассогласования $G1$ и $G2$.

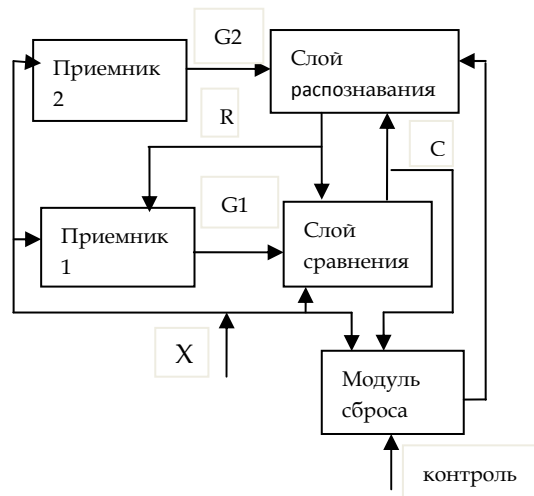


Рисунок 1- Структура сети ART

Нейросетевая модель процесса изучения общеобразовательного (учебного) предмета (предметной области), отображаемая этой структурой, содержит в качестве:

- входного вектора X представленного вектором VSO или VPS . Вектор VSO сравнивается первоначально с образом VPS , затем с одним из образов VSO , запомненных ранее. Свое решение сеть ART выражает в форме возбуждения одного из нейронов распознающего слоя. Степень схожести (критерий сходства) VSO с VPS закладывается в вектор VPS . VDE – вектор предъявляемых дидактических единиц, из базы знаний учебной дисциплины; VPS – вектор элементов педагогического сценария обучения студента усвоению дидактического элемента;

- вектора контроля VSO , вектора параметров оценки качества усвоения знаний дидактического элемента студентом;

- вектора R , выходного вектора слоя распознавания, формируется с помощью VPD – вектора процедур управления процессом усвоения дидактического элемента студентом, сформированного средствами ИКТ, который модифицирует вектор VPS ;

- вектора C - выходного вектора слоя сравнения векторов VSO с VPS , является модифицированным вектором VPS , который принимается на следующих фазах работы слоя сравнения в качестве вектора VSO ;

- вектора $G1$, представленного VSU – вектором состояний студента в процессе изучения дидактической единицы, является подмножеством вектора VSO . Единичные значения элементов вектора VSU определяют положительный настрой студента на работу с дидактическим элементом;

- вектора $G2$, представленного вектором VPU – вектором процедур управления средствами ИКТ, сформированного студентом в процессе изучения дидактического элемента, является подмножеством вектора VSO . Единичные значения вектора VPU отражают операторскую работу студента со средствами ИКТ.

Модуль сброса измеряет сходство между векторами VSO и VPS , если это отличие сильнее, чем требует критерий сходства, вырабатывается сигнал сброса возбужденного нейрона в слое распознавания и формируется вектор VAD . VAD – вектор адаптационных действий, предпринятых средствами ИКТ для адаптации педагогического сценария к текущему состоянию обучаемого, поданным на вход, формируют на выходе из запомненных ранее векторов вектор VSO , наиболее похожий в некотором выбранном смысле на данный входной вектор VDE .

Для формирования педагогического сценария обучения студента дидактической единице собирают точки зрения экспертов – какие наборы показателей в зависимости от выбранной стратегии необходимо включать в модель. Далее происходит обучение сети и генерирование вариантов обучения в зависимости от той или иной стратегии. Работа нейросети состоит в преобразовании входного

вектора в выходной вектор.

Такое преобразование задается весами сети. Точки зрения экспертов являются входными данными (векторами) для нейросети.

По мере ввода данных осуществляется обучение сети (корректировка весовых коэффициентов).

Результат работы сети – стратегия обучения студента, которой соответствует определенный набор показателей.

После многократного предъявления примеров веса сети стабилизируются. В таком случае делается вывод, что сеть обучена. Важно отметить, что качество обучения сети напрямую зависит от количества примеров в обучающей выборке, а также от того, насколько полно эти примеры описывают данную задачу. После того как сеть обучена, система приобретает возможность генерировать варианты наборов показателей в зависимости от того, какую стратегию обучения выбрал студент.

Формирование индивидуальной траектории обучения студента на основе нейросети состоит из следующих этапов:

- отбор показателей;
- обучение нейросети на основе экспертных оценок;
- формирование модели студента в соответствии со стратегией обучения;
- формирование стратегической карты;
- оценка эффективности обучения студента на основе нейронной сети;
- мониторинг статистических данных результатов обучения.

Формирование траектории индивидуального изучения студентом дидактической единицы происходит на основе значений личностно ориентированной модели студента. Она включает как психофизиологические характеристики обучаемого, так и характеристики его текущего уровня знаний. К функциям модуля формирования модели относятся проведение первоначального психофизиологического тестирования, определение начального уровня знаний по учебному курсу и собственно формирование модели. На основе знаний модели обучаемого нейронная сеть, как управляющий элемент информационной системы, строит обучающую стратегию.

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 единице 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.

Таким образом, интеллектуальную информационную обучающую систему можно использовать для управления процессом обучения; с целью формирования индивидуальной траектории обучения; автоматизации процессов контроля и коррекции результатов учебной деятельности.

ЗАКЛЮЧЕНИЕ

Интеллектуальная информационная система организация учебного процесса на основе искусственных нейронных сетей позволяет формировать и управлять индивидуальной траекторией изучения студентом некоторой предметной области знаний, позволяет сохранять, анализировать и объединять обычно разрозненные, субъективные мнения экспертов с помощью нейронной сети.

В данной статье предложен метод интеллектуализации индивидуальной траектории изучения студентом некоторой предметной области знаний, основанный на декартовом произведении векторов, описывающих состояние и поведение студента во время изучения дидактической единицы модуля знаний, что позволяет применить для обработки этих векторов искусственные нейронные сети

Интеллектуальная информационная система способна формировать индивидуальную траекторию обучения студента в соответствии со спецификой отдельных показателей обучения студента, оценивать эффективность работы студента в целом.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Барский, 2004] Барский А.Б. Нейронные сети: распознавание, управление, принятие решений. – М.: Финансы и статистика, 2004.

[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.

[Роберт, 2009] Роберт И.В. Теория и методика информатизации образования (психолого-педагогический и технологический аспекты), 2-е издание, дополненное.– М.: ИИО РАО, 2009.

[Хайкин, 2008] Хайкин Саймон. Нейронные сети: полный курс, 2-е издание: Пер. с англ. – М.: Издательский дом «Вильямс», 2008.

[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

CONSTRUCTION OF INTELLIGENT INFORMATION SYSTEMS THE FORMATION OF AN INDIVIDUAL TRAJECTORY EXAMINE THE STUDENT SOME DOMAIN KNOWLEDGE ON THE BASIS OF ARTIFICIAL NEURAL NETWORK THEORY OF ADAPTIVE RESONANCE

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 an individual trajectory of study of the student of a certain domain of knowledge with the help of artificial neural network ART. Key words: vector of parameters for assessing the quality of mastering of knowledge, декартовое произведение векторов, a neural layer of comparison, a neural layer recognition, intelligent training system, module adaptation

INTRODUCTION

Topical from the point of view of personality-oriented education is the notion of individual trajectory of teaching and learning. The formation 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.

Proposed scheme of the organization of the training process, with the use of artificial neural networks, built on the basis of the theory of adaptive resonance network (ART), describes the process of functioning of the system. The core of the proposed awareness training system, affecting the construction of the learning path, is the individual trajectory of study of the student of a certain domain of knowledge, which can be presented in the form of декартового произведения векторов. Introduction in structure of information learning system model of individual trajectory of study of the student of a certain domain of knowledge, which includes personal and professional qualities of the student, as

well as neural networks, which are classified by the trainees and define the next stage of learning path, allow you to individualize and differentiate the learning process.

MAIN PART

Individual trajectory of training reflects a unique way, by which the personality of the trainee should move to the objectives of the training. Choice of one of several tens and hundreds of possible learning paths generates the individual trajectory of study of the student of a certain subject area knowledge. One of the variants of solving this problem are neural network technologies.

Individual trajectory of study of the student of the some domain knowledge can be represented in the form of works vectors: vector of the requirements of the didactic units, vector states of the student, the student in the process of studying the didactic unit, vector elements pedagogical scenario pupil learning, vector of parameters for assessing the quality of mastering of knowledge didactic element of the trainees, vector of adaptation action (to adapt pedagogical scenario to the current state of the learner), vector management procedures the process of learning didactic element disciple, which was formed by means of ICT, vector management procedures (pu) means of ICT, which was formed by the student in the process of study of didactic element.

The integrated characteristics of the learner allocate rating of the student, the average time of solution to the problem, the assessment of the head of the section, the error on the sections with the frequency of their occurrence, as well as a number of others.

For the analysis of errors arising from the is database error of the learner, with the indication of the weight factor errors, filled the author of the course. When this task is divided into subtasks (to specify), and the appearing point of n-dimensional ball errors are analyzed system. This organization helps to avoid accidental coincidence user response with the right answer.

In the context of implementation of ICT on the basis of the achievements of nanotechnologies and nanoindustry (multi-core processors, нейропроцессоры) you can talk about paralleling processes in the information-communication of the subject environment, and as a consequence of this approach, the application of artificial neural networks in building information and communication of the subject environment and the construction of the school of information interaction on the basis of intelligent tutoring systems, the neural network model of the process of the study of education (training) of the object (domain) can be implemented on the architecture of artificial neural network ART - network adaptive resonance.

It consists of two layers of neurons: a layer of comparison and a layer of recognition. The input

parameters for assessing the quality of mastering of knowledge of didactic element student is compared with the image of the vector elements pedagogical scenario of a student learning didactic element, then an image of the VSO, memorized earlier. His decision to the network of ART expressed in the form of excitation of one of neurons распознающего layer. The degree of similarity (the criterion of similarity) is characterized teacher

For the formation of a pedagogical scenario of a student of the didactic unit collect the point of view of experts - what sets of indicators depending on the chosen strategy should be included in the model. Further there is a training network and the generation of training options depending on this or that strategy. The work of neural network consists in the transformation of the input vector in the output vector.

Such a transformation is weights of the network. The point of view of experts are the input data (vectors) for neural networks.

The formation of the individual trajectory of the students training on the basis of neural network consists of the following stages:

- selection of indicators;
- training of neural network on the basis of expert assessments;
- forming a model of the student in accordance with the strategy of training;
- monitoring of statistical data of the results of education.

Parameters of the renewed model of the learner are sent to the input of the neural network, which by analogy with the work of the human brain, can choose one of several hundred versions of the next stage of individual learning path.

Conclusion

Intellectual information system of organization of educational process on the basis of artificial neural networks allows you to create and manage individual trajectory of the student studied some domain knowledge, allows you to store, analyze and combine usually scattered, subjective opinions of experts with the help of neural networks.

In this article the method of intellectualization of individual trajectory of study of the student, based on the rectangular product of the vectors that describe the state and behavior of a student during the study of didactic unit module the knowledge that allows you to apply for the processing of these vectors artificial neural networks.

Intellectual information system is capable of forming individual trajectory of a student in accordance with the specifics of the individual indicators of a student, to evaluate the effectiveness of the work of the student as a whole.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.82:004.55:007.52

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ИНТЕРАКТИВНЫХ МЕТОДОВ ОБУЧЕНИЯ ИНОСТРАННОМУ ЯЗЫКУ: ПОСТАНОВКА ЗАДАЧИ

Елисеева О. Е., Ерченко Е. А.

*Белорусский государственный университет,
г. Минск, Республика Беларусь*

olae@open.by
zheton_88@mail.ru

Работа посвящена рассмотрению интерактивных методов обучения языку с использованием компьютерных программ. В настоящее время это один из наиболее эффективных способов повышения интереса и мотивации к самостоятельной познавательной деятельности. Данное направление интересно также и тем, что здесь еще предстоит найти новые методы и средства для реализации интеллектуальных систем обучения, в которых в качестве полноценного собеседника и учителя выступает сама система.

Ключевые слова: интерактивное обучение, изучения языка, автоматический анализ текстов, распознавание речи, когнитивная графика.

ВВЕДЕНИЕ

Изучение иностранного языка требует от учащихся активной самостоятельной работы. Основными видами учебной деятельности при изучении языка являются следующие:

- Изучение новых слов, постоянное расширение словарного запаса
- Усвоение правил словообразования, принятого в том или ином языке (как правило, это образование множественного числа, падежных форм, временных форм глаголов и др.)
- Усвоение грамматических правил построения словосочетаний и предложений
- Формирование базовых речевых навыков – произношение отдельных звуков, слов, фраз
- Формирование навыков ведения беседы, рассказа, монолога и пр.
- Формирование навыков чтения
- Формирование навыков грамотного письма.

Известно, что наиболее эффективно процесс усвоения иностранного языка происходит в процессе общения. В идеале это общение изучающего язык с носителем языка. Однако такая возможность существует далеко не всегда. Поэтому к настоящему времени разработано множество всевозможных методик обучения иностранному языку, способствующих повышению эффективности указанного процесса. Большинство эффективных методик подразумевает взаимодействие (общение) учащегося с учителем либо учащихся друг с другом.

Только в процессе такого взаимодействия возможно приобретение перечисленных выше навыков. К сожалению, при этом преподавателю зачастую приходится неоднократно повторять для учащихся одни и те же простые и известные для него слова, фразы, тексты и пр. В таких условиях преподаватель языка практически лишен возможности творчески мыслить. Поэтому очевидной становится целесообразность использования вычислительной техники со средствами мультимедиа. Однако в этом случае учащийся зачастую становится пассивным созерцателем, нежели участником учебного процесса. В связи с этим эффективность обучения иностранному языку падает.

Учитывая сказанное, становится понятным, что необходимо применение неких «особых» методов обучения, вовлекающих учащихся в **активную познавательную деятельность**. Одним из таких востребованных на сегодняшний день подходов является интерактивное обучение.

В данном сообщении систематизируются интерактивные методы обучения и анализируются существующие на сегодняшний день программные решения, реализующие либо поддерживающие соответствующие интерактивные методы.

1. Интерактивное обучение

1.1. Типология интерактивных методов обучения

Интерактивным называется такое обучение,

которое основано на взаимоотношении и взаимодействии [Кашлев, 2005]. Это, как правило, группа взаимодействующих учащихся, которые, обсуждая вопросы, спорят и соглашаются между собой, стимулируют и активизируют друг друга. При применении интерактивных методов на интеллектуальную активность сильнее всего действует дух соревнования, состязательности, который проявляется, когда люди коллективно ищут решение задач. Действует и такой психологический феномен, как заражение, и высказанная соседом мысль способна произвольно вызвать собственную аналогичную или противоположную. В процессе решения той или иной учебной задачи учащиеся вовлечены в активную **деятельность**, которая заставляет их применять и закреплять на практике имеющиеся знания и более целенаправленно, мотивированно и заинтересованно приобретать новые.

Технологию интерактивного обучения можно определить как совокупность способов целенаправленного усиленного межсубъектного взаимодействия педагога и учащихся, последовательная реализация которых создает оптимальные условия для их развития [Кашлев, 2005]. С помощью интерактивного обучения активизируется самостоятельная целенаправленная познавательная деятельность учащихся.

На сегодняшний день в педагогике известно много различных интерактивных методов обучения. Рассмотрим те, которые применимы в обучении языкам [Фопель, 1998], [Якубовская, 2006]:

- эвристическая беседа;
- метод дискуссии;
- «мозговая атака»;
- метод «круглого стола»;
- метод «деловой игры»;
- конкурсы практических работ с их обсуждением;
 - работа в микро-группах (малых группах);
 - кейс-метод (разбор ситуаций);
 - технология Open Space (Открытое пространство);
 - технология Varsamp;
 - метод проектов.

Эвристическая беседа – это метод, направленный на умелую постановку вопросов, которые в итоге заставляют учащихся, опираясь на свой собственный опыт и знания, путем логических рассуждений, самостоятельно прийти к правильным выводам. Как правило, эвристическая беседа начинается с простого сообщения определенных фактов по теме, описания явлений, событий, в некоторых случаях демонстрируются фильмы. Эвристическое обучение способствует развитию творческого, креативного, логического мышления.

Метод дискуссии [Оганесян, 2002] или метод «активного диалога» представляет собой групповое взаимодействие учащихся, обсуждающих заранее заданную проблему. Обсуждаемые проблемы и

основные вопросы для обсуждения должны быть заранее известны всем участвующим в дискуссии учащимся.

Часто метод дискуссии реализуется в форме **«мозговой атаки»** (мозгового штурма, brainstorming) – это оперативный метод решения проблемы на основе стимулирования творческой активности, при котором участникам обсуждения предлагают высказывать как можно большее количество вариантов решения, в том числе самых фантастических. Затем из общего числа высказанных идей отбирают наиболее удачные, которые могут быть использованы на практике. При изучении языка метод «мозгового штурма» можно использовать, например, как предварительный этап перед переходом к изучению новой темы, когда учащимся предлагается сначала попробовать «изобрести» что-то новое.

Метод «круглого стола» также является одной из форм реализации метода дискуссии, с тем отличием, что в обсуждении принимают участие специально приглашенные специалисты разного профиля (профессии). Перед каждой такой встречей преподаватель предлагает учащимся выдвинуть интересующую их тему и сформулировать вопросы для обсуждения. При обучении иностранному языку таким приглашенным специалистом может являться носитель изучаемого языка.

Метод «деловой игры» является одним из известных методов активного обучения, который применяется практически во всех сферах профессионального обучения. При реализации данного метода в процессе обучения языку учащиеся распределяются по ролям в рамках различных коммуникативных ситуаций и моделируют таким образом диалоги различного вида, закрепляя речевые навыки. В качестве основы «деловой игры» могут также выступать различные художественные произведения, по сюжетам которых учащиеся разыгрывают «сценки».

Конкурсы практических работ с их обсуждением – метод, в чем-то схожий с рассмотренными выше методами и заключается в том, что учащиеся сначала выполняют некоторые практические работы, которые затем коллективно обсуждаются. Данный метод очень часто применяется для работы над ошибками. Так как метод в своей основе имеет конкурс, то это дополнительный стимул соревновательного типа.

Работа в микро-группах (малых группах) отличается от группового обучения и предполагает работу 3-5 человек одновременно. Данный метод применяется как форма реализации других методов и предназначен для повышения индивидуализации обучения.

Кейс (Case study) - метод анализа ситуаций. Суть его в том, что учащимся предлагают осмыслить реальную жизненную ситуацию, описание которой одновременно отражает не только какую-либо

практическую проблему, но и актуализирует определенный комплекс знаний, который необходимо усвоить при разрешении данной проблемы. Будучи интерактивным методом обучения, он завоевывает позитивное отношение со стороны учащихся, которые видят в нем игру, обеспечивающую овладение практическим использованием материала. Анализ ситуаций воздействует на профессионализацию учащихся, формирует интерес и позитивную мотивацию по отношению к учебе [Современные технологии обучения, 2011]. Метод анализа ситуаций уместен при формировании профессионально-ориентированного владения языком в той или иной практической сфере деятельности.

Технология «Открытое пространство» (Open Space) – это методика проведения конференций и встреч, которая позволяет индивидам и группам эффективно взаимодействовать и принимать коллективные решения. Изобретатель технологии - Харрисон Оуэн [Интерактивные методы обучения, 2011]. Технология «Открытое пространство» проводится в группах от 5 до 1000 человек. Она может проводиться в виде семинара в течение одного дня, в виде конференций в течение 3 дней, или в виде еженедельных встреч членов организаций (учебных заведений, групп и пр.) [Open Space, 2011].

Технология Barcamp (Баркемп, англ. BarCamp) – международная сеть конференций, которая создаётся её участниками. Конференции открыты для всех, проходят в формате докладов, тренингов, презентаций, обсуждений. Весь материал предоставляется самими участниками [BarCamp, 2011]. Обычно для организации используется онлайн ресурс, на котором используется wiki или что-то наподобие социальной сети для обсуждения всех вопросов, от общих вопросов до программы мероприятия и т.п.

Метод проектов – это способ достижения учебной цели через детальную разработку некоторой заранее определенной проблемы. Разработка должна завершиться вполне реальным, осязаемым практическим результатом. Метод проектов – это комплексный метод обучения, дающий возможность учащемуся проявить самостоятельность в планировании, организации и контроле своей учебно-познавательной деятельности [Современные технологии обучения, 2011]. Результаты выполненных проектов должны быть "осязаемыми", т.е., если это теоретическая проблема, то конкретное ее решение, если практическая – конкретный результат, готовый к внедрению.

1.2. Интерактивность

Рассмотрим далее понятие «интерактивность». В [Интерактивность, 2011] даны следующие приведенные ниже определения.

Интерактивность – это принцип организации

системы, при котором цель достигается информационным обменом элементов этой системы. Степень интерактивности — это показатель, характеризующий, насколько быстро и удобно пользователь может добиться своей цели. Элементами интерактивности являются все элементы взаимодействующей системы, при помощи которых происходит взаимодействие с другой системой/человеком (пользователем).

Интерактивность – это способность информационно-коммуникационной системы активно и разнообразно реагировать на действия пользователя. Говорят, что система «умная», т.е. как бы обладает каким-то интеллектом.

Выше мы использовали определения, которые в большей степени отражают процесс взаимодействия пользователя с компьютерной системой. Сопоставив значение интерактивности в рамках соответствующих методов обучения и с использованием компьютерных систем, можно сделать вывод, что при создании электронных средств обучения интерактивность преобразуется в иную форму, т.к. один или несколько субъектов взаимодействия в учебном процессе, заменяются компьютерной системой или ее частью. Если компьютерная система заменяет преподавателя, то функции учителя должна выполнять сама эта система и обеспечивать необходимый уровень интерактивности, способствуя тем самым повышению уровня активности и самостоятельности учащегося, повышая его мотивацию и заинтересованность в обучении, помогая преодолевать возникающие проблемы. Если же компьютер используется для технической поддержки тех или иных интерактивных методов обучения (таких как, например, рассмотренные выше технологии «Открытое пространство» и Barcamp), то в рамках компьютерных систем должны быть обеспечены соответствующие функциональные возможности. Главное при этом – повышение эффективности взаимодействия субъектов учебного процесса при решении ими учебных задач.

2. Компьютерное моделирование и/или поддержка интерактивного обучения

Рассмотрим перечисленные выше виды учебной деятельности через призму интерактивных методов обучения, обозначив возможности компьютерного моделирования и/или поддержки соответствующих форм обучения. Отметим при этом, что многие из них реализуемы в виде занимательных игр, что является дополнительным стимулом к обучению как детей, так и взрослых людей. Некоторые программные решения дополняют рассмотренный выше арсенал интерактивных методов обучения.

2.1. Изучение новых слов, постоянное расширение словарного запаса

При изучении лексики языка очевидным

становится то, что учащемуся необходимо дать возможность самостоятельно исследовать словарный состав того или иного языка, анализировать близкие по смыслу слова и пр. Это должно способствовать лучшему их пониманию и запоминанию. В контексте интерактивности интересными на наш взгляд являются следующие решения.

Программа для изучения лексики английского языка LearnWords, разработчик LearnWords Software [LearnWords, 2011]. Здесь используются различные упражнения: Карточка (рис.1), Мозаика (рис.2), Выбор перевода, Угадать перевод, Выбор слова, Написание (рис.3).

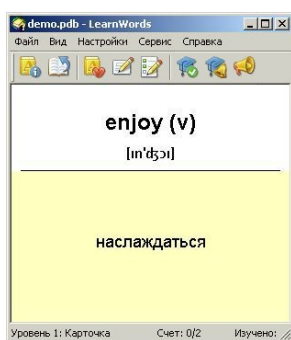


Рисунок 1 – Карточка слова



Рисунок 2 – Мозаика слов

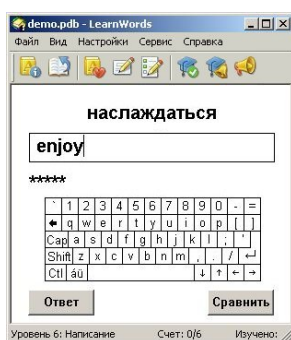


Рисунок 3 – Написание слов

Все они основаны на особенностях работы ассоциативной памяти человеческого мозга и способствуют более быстрому и качественному запоминанию слов.

Еще одна программа для заучивания слов VX

Language acquisition [BX Language acquisition, 2011] основана на накоплении статистики ответов на специально подготовленные задания. За каждое правильно выполненное задание учащийся получает определенное количество баллов, а за каждую ошибку баллы снимаются. В каждом последующем упражнении даются задания с наименьшим баллом, некоторое количество новых заданий, а также задания на повторение выученного. При условии получения определенного количества баллов задание считается условно-выученным. Условно-выученные слова предлагаются для повторения по методике Эббингауза. Данная программа позволяет заучивать написание и произношение иностранных слов как в режиме вопрос-ответ, так и в режиме диктанта. Здесь имеется также возможность создавать видеословари на базе субтитров к фильмам и видеофайлов формата AVI.

В рамках программы BX Language acquisition реализовано три активных метода заучивания иностранных слов:

- *Вариант* – учащемуся предлагается слово на русском языке и несколько вариантов его перевода на иностранный язык, из которых он должен выбрать правильный. При этом при выборе неправильного ответа рядом с выбранным иностранным словом выводится его перевод.

- *Мозаика* – здесь учащийся должен собрать «мозаику» из 8-ми пар слов, подбирая переводы, перетаскивая слова мышью.

- *Написание* – при выполнении данного задания учащиеся вводят с клавиатуры слова, являющиеся переводом предложенного слова, заучивая тем самым правильное написание.

В BX Language acquisition интересным образом реализован один пассивный метод заучивания *Просмотр карточек слов*. Просматривать карточки слов можно в пассивном режиме, деля при этом любую другую работу на компьютере. В углу экрана попеременно отображаются заучиваемые слова из словаря. При наличии свободного времени взгляд учащегося привлекает задание в карточке, а задержка показа ответа даёт возможность вспомнить или предположить ответ и сразу же свериться с показанным ответом карточки. Если для обучения используется видеословарь на базе субтитров к фильму, то вместе с карточкой слова отображается и соответствующая видеокартинка.

Способствуют заучиванию слов также и многие электронные словари, как, например, многоязычный словарь TranslateIt! Teacher!, TranslateIt Team, [TranslateIt!, 2011] и многие другие.

В качестве других примеров использования интерактивного изучения лексики языка приведем следующие:

- Интерактивные иллюстрации, щелкая мышью по которым учащийся может получать возможность прочитать (услышать) соответствующее слово. Например, при изучении таких групп понятий, как части/органы

человеческого лица/тела и т.п., предметы мебели в помещении, обстановка в том или ином заведении (магазин, рынок, аптека, университет, музей, театр, Встреча, Прощание, Знакомство, Разговор по телефону, Ресторан, Покупки, Транспорт, Отдых, Семья, и т.д.) и т.п.

- Исследование синонимических рядов и групп антонимов по интерактивному тексту. Например, учащемуся для изучения дается тот или иной текст, в котором словам поставлены в соответствие синонимы (антонимы). Пользователь выделяет то или иное слово и получает список его синонимов (антонимов). Кроме того, дается возможность увидеть (прочитать) и услышать новые предложения с употреблением синонимов (антонимов), которые сформированы на основе исходного текста.

- Словарные упражнения по методу карточек: иностранные слова и их переводы на родной для учащегося язык разбросаны в виде карточек; требуется с помощью мыши перетащить иностранные слова к их переводам. Аналогично – разбросаны изображения некоторых предметов и слова-карточки. Требуется перетащить изображения к соответствующим словам (или наоборот).

- Игра «словарная мозаика», грамматическая мозаика, дано игровое поле, на котором перевернуты карточки со словами и соответствующими им изображениями. Необходимо за ограниченное время перевернуть все пары карточек.

- Кроссворд.

- Побуквенное собирание слов: дана либо аудиозапись слова, либо изображение, либо перевод на родной язык и изображения букв, из которых состоит это слово. Учащемуся требуется собрать это слово из предложенных букв. Более сложное упражнение, когда даны в том числе буквы, которых в искомом слове нет.

- Игра в словообразование. Даны корни, приставки, суффиксы, окончания, из которых учащийся должен собрать как можно больше слов.

2.2. Усвоение правил словообразования, принятого в том или ином языке

Как правило, это образование множественного числа, падежных форм, временных форм глаголов и др. В компьютерных программах изучение грамматических правил представлено в виде уроков, каждый из которых посвящен изучению определенной грамматической темы.

Например, в программе Bridge to English, "Интенс" [Bridge to English, 2011] грамматический материал объясняется на примерах современного английского разговорного языка. Грамматические задания представлены в виде упражнений различной степени сложности – от более простых к сложным.

При выполнении упражнений на начальной стадии изучения курса грамматики обучаемый должен выбрать один из двух вариантов ответа, а на

заключительном, более сложном этапе, наряду со знанием языкового материала он должен показать широту кругозора, отвечая на специальные вопросы.

Теоретический и практический материал позволяет решить следующие методические задачи (например, при закреплении форм глаголов): сформировать навык формулирования ответа на вопрос в той же видовойременной форме, в которой поставлен вопрос; закрепить навык узнавания формы глагола изучаемого времени; сформировать навык внимательного чтения вопроса на основе установки, заданной компьютером; сформировать моторный навык написания изучаемого грамматического явления на примерах упражнений, в которых обучаемый должен сам набрать правильный ответ; закрепить зрительный образ отрабатываемого грамматического явления.

2.3. Усвоение грамматических правил построения словосочетаний и предложений

Для отработки и усвоения грамматических правил в интерактивных упражнениях учащимся, как правило, предлагается исправить текст. Для этого достаточно щелкнуть мышью по слову с ошибкой и внести изменения с помощью клавиатуры. В программе Sentence Exerciser [Sentence Exerciser, 2011] предлагается набор тестов по грамматике английского языка. В основном, в упражнении необходимо выбрать правильный ответ из нескольких вариантов, но не просто кликнуть мышкой, а написать (ввести с клавиатуры) ответ, а при написании слова запоминаются намного лучше.

Разработчики программы Reward (Young Digital Planet S.A., Macmillan Publishers Limited) [Reward, 2011] использовали следующее решение на усвоения формирования грамматических правил построения словосочетаний и предложений: здесь применяется техника "перетащить-и-сбросить" для исправления порядка слов в предложении.

2.4. Формирование базовых речевых навыков – произношение отдельных звуков, слов, фраз

В современных программах уделяется большое внимание формированию речевых навыков, ведь без них невозможно говорить о полноценном владении языком. Одним из известных примеров компьютерных средств обучения языку, в котором большое внимание уделяется приобретению речевых навыков, является программа «Профессор Хиггинс, Английский без акцента» [IstraSoft, 2011]. В данной программе каждый урок содержит несколько упражнений с поддержкой распознавания речи. Учащийся может заполнять окошки и столбцы таблиц в упражнениях, используя свой собственный голос. Чтобы заполнить окошко, необходимо поместить в него курсор и произнести соответствующее слово или выражение. В каждом окне словаря с новыми словами урока содержится тест на произношение. Чтобы пройти тест,

необходимо правильно произнести каждое слово из словарного окна. Процесс обучения говорению визуализируется с помощью осциллограммы записи эталонного и произнесенного учащимся слова или фразы. Благодаря этому, обучаемые могут увидеть, в каком месте они допустили ошибку, и исправить ее, заново пробуя произносить новые слова, стремясь к тому, чтобы осциллограммы стали максимально похожи друг на друга.

2.5. Формирование навыков ведения беседы, рассказа, монолога и пр.

Как было указано выше, при рассмотрении интерактивных методов обучения, формирование коммуникативных навыков возможно только в условиях общения учащихся друг с другом либо с учителем. Такого рода упражнения могут выполняться посредством форума или чата в сети Интернет. Обучающиеся встречаются в виртуальной гостиной и обсуждают предложенные темы в текстовом или голосовом режимах. Можно разместить свое мнение на специальном дискуссионном форуме, аналогичном NewsGroup в Интернете. Эффективным также по-прежнему остается заучивание заранее подготовленных аудиозаписей, созданных с помощью носителей языка. Большую популярность приобрел также Skype и др. подобные средства связи, через которые предлагаются услуги по обучению речевой практике.

Для повышения мотивации и интереса к обучению активно используется видео: Intellect - Видеокурс "Intellect" (25-й кадр), Центр Учебных Программ "Интеллект", <http://www.cd-intellect.ru> [Видеокурсы, 2011].

2.6. Формирование навыков грамотного письма

Простейшие интерактивные компьютерные программы, способствующие формированию навыков грамотного письма, рассмотрены выше. Это задания типа «Написание», в которых учащемуся предлагается то или иное слово, перевод которого он должен самостоятельно ввести с клавиатуры. Развитием таких заданий является предъявление в качестве исходных данных не отдельных слов, а словосочетаний или предложений. К сожалению, в настоящее время не существует полноценных программных решений, осуществляющих автоматический анализ произвольных текстов. Тем не менее, некоторые такие возможности уже существуют при условии использования так называемого ограниченного естественного языка с заранее заданным множеством слов (словоформ) и ограничениях на грамматическую структуру предложений (в частности, на порядок слов, что возможно далеко не во всех языках).

В большинстве случаев учащимся предлагается посылать письменные упражнения выбранному им учителю по e-mail. В сети Интернет в настоящее

время реализуются также проекты в стиле социальных сетей, где упражнения по написанию различных текстов могут быть проверены кем угодно из сетевого сообщества учащихся. Примером такого ресурса является сообщество по изучению иностранного языка <http://www.busuu.com>.

Интересным и перспективным, на наш взгляд, является применение средств когнитивной графики и разработка на их основе программ, которые могли бы формировать различные изображения (рисунки) по тем текстам, которые вводит с клавиатуры учащийся. В процессе визуализации описанных сцен в случае ошибок учащийся по изображению может сразу увидеть ошибку и понять, как ее исправить. Очевидно, что для разработки такой программы требуется использование методов автоматического анализа текстов, а также семантическое описание языка либо его части.

3. Интеллектуализация интерактивных компьютерных средств обучения языку

Рассмотрим перспективы развития интерактивных компьютерных средств обучения языку. Очевидно, что традиционные алгоритмы интерактивного взаимодействия с пользователем практически исчерпали себя. В особенности это заметно в случае обучения языкам, т.к. естественный язык является примером предметной области, которая заслуженно является одной из самых трудно формализуемых. Кроме того, проверка ошибок учащихся, изучающих язык, зачастую является достаточно сложной интеллектуальной задачей даже для преподавателя. Тем не менее, на наш взгляд, уже сегодня некоторые учебные задачи могут быть представлены в виде адекватных компьютерных интеллектуальных моделей. Рассмотрим здесь словесное описание некоторых из них, которые могут послужить постановкой задачи для соответствующих программных реализаций.

Для приведенного в разделе 2.1 примера игры в словообразование на основе наличия заранее заготовленных наборов приставок, суффиксов и основ можно разработать систему, которая могла бы автоматически осуществлять разбор по составу любого введенного пользователем слова. В развитие этого, для усвоения процессов словообразования (см. раздел 2.2) целесообразно разработать алгоритмы, которые бы выдавали всю морфологическую информацию о введенной пользователем словоформе. Аналогично – для каждой словоформы в составе введенного пользователем словосочетания, предложения или текста. Если быть более точным, то требуется адаптировать существующие алгоритмы автоматического анализа текстов под учебные цели и создать на их основе обучающие ресурсы. Не менее интересным было бы наличие программы, которая «умела» бы склонять (ставить в нужную форму) любые введенные пользователем слова в соответствии с описанными им характеристиками.

Например, таким образом можно было бы осуществлять обучение на примерах образования падежных форм имен существительных, прилагательных, числительных и пр. или форм глаголов. Интересны также такие задачи, как образование отглагольных существительных, прилагательных и причастий на основе введенных пользователем глаголов, образование сравнительных степеней прилагательных, кратких прилагательных на основе полных и наоборот. Аналогично можно привести еще немало примеров конкретных учебных задач, которые в процессе освоения конкретного языка вызывают у учащихся большое количество затруднений, тогда как на занятиях преподаватель по естественным причинам неограниченности естественного языка не успевает рассмотреть все особенности того или иного языкового явления.

В некоторых рассмотренных в разделе 2 примерах интерактивных программ указывалось, что в процессе работы сохраняются действия учащегося, подсчитываются баллы. Эта возможность может быть расширена путём более тщательного сбора и обработки информации об успехах и ошибках учащегося. Так, например, в случае увеличения количества ошибок в тех случаях, где раньше учащийся не ошибался, компьютерная система могла бы подсказать ему сделать перерыв, предположив, что он устал. В зависимости от индивидуальных особенностей система могла бы подбирать наиболее оптимальный для тех или иных учащихся сложность, объём и темп работы.

Для задач усвоения правил словообразования, а также грамматических правил построения словосочетаний и предложений «умение» системы автоматически анализировать словоформы и выделять, например, окончания, может помочь в автоматическом формировании упражнений на усвоение образования падежных форм и т.п.

Как указывалось выше, использование алгоритмов автоматической обработки текстов может позволить создавать упражнения открытого типа, при выполнении которых учащимся предлагается ввести с клавиатуры произвольных текст на заданную тему. Проверка такого текста в этом случае будет осуществляться системой обучения автоматически. При этом при наличии в системе достаточно развитого механизма объяснения учащемуся должны выдаваться объяснения относительно допущенных ошибок с выводом соответствующих правил и предложением выполнить упражнения на их закрепление.

Выше уже упоминалось также актуальное направление автоматического синтеза и распознавания речи. Так, уже сегодня при изучении иностранного языка можно использовать синтезаторы речи. Это, кстати, уже используется в ряде электронных словарей и переводчиков в сети Интернет (например, в Google translate и др.), но данная возможность доступна лишь для отдельных

слов. Чтение же полноценных текстов предполагает также и моделирование ритмики и интонации, присущей тому или иному языку.

Как указывалось выше, большие перспективы имеет применение средств когнитивной графики и разработка на их основе программ, которые могли бы формировать различные изображения (рисунки) по тем текстам, которые вводит с клавиатуры учащийся. Не менее интересна на наш взгляд и обратная задача – вывод произвольного изображения (в идеале – автоматически сгенерированного системой) с задачей для учащегося описать его с помощью нескольких предложений. Очевидно, что разные учащиеся будут описывать одно и то же изображение с использованием разных слов и грамматических конструкций, которые система должна «уметь» анализировать автоматически. В качестве примера конкретного упражнения с использованием когнитивной графики можно привести задачу на усвоение пространственных предлогов. Для моделирования этой задачи в качестве основы могут быть взяты изображения предметов реальной окружающей действительности и все предлоги. Учащемуся необходимо будет писать достаточно простые фразы типа: «Книга лежит на столе», которые, очевидно, достаточно несложно проанализировать. Однако, в случае наличия в системе достаточно большой исходной базы слов и изображений можно получить достаточно гибкую и универсальную систему, обучающую всем нюансам использования предлогов указанной группы.

Естественно, что практически каждая из рассмотренных выше учебных задач при ее моделировании в виде соответствующей интеллектуальной компьютерной программы приобретёт ещё большую гибкость и универсальность при условии подключения к ней модуля приобретения знаний или самообучения. Так, например, множества слов, предложений и текстов для создания тех или иных упражнений могут извлекаться из любых доступных системе электронных текстов, в том числе, размещенных в сети Интернет.

ЗАКЛЮЧЕНИЕ

В данном сообщении рассмотрены далеко не все возможные интерактивные средства обучения языку. Здесь сделана лишь первая попытка разработки методологии систематизации указанных средств и поиска новых эффективных решений.

Так как при изучении языка учащимся необходимо дать максимум возможностей оперирования объектами изучаемого языка (словами, их частями, словосочетаниями, предложения и др. характеристиками), то интерактивные методы обучения являются одним из самых адекватных и перспективных средств вовлечения учащихся в активную познавательную деятельность.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Видеокурсы, 2011] Видеокурсы ускоренного изучения иностранных языков "Intellect" [Электронный ресурс]. – 2011. Режим доступа: <http://www.cd-intellect.ru> – Дата доступа: 10.12.2011.

[Интерактивность, 2011] Интерактивность // Материал из Википедии - свободной энциклопедии [Электронный ресурс]. – 2011. Режим доступа: <http://ru.wikipedia.org/wiki/Интерактивность> – Дата доступа: 10.12.2011.

[Интерактивные методы обучения, 2011] Интерактивные методы обучения // Материал из Letopisi.Ru. «Время вернуться домой» [Электронный ресурс]. – 2011. Режим доступа: http://letopisi.ru/index.php/Интерактивные_методы_обучения – Дата доступа: 17.12.2011.

[Кашлев, 2005] Кашлев С. С. Технология интерактивного обучения. – Мн.: Белорусский верасень, 2005.

[Оганесян, 2002] Оганесян Н. Т. Методы активного социально-психологического обучения: тренинги, дискуссии, игры. – М.: Ось-89, 2002. – 176 с.

[Современные технологии обучения, 2011] Современные технологии обучения школьников иностранному языку // Foreign Language Teaching. Сайт ассоциации учителей-практиков Московской области [Электронный ресурс]. – 2011. Режим доступа: <http://fl-teaching.ru/forum/55-413-1> – Дата доступа: 18.12.2011.

[Фопель, 1998] Клаус Фопель. Копилка игр от "классика": Как научить детей сотрудничать? Психологические игры и упражнения - практическое пособие для педагогов и школьных психологов. // Перевод с немецкого, — Москва: Генезис, 1998.

[Якубовская, 2006] Якубовская Л.П. Методика преподавания психологии: учебное пособие/ К. П. Якубовская. – М.: «Просвещение», 2006. – 270 с.

[BarCamp, 2011] BarCamp. Материал из Википедии – свободной энциклопедии [Электронный ресурс]. – 2011. Режим доступа: <http://ru.wikipedia.org/wiki/BarCamp> – Дата доступа: 20.12.2011.

[Bridge to English, 2011] Bridge to English - английский язык [Электронный ресурс]. – 2011. Режим доступа: <http://www.intense.ru/> – Дата доступа: 22.12.2011.

[BX Language acquisition, 2011] BX Language acquisition [Электронный ресурс]. – 2011. Режим доступа: <http://bxmemo.narod.ru/> – Дата доступа: 12.12.2011.

[IstraSoft, 2011] IstraSoft [Электронный ресурс]. – 2011. – Режим доступа: <http://istrasoft.ru/main/Продукты/ПрофессорХиггинсАнглийскийбезакцентаПодробнееопрограмме/tabid/67/language/ru-RU/Default.aspx> – Дата доступа: 15.12.2011

[LearnWords, 2011] LearnWords [Электронный ресурс]. – 2011. Режим доступа: <http://www.learnwords.ru/index.html> – Дата доступа: 3.12.2011.

[Open Space, 2011] Технология Open Space [Электронный ресурс]. – 2011. Режим доступа: http://hrvesna.com/publ/obuchenie_i_razvitie/tehnologija_open_space/7-1-0-96 – Дата доступа: 20.12.2011.

[Reward, 2011] Reward, Young Digital Planet S.A., Macmillan Publishers Limited [Электронный ресурс]. – 2011. Режим доступа: www.reward.ru – Дата доступа: 15.12.2011.

[Sentence Exerciser, 2011] Sentence Exerciser – набор тестов по грамматике английского языка [Электронный ресурс]. – 2011. Режим доступа: <http://soft.mydiv.net> – Дата доступа: 23.12.2011

[TranslateIt!, 2011] TranslateIt! - многоязычный словарь [Электронный ресурс]. – 2011. Режим доступа: <http://mac.translateit.ru/ru/index.shtml> – Дата доступа: 6.12.2011.

COMPUTER MODELLING OF INTERACTIVE METHODS OF THE FOREIGN LANGUAGE STUDYING: PROBLEM STATEMENT

Eliseeva O.E., Erchenko E. A.

*Belarusian State University,
Minsk, Belarus*

olac@open.by

zheton_88@mail.ru

Work is devoted to the consideration of interactive methods of language studying with the usage of computer programs. Now it is one of the most effective ways to increase the interest and motivation to independent informative activity. This direction is interesting as well, that here it is necessary to find new methods and means for realization of intellectual systems of studying where the system acts as the high-grade interlocutor and the teacher

It is known, that most effectively process of mastering of a foreign language occurs in the course of dialogue. Ideally, this is an intercourse of a studying language person with the native speaker. However, such possibility doesn't always exist. Therefore by present time it is developed a set of various foreign language studying techniques. The majority of effective techniques mean interaction (dialogue) of the pupil with the teacher or pupils with each other. Unfortunately, the teacher frequently should become a certain similarity of the passive participant of dialogue. In such situation it is clear there is an expediency of the usage of computers with multimedia.

In work are considered kinds of educational activity at language studying through a prism of interactive methods of studying, having meant possibilities of computer modeling and-or support of corresponding forms of studying.

There are shown prospects of development of interactive computer tutorials. It is obvious, that traditional algorithms of interactive interaction with the user have practically settled themselves. It is seen in case language studying.

As at the language studying by the pupil it is necessary to give a maximum of possibilities of operating objects of studied language (words, their parts, word-combinations, sentences, etc.) interactive methods are one of the most adequate and perspective techniques of involving of pupils in active informative activity.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8:528.9

ИНТЕЛЛЕКТУАЛЬНЫЕ ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

Самодумкин С.А.

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

Samodumkin@bsuir.by

Рассмотрены принципы и подходы к проектированию интеллектуальных геоинформационных систем. В основу предлагаемого подхода положено понятие семантической модели интеллектуальной геоинформационной системы и кодирование информации с использованием семантического SC-кода. Предложенные в работе принципы иллюстрируются фрагментами интеллектуальной справочной системы по Республике Беларусь.

Ключевые слова: семантическая сеть, семантическая модель интеллектуальной геоинформационной системы, SC-код.

ВВЕДЕНИЕ

В настоящее время геоинформационные системы являются инструментом обработки пространственной информации [Абламейко, 2000], [Крючков, 2006]. Общей характерной чертой для такого класса систем является решение информационно-поисковых задач с привязкой объектов на некоторую территорию местности. Однако многообразие типов объектов местности, большое количество различных топологических отношений и множество встречаемых на практике задач, для которых не всегда существуют алгоритмические решения, требуют обратиться к вопросу интеллектуализации геоинформационных систем.

Классическая архитектура геоинформационной системы в качестве информационных компонентов включает пространственную и атрибутивную базы данных. Таким образом, все объекты местности имеют пространственную привязку к территории местности, а также задаются характеристики объектов. На практике же для решения прикладных задач требуется установление отношений между объектами местности, и в данном случае, используя инструментальные геоинформационные системы, возможно только установление топологических отношений. Установление других типов отношений, в том числе предметных отношений, весьма затруднительно и может быть решено в частном виде путем создания программ на встраиваемых в инструментальные геоинформационные системы языках программирования. Причем для установления какого-либо вида семантической связи требуется разработка алгоритма и его программирование.

В основе предложенного в работе подхода лежит создание семантической модели (sc-модели) [Голенков, 2001] геоинформационной системы, которая включает в себя базу знаний, машину обработки знаний и интеллектуальный пользовательский интерфейс со средствами визуального взаимодействия с объектами карты. Особенностью такой модели является представление знаний предметной области в виде семантической сети, а проектирование прикладных интеллектуальных систем осуществляется по технологии ОСТИС [Голенков, 2011].

1. Способ описания различных классов объектов местности

Для каждого объекта местности выделены основные, присущие только ему, семантические характеристики. Особо отметим, что метрические характеристики таким свойством не обладают. Для указания семантических свойств классам объектов местности используется разработанный и действующий классификатор топографической информации, отображаемой на топографических картах и планах городов ОКРБ 012-2007 [ОКРБ, 2007].

Согласно данному классификатору каждый класс объектов местности имеет уникальное однозначное обозначение. Иерархия классификатора имеет восемь ступеней классификации и состоит из кода класса, кода подкласса, кода группы, кода подгруппы, кода отряда, кода подотряда, кода вида, кода подвида. Таким образом, благодаря способу кодирования уже заданы родовидовые связи, отражающие соотношения различных классов объектов местности, а также установлены характеристики конкретного класса объектов местности. В связи с

тем, что задаются основные свойства и отношения не конкретных физических объектов, а их классов, то такая информация является по отношению к конкретным объектам местности метаинформацией, а совокупность данной метаинформации представляет собой онтологию объектов местности, которая в свою очередь является частью базы знаний интеллектуальной геоинформационной системы.

Онтология объектов местности включает описание следующих классов объектов местности:

- водные объекты и гидротехнические сооружения;
- населённые пункты;
- промышленные, сельскохозяйственные и социально-культурные объекты;
- дорожная сеть и дорожные сооружения;
- растительный покров и грунты.

Онтология объектов местности представляет собой дерево классификации в соответствии с иерархией, приведенной на рисунке 1. Для каждого класса объектов местности установлены

Далее рассмотрим получение онтологии объектов местности и ее представление в sc-модели на примере кодирования одного из классов объектов местности «реки».

На первом этапе создается статья, кодированная с использованием SCn-кода (псевдоестественного языка кодирования семантических сетей) [Ивашенко, 2012]:

```

реки
  ≙ [ "СК_31410000" ]
  ∈ реальный объект
  ∈ подгруппа объектов местности
  ⊂ водотоки
  - Отношения, заданные на понятии:
    • собственное название*
    • ширина по шкале*
    • признак судоходства*
    • качественные особенности воды*
  
```

На втором этапе sc.n-статья транслируется во внутреннее представление, т.е. в семантическую

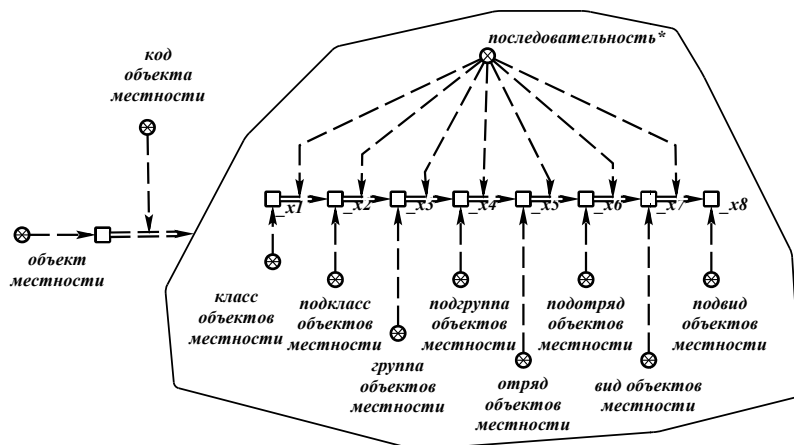


Рисунок 1 - Уровни иерархии классов объектов местности

родовидовые связи. В качестве примера на рисунке 2 приведена иерархия водных объектов.

сеть специального вида. Эквивалентное кодирование в виде графического представления sc.g-текста показано на рисунке 3.

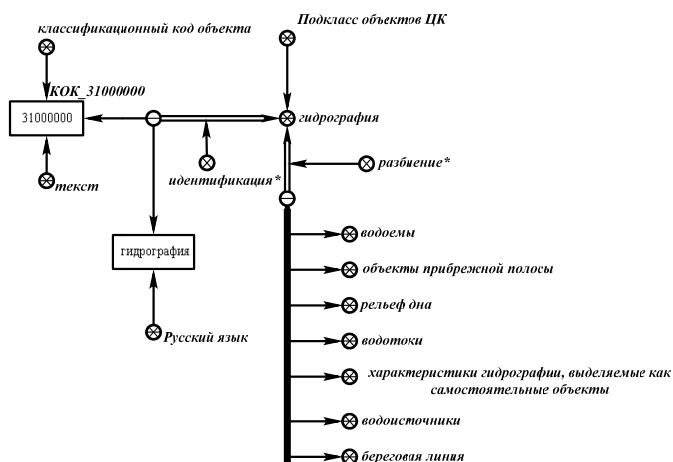


Рисунок 2 – Иерархия водных объектов местности

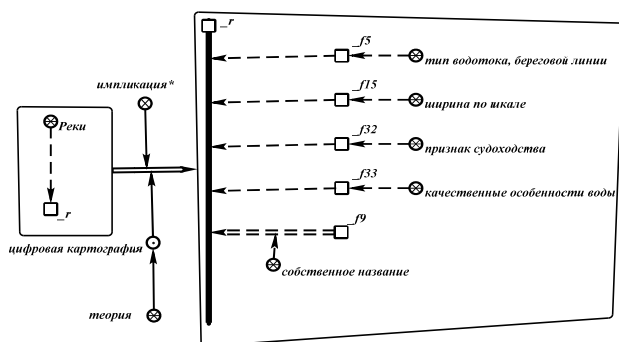


Рисунок 3 - Описание класса объектов местности «реки» в виде семантической сети и его представление в виде sc.g-текста

Кроме статей, описывающих классы объектов местности, в онтологию объектов местности входят статьи с описанием признаков, характеризующих

объекты местности. Отметим, что для каждого класса объектов местности выделен свой, характерный только ему, набор признаков (например, на рисунке 3 для всех объектов местности типа «реки» могут быть заданы отношения «собственное значение*», «ширина по шкале*», «признак судоходства*», «качественные особенности воды*»). В качестве примера приведем пример sc.n-статьи, описывающей признак «количество жителей*».

количество жителей*

= быть количеством жителей*

∈ бинарное отношение

– Область определения:

- города
- поселки городского типа (ПГТ)
- поселки сельского типа
- поселки дачного типа
- поселки, не отнесенные к категории ПГТ, и коттеджного типа

– Домен:

- объект местности

/* домен по первому атрибуту */

- величина

– Значение:

- человек_ : число

/* домен по второму атрибуту */

– Схема отношения:

- объект местности_
- количество жителей_

– Определение:

- Поясн.(количество жителей*)

▲ [количество жителей - численное количество жителей]

Таким образом, рассмотренная онтология объектов местности и способ ее формального задания позволяют описать все основные классы объектов местности и установить для этих классов набор признаков, характерных для рассматриваемого класса объектов местности, что в

свою очередь позволяет в дальнейшем создать базу знаний объектов местности с уже установленными родовидовыми отношениями, а также семантическими атрибутами [Самодумкин, 2011].

2. База знаний интеллектуальных геоинформационных систем

Описанная в предыдущем разделе онтология объектов местности представляет собой фрагмент базы знаний для рассматриваемого класса систем, который оформлен в виде повторно используемого компонента и используется при разработке прикладных геоинформационных систем.

Следующая часть базы знаний – это представление предметных знаний в рамках проектируемой прикладной системы, что представляет собой набор статей, описывающих конкретные объекты местности. Такой раздел базы знаний будем называть «предметными» знаниями, т.е. раздел, где представляются знания предметной области.

В основу формирования «предметной» базы знаний положен принцип эволюционного проектирования. Это означает, что данный раздел формируется поэтапно, как показано на рисунке 4.

Первый этап формирования «предметной» базы знаний – это анализ электронной карты и трансляция в базу знаний объектов местности на заданную территорию. На этом этапе определяется, во-первых, к какому классу принадлежит исследуемый объект местности и, далее в зависимости от типа объекта, формируется статья, соответствующая конкретному физическому объекту местности.



Рисунок 4 - Компоненты баз знаний и этапы их получения

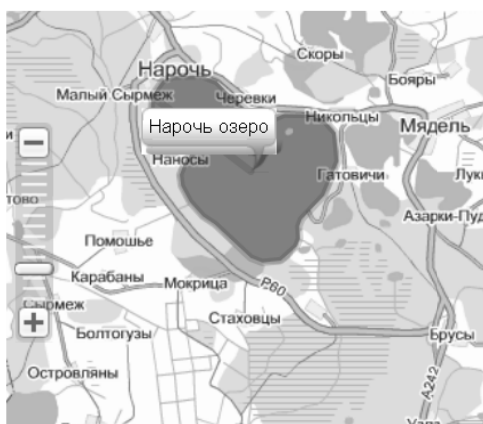
Таким образом, создается множество статей, описывающих конкретные объекты местности для каждого класса объектов местности.

В качестве примера приведем sc.n-статью, описывающую озеро Нарочь

Нарочь озеро

∈ озёра

- Абсолютная высота:
 - метр_165
- Тип водотока, береговой линии:
 - постоянный
- Глубина:
 - метр_24,08
- Собственное название:
 - Нарочь
- Качественные особенности воды:
 - пресная
- Площадь:
 - километр квадратный_79,6



Следует отметить, что на данном этапе формирования базы знаний могут быть установлены дополнительные отношения, в частности, отношения принадлежности и топологические отношения.

Второй этап формирования «предметной» базы знаний – это интеграция с внешними базами знаний. На этом этапе, помимо географических знаний, могут добавляться знания смежных предметных областей, тем самым становится возможным отражение межпредметных связей. Наглядным примером служит интеграция с биологическими классификаторами, которые в реализации представляют собой онтологию объектов флоры и фауны. Такая интеграция расширяет функциональные и интеллектуальные возможности прикладной геоинформационной системы. Отметим, что на данном этапе снимается омонимия в названиях географических объектов, принадлежащих классам населенных пунктов за счет использования классификатора СОАТО (система обозначений объектов административно-территориального деления и населенных пунктов).

В качестве примера приведем статью базы знаний «Березинский заповедник», полученную в результате интеграции внешних баз знаний.

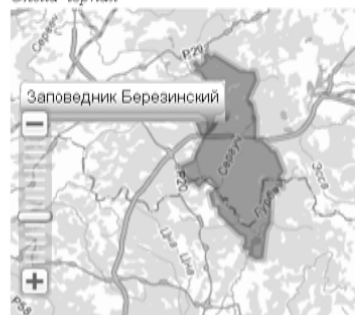
заповедник Березинский

= Березинский заповедник

∈ природоохранные территории

∈ заповедники

- Собственное название:
 - Березинский
- Состояние:
 - действующий
- Площадь:
 - гектар_80929
- Местонахождение:
 - область_Витебская область
 - область_Минская область
 - район_Лепельский район
 - район_Дошицкий район
 - район_Борисовский район
- Характер грунта:
 - дерново-подзолистая почва
 - дерново-подзолистая заболоченная почва
 - дерновая заболоченная почва
 - торфяно-болотная почва
- Рельеф:
 - моренная возвышенность
 - озерно-ледниковая равнина
- Реки:
 - Березина
 - Эсса
- Озера:
 - Спыцца
 - Плавно
 - Манец
 - Домжерцкое
 - Московца
 - Пострежское
 - Палих
- Редкие растения:
 - Венерин башмачок настоящий
 - Бровник одноклубневый
 - Камнеломка болотная
 - Пыльцеголовник красный
 - Береза карликовая
 - Лишняя северная
 - Касатик сибирский
 - Пололепестник зеленый
- Редкие животные:
 - Зубр
 - Бурый медведь
- Лесообразующие породы:
 - Сосна
 - Ель
 - Береза
 - Спыха черная



3. Машина обработки знаний интеллектуальных геоинформационных систем

Одним из достоинств интеллектуальных систем, разработанных по технологии ОСТИС, является решение предметных задач, когда нет четкой спецификации и алгоритма ее решения. Это достигается с помощью формирования продукций, которые записываются и хранятся так же в базе знаний. Технология проектирования машин обработки знаний и модели решения задач в соответствии с указанной технологией рассмотрена в работе [Заливако и др., 2012].

В качестве примера рассмотрим решение следующей задачи: «Определить, существует ли водный путь между городами Минск и Речица. При наличии такого водного пути результат отобразить на карте».

В качестве исходных данных являются следующие утверждения базы знаний:

1. Через город Минск протекает река Свислочь.
2. Через город Речица протекает река Днепр.
3. Река Свислочь является притоком реки Березина.
4. Река Березина является притоком реки Днепр.

В процессе решения на первой итерации на основе анализа топологических отношений «примыкание» будет установлено:

1. Река Свислочь впадает в реку Березина (т.е. река Свислочь примыкает к реке Березина)
2. Река Березина впадает в реку Днепр (т.е. река Березина примыкает к реке Днепр)

и будет сделано заключение о том, что по реке Свислочь можно попасть в реку Днепр.

На второй итерации сопоставляется топологическая принадлежность города Минска и реки Свислочь, а также города Речица и реки Днепр. В итоге будет найден путь, который отображается на карте местности.

Необходимо отметить, что в процессе решения задач генерируются дополнительные знания, необходимые в процессе вывода, которые могут быть сохранены и в дальнейшем использоваться при решении других задач или участвовать в выводе при ответе на поисковые запросы.

Интерфейс пользователя поддержан картографическим редактором, совместимым с системой координат исходного набора картографической информации. Для визуализации объектов местности используется картографический сервис <http://maps.yandex.ru>. С целью интеграции с последним для каждого объекта местности осуществляется геокодирование физических объектов местности.

ЗАКЛЮЧЕНИЕ

Приведенный в работе подход к проектированию интеллектуальных геоинформационных систем в соответствии с технологией OSTIS позволяет создавать прикладные интеллектуальные системы указанного класса. Причем сроки проектирования сокращаются за счет повторного использования следующих компонентов: онтологии объектов местности, базы знаний объектов местности, машины обработки знаний и картографического интерфейса. В общем случае проектировщику прикладной геоинформационной системы необходимо в соответствии с онтологией объектов местности сформировать базу объектов местности на заданную территорию и записать утверждения предметной области в виде продукций.

Работа поддержана грантом БРФФИ-РРФФИ Ф10Р-148.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Абламейко, 2000] Абламейко, С.В. Географические информационные системы. Создание цифровых карт / С.В. Абламейко, Г.П. Апарин, А.Н. Крючков. – Минск : Ин-т техн. кибернетики НАН Беларуси, 2000.

[Голенков и др, 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков [и др.] – Мн. : БГУИР, 2001.

[Голенков и др, 2011] Голенков, В.В. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем / В. В. Голенков, Н.А. Гулякина // Материалы междунар. Научно-технической конференции Открытые семантические технологии проектирования интеллектуальных систем – Мн. , 2011.

[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, 2012. –С. 297-314.

[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

[Крючков, 2006] Крючков, А.Н. Интеллектуальные технологии в геоинформационных системах / А.Н. Крючков, С.А. Самодумкин, М.Д. Степанова, Н.А. Гулякина; Под науч. ред. В.В.Голенкова. – Мн.: БГУИР, 2006.

[ОКРБ, 2007] Цифровые карты местности. Топографическая информация, отображаемая на топографических картах и планах городов / ОКРБ 012-2007.

[Самодумкин, 2011] Проектирование интеллектуальных геоинформационных систем справочного назначения / С.А. Самодумкин // Информационные технологии и системы 2011 (ИТС 2011) : материалы международной научной конф. (Минск, 26 октября 2011г.) – Минск : БГУИР, 2011.

INTELLECTUAL GEOINFORMATION SYSTEMS

Samodumkin S.

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

Samodumkin@bsuir.by

Principles and approaches to designing of intellectual geoinformation systems are considered. In a basis of the offered approach the concept of semantic model of intellectual geoinformation system and coding of the information with use of a semantic SC-code is necessary. The principles offered in work are illustrated by fragments intellectual system across Byelorussia

INTRODUCTION

Nowadays geographic information systems can be called an instrument for spatial information processing. Common feature for such system class is solving information retrieval problems with objects lock-on to some area territory. But variety of area object types, large quantity of different topological relations and great number of practical problems, which sometimes don't have algorithmic solution, make it necessary to turn to the question of geographic informational systems intellectualization.

Point of view, suggested in this work, is based on creation of geographic information systems semantic model (sc-model), which consists from knowledge domain, knowledge processing engine and user interface, which is able to visually interact with map objects. Particular feature of such model is representation of problem domain in the form of semantic web. Applied intelligent system design is making with the help of OSTIS technology.

MAIN PART

For each area object main, individual characteristics are named. Notice that metric characterizations don't possess such properties. To indicate semantic qualities to area objects classes it is used designed and active classifier of topological information, which is represented on topological and city maps ОКРБ 012-2007 [ОКРБ, 2007].

According to this classifier, each area objects class has its own unique one-digit indication. Classifiers hierarchy has eight classification stages and consists from class code, subclass code, group code, subgroup code, order code, suborder code, species code, subspecies code. So, thanks to coding method, relations are already named, which represent ratio of different area objects classes, and also characteristics of certain

area object class are determined. As a result of the fact, that main qualities of not certain physical objects, but their classes are indicated, such information can be called metainformation in respect of certain area objects, and totality of this metainformation represent area objects ontology, which is the part of geographic information systems knowledge base.

The area objects ontology includes description of the following area objects classes:

- Water objects and hydro technical constructions;
- Settlements;
- Industrial, agricultural, social and cultural objects;
- Road net and road constructions;
- Cover and ground.

The area objects ontology is a classification tree according to hierarchy.

The area objects ontology is a fragment of knowledge base for viewing system class, which is shaped like returnable component and is used in applied geographic information systems design.

The next part of the knowledge base is representation of subject knowledge within the limits of applied system, which is designing, and set of articles, describing certain area objects. Such part of knowledge base we will call "subject" knowledge, in other words, part, where knowledge of subject area are represented.

Formation of "subject" knowledge base is based on evolutionary design principle. This means that this part forms step by step.

CONCLUSION

Suggested method of geographic information systems design according to OSTIS technology let us create applied intelligent systems of this class. What is more, work time contracts because of reusing the following components: area objects ontology, knowledge processing engine and cartographical interface. In common case, designer of applied geographic information system have to form area objects base on specified territory, according to area objects ontology, and form statements in the form of production.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПРИМЕНЕНИЕ НЕЙРОННЫХ И СЕМАНТИЧЕСКИХ СЕТЕЙ ДЛЯ СЕГМЕНТАЦИИ РАСТРОВЫХ ИЗОБРАЖЕНИЙ ЗЕМНОЙ ПОВЕРХНОСТИ

Паркалов А.В.

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

a.parkalov@gmail.com

В статье рассматривается возможность применения нейросетевых и семантических технологий для сегментации растровых изображений земной поверхности в задаче построения векторной карты местности. Описана реализация нейросетевого подхода решения задачи сегментации и первичной классификации сегментов. Выделены предпосылки для использования семантических сетей при классификации сегментов и хранении данных об объектах, преимущества представления векторных карт в виде семантической сети.

Ключевые слова: нейронные сети, семантические сети, векторизация растровых изображений земной поверхности, геоинформационные системы.

ВВЕДЕНИЕ

Широкое использование векторных карт в навигационных системах обусловлено многими факторами, основными из которых являются меньший размер векторного представления картографической информации по сравнению с растровым, возможность масштабирования и изменения формы представления объектов, что облегчает анализ карты конечным пользователем.

Основным недостатком векторной формы представления является высокая трудоемкость и низкая автоматизация процесса векторизации. Часто для облегчения труда векторизируются топографические планы местности, составленные по данным геодезической разведки, такие автоматизированные и автоматические системы существуют [Самардак, 2005], однако следует отметить, что с момента получения первичных данных до построения векторной карты может пройти достаточно большое время и такая карта устареет раньше момента создания. Очевидным решением является совмещение векторной карты и растрового изображения, полученного с помощью аэрофотосъемки или космической съемки земной поверхности соответствующего региона. Недостатком такого подхода является необходимость совершать двойную работу: хранить и обрабатывать как векторное, так и растровое представление одного и того участка местности.

В данной работе рассматриваются некоторые вопросы, касающиеся второго подхода к проблеме

релевантности карты – автоматизации построения векторных карт из растровых изображений земной поверхности.

Съемка земной поверхности, в т.ч. аэрофотосъемка и космическая съемка, позволяет получить детальное изображение поверхности, однако поиск и выделение информации на таком снимке требует непосредственного участия человека. Основными этапами векторизации являются выделение объектов и их границ на растровых изображениях и классификация найденных объектов.

Задача распознавания объектов на растровых изображениях земной поверхности состоит в поиске участков изображения, отличающихся своими атрибутами и представляющих собой некоторые объекты геоинформационных систем, определении их границ, отнесении их к тому или иному классу объектов. В том случае, если векторная карта заданного участка местности уже существует, поиск в окрестности местоположения известного объекта позволит уточнить координаты и атрибуты объекта. Такого рода задача может возникнуть вследствие изменения местности как в результате хозяйственной деятельности человека, так и в результате природных явлений. Примером может служить прокладка дорог, высадка и вырубка леса, возведение зданий и сооружений, изменение русла рек, затопление низинных земель и др. В случае, если карты заданного участка нет, требуется классификация каждого выделенного на местности объекта.

Автоматическая сегментация изображений

Задачи автоматической сегментации растровых изображений делятся на два класса:

- выделение областей изображения с известными свойствами;
- разбиение изображения на однородные области.

В случае работы с изображениями земной поверхности, задача сегментации относится ко второму классу, при этом никакая априорная информация о свойствах областей не используется, зато на само разбиение изображения накладываются некоторые условия (например, все области должны быть однородны по цвету и текстуре). Методы этого класса универсальны и применимы к любым изображениям.

Задача разбиения изображения на однородные области является по своей природе некорректной, так как далеко не всегда для изображения есть единственно «правильная» сегментация, и далеко не всегда задача сегментации имеет единственное решение. По той же причине нет и объективного критерия оценки качества разбиения изображения [Барина и др., 2006].

Качество работы метода оценивается в зависимости от того, насколько полученная сегментация обладает набором определенных свойств. Наиболее часто используются следующие свойства [Haralick et al, 1985]:

- однородность регионов (однородность цвета или текстуры);
- непохожесть соседних регионов;
- гладкость границы региона;
- маленькое количество мелких «дырок» внутри региона.

Основные подходы к сегментации изображений [Барина и др., 2006]:

- кластеризация цветового пространства;
- выращивание регионов;
- дробление-слияние регионов;
- моделирование изображения марковским полем;
- методы, основанные на операторах выделения краев;
- методы теории графов;
- оптимизационный подход.

Результатом сегментации является набор сегментов – связанных областей точек, обладающих сходными свойствами. Картографический объект может состоять из нескольких сегментов, одни объекты могут быть частично перекрыты другими.

В настоящее время технологии достаточно просто позволяют получить растровые изображения земной поверхности с разрешением до 0,2 м на пиксель изображения. На характер изображений влияют такие факторы, как освещенность, угол наклона, под которым производится съемка, используемое оборудование, сезон года. Цветовая

гамма изображения от одного участка к другому как правило изменяется незначительно, природные объекты имеют нерегулярную структуру, антропогенные как правило имеют большие перепады яркости (блики и тени в ясную погоду). Всё это затрудняет выбор наилучшего метода сегментации и ухудшает качество получаемых результатов.

Сегментация с помощью нейронной сети

С точки зрения конечного пользователя, система векторизации растровых изображений должна получать на входе некоторый набор растровых изображений (снимков) земной поверхности с географической привязкой, производить их обработку: выделение объектов на местности, границ и атрибутов этих объектов, осуществлять сохранение полученных данных в формате, удобном для построения векторных карт. Эта задача сложна и в автоматическом режиме на сегодняшний день окончательно не решается. Повышение степени автоматизации требует комплексного подхода к проблеме.

Вывод, сделанный в работе [Skarbek et al, 1994] дает основания предполагать, что повышение качества сегментации и распознавания объектов может быть достигнуто путем объединения этапов сегментации изображений и классификации сегментов. Действительно, точные границы многих картографических объектов могут быть определены человеком-экспертом только после отнесения данного конкретного участка местности к тому или иному типу объекта, и наоборот, установить тип и свойства объекта значительно проще, имея его точные границы. Большинство алгоритмов сегментации изображений работает с изображением на низком уровне, использует один или несколько критериев оценки принадлежности соседних пикселей одной и той же области, например, градиент яркости, и не может выполнять классификацию получаемых сегментов. Это не означает, что от данных методов следует отказываться, но их следует дополнить методами, основанными на классификации фрагментов изображения.

В данной статье предлагается использовать методы выделения краев для сегментации изображения на множество мелких фрагментов. Как хорошо зарекомендовавшие себя методы, они используются в системах обработки видео в реальном времени и не требовательны к ресурсам. Для объединения сегментов в регионы и первичной классификации предлагается применить многослойную нейронную сеть с обратным распространением ошибки. Нейронные сети широко применяются для решения таких задач, как классификация объектов на снимке, поиск объектов заданного типа с учетом изменения местоположения объекта и аффинных преобразований, и хорошо работают тогда, когда объекты заметно отличаются друг от друга, от фона, или когда число

различаемых классов невелико [Szeliski, 2010].

Картографическая информация на снимке земной поверхности как правило представлена большим числом объектов и классов объектов, которые имеют сходную текстуру, яркость, контрастность, имеют небольшую разницу в цвете, нечеткие границы, что затрудняет их прямой поиск и классификацию, поэтому прямое использование известных нейросетевых методов распознавания объектов для поставленной задачи нецелесообразно как в плане требуемых вычислительных ресурсов, так и в плане возможности обучения нейросети для успешного решения задачи. Двухступенчатый подход к распознаванию объектов, когда на первом шаге производится сегментация и первичная классификация сегментов, а на втором – более детальная классификация с уточнением атрибутов полученных объектов, позволит улучшить качество распознавания объектов и их границ.

В качестве первичной классификации были приняты следующие типы рельефа:

- лесной покров (леса, парки);
- травяной покров (поля, сельхозугодия, пустыри);
- техногенный рельеф (дороги, здания, сооружения).

Обработка изображения нейронной сетью производилась методом скользящего окна 16x16 пикселей с перекрытием 8 пикселей. В качестве обучающей выборки было использовано изображение земной поверхности с тремя типами рельефа, классифицированное вручную. Размер изображения 740x723 пикселя, изображение приведено к оттенкам серого по формуле (1):

$$k = (r \cdot 11 + g \cdot 16 + b \cdot 5) / 32, \quad (1)$$

где k – значение яркости пикселя приведенного изображения, r , g , b – соответственно красная, зеленая, синяя компоненты исходного пикселя.

Наилучший результат из рассмотренных показала нейронная сеть следующего вида:

- входной слой состоит из 256 нейронов (по числу пикселей окна сканирования);
- первый внутренний слой состоит из 12 нейронов с сигмоидальной функцией активации;
- второй внутренний слой состоит из 3 нейронов с сигмоидальной функцией активации;
- выходной слой состоит из 3 нейронов с линейной функцией активации;
- в качестве результата принимается максимальное значение выходного слоя сети.

Описанная выше нейронная сеть показала себя склонной к переобучению, большинство ошибок сети связано с выделением густой тени на краю лесных массивов в класс «техногенный рельеф».

Возможные причины:

- класс «техногенный рельеф» на использованном снимке преимущественно состоит

из чередования бликов и теней, сеть обучается на распознавание перепадов яркостей;

- цветовой оттенок дорог, зданий, сооружений является наиболее информативным признаком и был отсеян приведением изображения к оттенкам серого;

- обучающая выборка классифицирована неверно;

- число классов, на которые сеть способна эффективно разделить входные данные, больше трех, или же классы выделены неверно, промежуточные результаты приводят к неопределенности.

Повышение качества классификации нейронной сети может быть достигнуто при наращивании обучающей выборки, так как сеть является многослойной, нелинейной, число связей сети достаточно велико.

Применение семантических технологий

Знание априорной информации о классифицируемом объекте часто позволяет уточнить неопределенный результат классификации. Примером может служить тень высокого дерева посреди лесного массива, которая может быть классифицирована как участок с техногенным рельефом, однако сравнив вероятность нахождения одиноко стоящего объекта в лесу и вероятность ошибки классификатора, можно избежать ошибки классификации. Представление векторной карты в виде семантической сети имеет следующие преимущества:

- определение для такой сети количественных, качественных атрибутов узлов и набора отношений (описывающих взаимное положение сегментов и свойства, на основе которых была произведена сегментация), позволит произвести объединение сегментов, принадлежащих одному объекту;

- использование семантической сети в качестве хранилища данных позволит осуществлять интеграцию и слияние различных наборов данных унифицированным способом [Самодумкин и др, 2011];

- применение знаний, содержащихся в базе, позволит на этапе обучения нейронной сети увеличить репрезентативность классов обучающей выборки;

- использование семантической классификации сегментов исходного изображения позволит определить число классов возможных объектов для каждого изображения и увеличить эффективность обучения нейросети.

С точки зрения логической архитектуры системы векторизации растровых изображений земной поверхности, семантическая сеть является хранилищем знаний об объектах, предметной области. Обращение к базе знаний должно осуществляться:

- после проведения этапа первичной

классификации регионов, когда необходимо использовать априорную информацию о предметной области для отсеивания маловероятных результатов классификации;

- на этапе вторичной классификации для уточнения атрибутов объектов;

- по окончании вторичной классификации для сохранения результатов в виде, удобном для построения векторных карт.

На последнем этапе, когда в базе знаний появляются знания о новых объектах и связях между ними, может производиться их анализ и реорганизация с той целью, чтобы в дальнейшем при векторизации изображений использовались наиболее релевантные знания о предметной области.

ЗАКЛЮЧЕНИЕ

Сегментация изображений земной поверхности является сложной задачей. Существует множество подходов к ее решению, однако наибольший интерес вызывают методы, объединяющие сегментацию и классификацию объектов. Одним из перспективных путей решения задачи является совместное использование алгоритмов выделения краев и нейросетевых алгоритмов для классификации и объединения сегментов изображения в регионы, а так же использование семантических сетей для хранения и обработки статистической и прочей априорной информации о геоинформационных объектах.

Интегрирование этих подходов позволит улучшить результативность процесса сегментации и выделения объектов на растровых изображениях земной поверхности в задаче построения векторных карт, повысить степень автоматизации процесса векторизации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Самардак, 2005] Самардак, А.С. Геоинформационные системы: Учебное пособие. / А.С. Самардак. Владивосток: ТИДОТ ДВГУ, 2005. - 123 с.

[Барнинова и др., 2006] Барнинова, О. Методы сегментации изображений: автоматическая сегментация [Электронный ресурс] / О. Барнинова, А. Вежнев // Графика и Мультимедиа. Научно-образовательный сетевой журнал. Выпуск №4(4), 2006. - Режим доступа: <http://cgm.computergraphics.ru/content/view/147>. - Дата доступа: 6.12.2011.

[Haralick et al, 1985] Haralick, R. M. Image Segmentation Techniques. / R. M. Haralick, L. G. Shapiro // Computer Vision, Graphics, and Image Processing, Vol 29, No 1, 1985.

[Skarbek et al, 1994] Skarbek, W. Colour Image Segmentation: A Survey [Электронный ресурс] / W. Skarbek, A. Koschan. - 1994. Режим доступа: imaging.utk.edu/~koschan/paper/coseg.pdf. Дата доступа: 7.12.2011.

[Самодумкин и др, 2011] Самодумкин, С.А. Семантическая технология компонентного проектирования интеллектуальных геоинформационных систем. / С.А. Самодумкин, С.И. Сорока, А.И. Махина, А.С. Глазунов. // Материалы международной научно-технической конференции OSTIS-2011, 505-514 стр.

[Szeliski, 2010] Szeliski, R. Computer Vision: Algorithms and Applications [Электронный ресурс] / R. Szeliski. - 2010. Режим доступа: <http://szeliski.org/Book/>. - Дата доступа: 6.12.2011.

APPLICATION OF NEURAL AND SEMANTIC NETWORKS IN THE SEGMENTATION OF THE EARTH'S SURFACE BITMAPS

Parkalov A.V.

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

a.parkalov@gmail.com

This article considers the possibility of using neural and semantic technologies to segment raster images the earth's surface in the problem of constructing a vector map. Implementation of the neural network approach in solving the problem of segmentation and primary classification of the segments is described. The preconditions for the use of semantic networks in the segments classification and storing objects data and the advantages of presenting vector maps in the form of a semantic network are identified.

INTRODUCTION

The widespread use of vector maps in navigation systems is the result of many factors, the most important reasons are smaller vector representations of cartographic information, scalability and object representation alterability. The main drawback is the high complexity and low automation of the vectorisation process.

MAIN PART

The main steps of vectorisation are the selection of the objects, defining their boundaries on raster images and classification of objects found.

Improving the quality of segmentation and object recognition can be achieved by combining image segmentation and segment classification stages. It is expedient to use the edge detection algorithm for raster image segmentation and a multilayer neural network with back propagation to join the segments into the regions. In this case object recognition is divided into two stages. Neural network is also applicable in the initial segments classification.

Semantic network can be used as a repository of knowledge about the objects of the domain. Appeal to the knowledge base can be used to improve initial classification, to perform secondary classification.

CONCLUSION

One of the promising approaches in Earth surface images vectorisation is to merge the stages of segmentation and classification. It can be achieved using neural and semantic technologies. Integration will improve the efficiency of the bitmap segmentation and object classification processes in solving the problem of constructing vector maps.



OSTIS-2012

(Open Semantic Technologies for Intelligent Systems)

УДК 621.876.11 + 519.872.4

АЛГОРИТМЫ УПРАВЛЕНИЯ ЛИФТАМИ С ИСПОЛЬЗОВАНИЕМ СЕМАНТИЧЕСКОЙ ИНФОРМАЦИИ

Кузнецов А.П., Никонов В.Н., Шмарловский А.С., Силивонец М.В.

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

kuznap@bsuir.by

nrkn@mail.ru

sas@bsuir.by

a702@tut.by

Рассмотрены наиболее распространенные алгоритмы управления группами лифтов. Анализ эффективности алгоритмов осуществлен с использованием специально разработанной программной платформы. Описаны основные факторы, влияющие на пассажиропоток. Разработан универсальный алгоритм диспетчерского управления группой лифтов, эффективность которого подтверждена при помощи моделирования.

Ключевые слова: алгоритмы управления, лифты, управление группой лифтов, диспетчерское управление.

ВВЕДЕНИЕ

Быстрый рост городов, увеличение ценности земли являются причиной устойчивой тенденции роста этажности городской застройки. По мере увеличения этажности зданий возрастают требования к надежности, комфортности, качеству изготовления, монтажа и эксплуатации лифтов, их энергоэффективности. В крупных зданиях с интенсивным пассажиропотоком эффективность обслуживания пассажиров системой лифтов имеет большое значение. Минимизация времени ожидания лифта после вызова и времени доставки пассажиров на требуемый этаж является актуальной задачей. Несмотря на то, что лифты уже давно стали обычным явлением в повседневной жизни, задача распределения кабин лифта по вызовам таким образом, чтобы время ожидания и поездки было минимальным, далека от тривиальной.

Любую систему, в которой поток требований встречает ограниченные средства их удовлетворения, можно рассматривать как систему массового обслуживания (СМО). Если моменты поступления требований или продолжительность их обслуживания не регламентируются, то при пользовании системой возникают конфликты и образуется очередь. Длина этой очереди зависит от двух характеристик потока требований: интенсивности поступления требований и статистических флуктуаций этой интенсивности [Клейнрок, 1979]. Если интенсивность поступления

требований превышает пропускную способность системы, система не справляется с потоком требований и начинает расти очередь неограниченной длины. Но даже если интенсивность поступления требований меньше пропускной способности системы, очередь может образовываться из-за статистических флуктуаций и внезапного накопления требований. Влияние таких колебаний увеличивается, если средняя нагрузка приближается к пропускной способности системы.

В последнее время получили широкое распространение микропроцессорные системы управления лифтами, позволяющие реализовать достаточно сложные алгоритмы функционирования лифта или группы лифтов, позволяющие минимизировать время ожидания и добиться максимальной производительности с учетом направленности и напряженности пассажиропотока в здании. Существует большое количество типовых алгоритмов диспетчерского управления. Каждый из них имеет свои сильные и слабые стороны, проявляющиеся в различных условиях при использовании в зданиях с различной интенсивностью пассажиропотока. Целью работы является разработка универсального алгоритма, сочетающего в себе наиболее сильные стороны стандартных алгоритмов и обладающего интеллектуальной составляющей, позволяющей оптимизировать работу группы лифтов в контексте текущих условий. При этом алгоритм не должен быть требовательным к вычислительным ресурсам. Оптимальное использование ресурсов группы

лифтов конкретного здания должно достигаться настройкой диспетчерских алгоритмов за счет учета семантической информации. Улучшение качества работы должно обеспечиваться за счет перераспределения назначенных одиночным лифтам вызовов в зависимости от изменения обстановки и текущего характера пассажиропотока. Характер пассажиропотока заранее неизвестен, однако его можно спрогнозировать за счет учета изменений месторасположения некоторых объектов в здании, текущего времени (времена обедов, будний или праздничный день, день недели, сезон года и т. п.), погоды, накопленной статистики и т. д. Анализ этой информации не является тривиальной задачей.

1. Система управления группой лифтов

Лифты представляют собой системы массового обслуживания, в которых алгоритмы диспетчерского управления используются для распределения кабин по вызовам, полученным от пассажиров. Здание определяется как система из N этажей и L кабин лифта. Кабины с индексом $\{1, 2, \dots, L\}$ либо стоят на этажах $\{1, 2, \dots, N\}$, либо движутся вверх или вниз. Вызовы с этажа инициируются пассажирами, желаемое направление движения определяется использованием соответствующей кнопки. Вызовы из кабины инициируются пассажирами, которые вошли в лифт и нажимают кнопку этажа внутри кабины. Вызванный этаж соответствует этажу, на котором пассажир хочет выйти из лифта. Вызовы с этажа регистрируются контроллером, управляющим работой группы лифтов. Контроллер назначает кабину для обслуживания данного вызова на основе диспетчерского алгоритма. Назначение может быть задержано до тех пор, пока не появится лифт с достаточными возможностями для обслуживания вызова. Вызовы из кабины немедленно попадают на контроллер лифта, в задачу которого входит сортировка вызовов, назначенных групповым контроллером, и формирование внутренней последовательности обслуживания. Если назначено обслуживание хотя бы одного вызова, лифт движется в направлении целевого этажа, то есть первого вызова в последовательности обслуживания. Обобщенная структурная схема данной системы приведена на рис. 1.

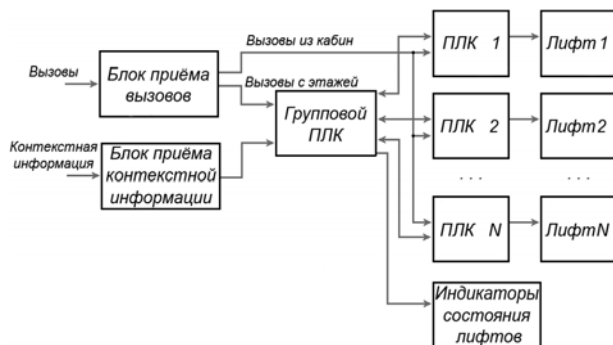


Рисунок 1 – Обобщенная структурная схема системы управления группой лифтов

Диспетчерское управление лифтами осуществляется на двух уровнях: контроллер группы лифтов и контроллер лифта. Контроллер группы лифтов отвечает за работу всех групп лифтов и осуществляет глобальное планирование. Как только вызов назначен для обслуживания определенным лифтом, соответствующий контроллер лифта выполняет локальное планирование в рамках одного лифта. Диспетчерское управление на уровне контроллера лифта обычно осуществляется классическим способом обслуживания в одном направлении: лифт отвечает на все вызовы в текущем направлении своего движения до тех пор, пока последний этаж по вызовам не окажется этажом изменения направления движения. После этого лифт начинает идти в противоположном направлении, обслуживая все вызовы по ходу нового направления движения до тех пор, пока вновь не будет достигнут этаж изменения направления движения.

2. Программный комплекс

Для определения значений параметров эффективности диспетчерских алгоритмов проведено моделирование системы управления группой лифтов. Проведение эксперимента с реально существующей группой лифтов требует больших затрат, поэтому была разработана библиотека классов C++, моделирующая группу лифтов, и создана компьютерная программа на ее основе. Интерфейс программы (рис. 2) предусматривает возможность ввода такой информации как количество лифтов и этажей в здании, скорость лифта, время нахождения лифта на этаже с открытыми дверями, время открывания дверей, направление и интенсивность пассажиропотока в рассматриваемый период, максимально возможное количество людей в здании.

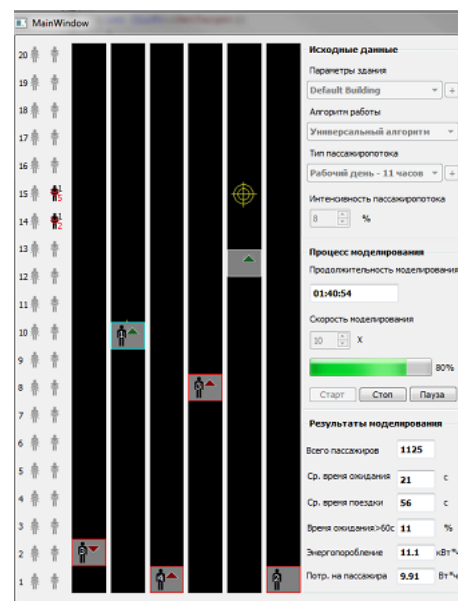


Рисунок 2 – Интерфейс программы
Скорость прибытия пассажиров моделируется

как случайная величина с распределением Пуассона. Программа производит моделирование работы системы управления исходя из перечисленных входных параметров и генерируемых потоков пассажиров. Результатом работы программы является расчет следующих параметров системы:

- общее количество пассажиров, воспользовавшихся лифтом;
- среднее время ожидания кабины лифта;
- среднее время нахождения в лифте;
- процент пассажиров, которые не были обслужены по истечении одной минуты;
- энергопотребление, в том числе в расчете на одного пассажира.

Графический пользовательский интерфейс позволяет строить графики мгновенного и усредненного энергопотребления, усредненного времени ожидания и доли пассажиров, не обслуженных в течение минуты.

3. Обзор наиболее распространенных алгоритмов диспетчерского управления

Ниже приведены алгоритмы, получившие наибольшее распространение в системах диспетчерского управления на уровне контроллера группы лифтов.

Преимуществами кругового алгоритма является простота реализации, равномерная загрузка каждого лифта, а также обеспечение приемлемого обслуживания пассажиров в условиях неинтенсивной загрузки. Основной целью круговой системы при диспетчерском управлении группой лифтов является достижение равной нагрузки на каждый лифт. Вызовы распределяются по мере их поступления последовательным образом по отдельным лифтам. Вызов 0 назначается к обслуживанию кабиной 0, вызов 1 – кабиной 1, вызов L – кабиной 0, вызов $L+1$ – кабиной 1 и т. д.

Частным случаем кругового алгоритма является диспетчерское управление при максимальном потоке вверх. Отличием данного варианта является использование особой стратегии парковки: если групповой контроллер обнаруживает простаивающий лифт, он инициирует вызов с первого этажа для отправки кабины на этаж с предположительно максимальным потоком пассажиров, для уменьшения времени ожидания будущих пассажиров.

В некоторых зданиях используют идею зонирования, заключающуюся в разделении здания на несколько прилегающих друг к другу зон, причем каждый из лифтов обслуживает вызовы с этажей только той зоны, которая назначена для обслуживания данным лифтом.

Алгоритм трех переходов [Ронг и др, 2003] представляет собой частный случай алгоритма управления группой лифтов с оценкой времени

прибытия. Он используется для определения последовательности обслуживания вызовов с этажей. Вызовы прохода 1 (P1) лифт может обслужить по пути следования в настоящий момент времени. Вызовы прохода 2 (P2) лифт может обслужить после того, как один раз изменит направление движения. Вызовы прохода 3 (P3) лифт может обслужить после того, как дважды изменит направление движения. Вызовы группы P1 сортируются от ближайшего к дальнему относительно текущего положения лифта и выполняются в этом порядке. Если нет вызовов группы P1, выполняются вызовы группы P2 от дальнего к ближнему. При отсутствии вызовов первых двух групп выполняются вызовы группы P3 в порядке от дальнего к ближнему.

4. Итерационный алгоритм

Одним из резервов улучшения качества работы является перераспределение уже назначенных одиночным лифтам вызовов в зависимости от изменения обстановки. Все множество M запросов, поступающих с различных этажей здания, разбивается на такие непересекающиеся подмножества M_i ($i = 1 \dots n$, где n – число лифтов в группе), чтобы максимальное время обработки i -м лифтом подмножества запросов M_i с учетом уже имеющегося внутреннего задания было минимальным:

$$T = \max \{T_i; i = 1, 2, \dots, n\}, \quad (1)$$

где T – общее время обслуживания всех запросов;

T_i – время обслуживания i -м лифтом подмножества запросов M_i .

Поставленную задачу можно отнести к классу комбинаторных задач, для решения которых существует ряд методов. Однако все эти методы так или иначе требуют полного перебора возможных вариантов, число которых составляет

$$C = (2^n - 1)^m, \quad (2)$$

где $m = |M|$ – число запросов.

Максимальное значение $m_{\max} = 2K - 1$, где K – число этажей в здании. Поэтому:

$$C = (2^n - 1)^{2K-1}. \quad (3)$$

К примеру, если число этажей в здании 20, а число лифтов 8, получаем:

$$C = (2^8 - 1)^{39} \approx 2^{312}. \quad (4)$$

Очевидно, что для большинства контроллеров такой полный перебор в реальном времени изменения ситуации (изменения множества запросов, внутренних заданий и положения лифтов) невозможен. Рассмотрим другой подход к решению проблемы комбинаторного целераспределения,

основывающийся на стратегии распределенного или коллективного принятия решений [Каляев, 2001]. В подходе используется итерационный алгоритм оптимизации, суть которого заключается в последовательном выборе каждым объектом, входящим в группу, такой цели, которая дает экстремальное приращение целевого функционала при фиксированном выборе всех остальных объектов группы. Итерационный процесс оптимизации продолжается до тех пор, пока в двух последовательных циклах итерации значение целевого функционала не изменяется. Для учета происходящих изменений ситуации процедура периодически повторяется заново, что позволяет оперативно оптимизировать распределение.

Описанная выше процедура является асимптотически сходящейся в смысле определения Ляпунова, причем число итерационных циклов D , за которые данная процедура сходится

$$D \leq \frac{|T_{i\dot{a}}^{\max} - T_{i\dot{a}}^{\min}|}{\Delta T_{\min}}, \quad (5)$$

где $T_{об}^{\max}$ – максимально возможное значение времени обслуживания;

$T_{об}^{\min}$ – минимально возможное значение времени обслуживания;

ΔT_{\min} – минимально возможное изменение времени обслуживания при перераспределении запросов между лифтами.

Значения $T_{об}^{\max}$, $T_{об}^{\min}$ и ΔT_{\min} можно оценить, исходя из следующих соображений:

1. Максимально возможное время обслуживания может быть в том случае, если все запросы последовательно будет обслуживать единственный лифт, причем в наихудшем случае общее число обслуживаемых запросов будет $2K$, где K – число этажей в здании. Поэтому:

$$T_{об}^{\max} \leq 2\tau(K + k \cdot K) = 2\tau K(1 + k). \quad (6)$$

2. Минимально возможное время обслуживания будет достигаться в случае, если каждый из лифтов выполняет единственный запрос, перемещаясь при этом на один этаж:

$$T_{об}^{\min} \geq \tau(1 + k). \quad (7)$$

3. Минимально возможное изменение времени обслуживания при перераспределении запросов достигается в случае, если один лифт «забирает» у другого один запрос. При этом общее время обслуживания в минимальном случае будет уменьшаться на время посадки-высадки пассажиров $k\tau$ или время, в течение которого лифт проезжает один этаж τ , в зависимости от того, какая из этих величин будет меньше для конкретного здания:

$$\Delta T_{\min} \geq \min\{k\tau, \tau\}. \quad (8)$$

Отсюда получаем

$$D \leq \frac{(2\tau K(1 + k) - \tau(1 + k))}{\min\{k\tau, \tau\}} = \frac{(1 + k)(2K - 1)}{\min\{k, 1\}}. \quad (9)$$

Описанная выше процедура оптимизации распределения запросов между лифтами обладает рядом преимуществ по сравнению со стандартными процедурами, основывающимися на полном переборе. Значительно снизится число анализируемых вариантов распределения запросов:

$$C = D \cdot n \cdot m_{ig}^j \leq D \cdot n \cdot K, \quad (10)$$

где $m_{ig}^j = |M_{ig}^j|$ – число запросов, включенных в множество допустимых запросов для i -го лифта на j -ом шаге итерации.

Подставляя (9) в (10), получаем

$$C \leq \frac{(k + 1)(2K - 1)}{\min\{k, 1\}} \cdot n \cdot K. \quad (11)$$

Если, например, $K=20$, $n=8$, $k=1$, получаем: $C \leq 24960$, то есть число анализируемых вариантов по сравнению со случаем полного перебора снижается приблизительно в $3,3 \cdot 10^{89}$ раз. Это позволяет реализовать описанную процедуру в реальном времени изменения ситуации с помощью большинства распространенных вычислительных устройств. Разумеется, такое существенное сокращение анализируемого числа вариантов достигается за счет отказа от достижения глобального оптимума целевой функции (достигается некоторый локальный оптимум). Однако следует отметить, что из-за быстрого и непредсказуемого изменения ситуации (например, появление новых вызовов) и неизвестного заранее точного времени посадки пассажиров по уже имеющимся вызовам, глобальный оптимум теряет свою актуальность. В таких условиях имеет смысл как можно чаще искать некий «рациональный» оптимум, что и обеспечивает предложенный итерационный алгоритм.

В программе моделирования реализован модуль сбора статистики, учитывающий информацию о каждом пассажире. Групповой контроллер реальной лифтовой системы такой информации иметь не может, однако ему доступна информация о времени регистрации и выполнения вызовов. Анализ этой информации может с определенной долей достоверности помочь определить характер пассажиропотока. Качественные критерии работы итерационного алгоритма определяются выбором критерия передачи вызова от более загруженного лифта к менее загруженному. Рассмотрены и промоделированы следующие варианты:

- Оптимизация общего времени выполнения вызовов. Условием перестановки в данном случае является возможность минимизации общего времени выполнения активных вызовов. То есть из всех вариантов перестановки двух вызовов выбирается тот, который обеспечивает

максимальное уменьшение общего времени выполнения текущих вызовов. Перестановка осуществляется для всех лифтов по очереди.

- Оптимизация общего времени выполнения вызовов с сортировкой. В отличие от предыдущего варианта для перестановки на каждом шаге выполняется сортировка лифтов по времени выполнения назначенных им вызовов. Для перестановки выбираются вызовы, назначенные самому загруженному лифту.

- Оптимизация конкретного вызова по времени выполнения. У лифта с наибольшей загрузкой выбирается вариант перестановки какого-либо вызова по критерию максимального уменьшения времени выполнения вызова, то есть из всех вариантов перестановки выбирается пара лифтов, которая дает максимальное уменьшение времени выполнения вызова. Данная оптимизация рассмотрена как с контролем общего времени выполнения вызовов (в качестве условия перестановки используется условие, запрещающее в результате перестановки увеличивать общее время выполнения вызовов), так и без него.

- Оптимизация конкретного вызова по пройденному расстоянию. К описанному выше добавляется требование перестановки вызовов только в случае уменьшения пути, который должна пройти кабина лифта для выполнения вызова.

Моделирование показало, что оптимизация конкретного вызова по времени выполнения значительно превосходит оптимизацию по общему времени вызовов. Это объясняется тем, что оптимизация общего времени вызовов в обстановке быстрого притока новых пассажиров, а следовательно, быстрого изменения общей картины вызовов не способна обеспечить приемлемого результата. Ситуация меняется столь быстро, что долгосрочное прогнозирование теряет смысл. При этом краткосрочный прогноз при условии оптимизации каждого конкретного вызова дает положительный результат. Оптимизация вызова по пройденному расстоянию показывает худшее быстрое действие, но меньшие энергозатраты по сравнению с оптимизацией вызова по времени выполнения. В дальнейшем эти алгоритмы упоминаются как итерационный алгоритм с оптимизацией скорости и итерационный алгоритм с оптимизацией экономичности соответственно. Алгоритм, обеспечивающий экономичность, получается из алгоритма оптимизации скорости с помощью дополнительного условия перестановки вызовов в итерации. Вариант с коэффициентом, регулирующим приоритет производительности или экономичности, реализуется за счет переключения алгоритмов с заданной периодичностью.

Для определения критериев оценки алгоритмов использовался типичный для офисного здания 11-часовой прогон, сочетающий в себе интенсивный поток вверх в начале прогона, интенсивный поток вниз в конце, постоянный межэтажный поток невысокой интенсивности и суперпозицию потоков

вверх и вниз высокой интенсивности в середине прогона (табл. 1). В рассматриваемом здании 12 этажей, 4 лифта, высота этажа 4 метра, пиковая интенсивность прибытия пассажиров 8 человек в минуту.

Таблица 1 – Результаты моделирования

Алгоритм	$T_{ож},$ с	$T_{п},$ с	$\%_{>мин}$	$P_{на}$ пасс., Вт
Кольцевой алгоритм	48	46	17	12,1
Алгоритм трех переходов	24	46	10	12,4
Максимальный поток вверх	23	45	13	16,1
Алгоритм зонирования	33	49	20	11,7
Итерационный алгоритм	12	46	2	11,3
Итерационный алгоритм (экономичный)	15	47	5	10,4

Результаты моделирования демонстрируют преимущество итерационного экономичного алгоритма над ближайшим по результатам на 11% по экономичности и на 9% по производительности над наиболее быстрым из стандартных – алгоритмом трех переходов. Итерационный алгоритм, адаптированный по скорости, имеет преимущество на 15% по производительности над алгоритмом трех переходов и на 3,4% по экономичности над алгоритмом зонирования. Итерационный алгоритм с оптимизацией по экономичности на 6,5% менее производителен, однако на 8% экономичнее алгоритма, оптимизированного по скорости.

5. Использование семантической информации

Важным преимуществом итерационного алгоритма является возможность значительно влиять на его качественные показатели за счет изменения или расширения условий перестановки вызова от более загруженного лифта к менее загруженному. Это позволяет гибко настроить алгоритм в зависимости от характера пассажиропотока и учесть закономерности его изменения. Отдельные здания могут иметь этажи с повышенной интенсивностью пассажиропотока: например, первый этаж; этаж, на котором располагаются объекты общественного питания – во время обеденного перерыва; этажи, на которых имеются переходы, соединяющие несколько смежных зданий; открытые либо закрытые смотровые площадки высотных зданий и т. д. Отдельные этажи имеют постоянные либо временные (зависящие от времени суток или погодных-климатических условий – как, например, посещаемость открытой смотровой площадки и т. п.) всплески интенсивности заданий.

При наличии информации о повышенной интенсивности вызовов с таких этажей не

составляет труда повысить приоритет этих вызовов, введя в условие перестановки вызовов на очередном шаге итерации пониженные весовые коэффициенты для менее приоритетных вызовов. Таким образом, у приоритетных вызовов появится преимущество распределения по отношению к наименее удаленным и загруженным лифтам. Также возможна отправка части свободных лифтов на приоритетные этажи даже при отсутствии в конкретный момент времени вызовов с этих этажей.

В табл. 2 приведены результаты моделирования работы итерационного алгоритма в режиме максимального потока вверх с учетом приоритета вызовов с первого этажа и без учета приоритетов. Интенсивность прибытия пассажиров при моделировании составляет 6 человек в минуту.

Таблица 2 – Результаты моделирования

Алгоритм	$T_{ож, с}$	$T_{п, с}$	$\%_{>мин}$	$P_{на пасс, Вт}$
Равный приоритет вызовов	22	58	9	15,2
Приоритет первого этажа	7	49	0	19,5

Как видно из таблицы, учет особенностей пассажиропотока обеспечил прирост производительности группы лифтов на 30% при увеличении затрат электроэнергии на 20%.

ЗАКЛЮЧЕНИЕ

Разработана платформа для оценки эффективности алгоритмов диспетчерского управления. Разработана методика, позволяющая оценить эффективность реализованного алгоритма для условий конкретного здания и найти разумный компромисс между экономичностью и качеством обслуживания пассажиров. Рассмотрены основные современные алгоритмы управления группами лифтов. Осуществлено исследование достоинств и недостатков данных алгоритмов в различных условиях работы с разными типами пассажиропотоков. Разработан эффективный итерационный алгоритм диспетчерского управления с возможностью регулирования приоритетов производительности или экономичности, а также гибкой подстройки под характер пассажиропотока за счет регулирования коэффициентов приоритетности вызовов с различных этажей.

Полученные результаты моделирования показывают, что система управления лифтами выигрывает от перехода к наиболее подходящему алгоритму в условиях доминирующего в данное время пассажирского потока. Для потоков малой интенсивности использование специально разработанных алгоритмов не дает существенного преимущества по сравнению с простейшими алгоритмами, однако с ростом этажности здания, количества лифтов и интенсивности

пассажиропотока эти преимущества становятся достаточно существенными.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Каляев, 2001] Каляев И.А. Использование принципов коллективного принятия решений при управлении группой автоматических лифтов. // Мехатроника, №4, 2001.

[Клейнрок, 1979] Клейнрок Л. Теория массового обслуживания. Пер. с англ./Пер. И.И. Грушко; ред. В.И. Нейман. – М.: Машиностроение, 1979.

[Ронг и др, 2003] Ронг А., Хаконен Х., Ладелма Р. Алгоритм управления группой лифтов на основе расчетного времени прибытия (ETA) с более точной оценкой // Технический отчет 584. Центр компьютерных наук г. Турку (TUUS). Турку, Финляндия, 2003.

ELEVATOR CONTROL ALGORITHMS USING SEMANTIC INFORMATION

Kuznetsov A.P., Nikonov V.N.,
Shmarlouski A.S., Silivonets M.V.

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

kuznap@bsuir.by
nrkn@mail.ru
sas@bsuirby
a702@tut.by

INTRODUCTION(ВВЕДЕНИЕ)

There are a lot of algorithms to control a group of elevators. But almost all of them are developed for specific buildings conditions. The universal control algorithm possibilities of which are extended due to application of intelligent control systems has been considered in this article.

MAIN PART

Standard algorithms of supervisory control have both advantages and disadvantages manifesting in different conditions. The purpose of our development is a universal algorithm combining all advantages of standard algorithms having an intelligent component for optimization of operation of a group of elevators in current conditions.

Improvement of operation is provided by the redistribution of calls assigned for a single elevator which depends on conditions change out on the current character of passenger traffic.

CONCLUSION

Mostly applied control algorithms for controlling groups of elevators have been considered with the help of the developed software platform we have carried out analysis. The main factors affecting the passenger traffic are described. A universal algorithm for supervisory control of a group of elevators has been developed and simulations have proved the effectiveness of this algorithm.



УДК 681.511.4

АНАЛИЗ ФАЗОВОГО ДЕТЕКТОРА ТИПА «ВЫБОРКА-ЗАПОМИНАНИЕ»

Шилин Д. Л.

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

dimashilin@gmail.com

В данной работе автором произведен анализ фазового детектора «выборка-запоминание», рассматривается метод проектирования фазового детектора, делающий возможным применения такого устройства в системах фазовой синхронизации в широком диапазоне частот, особенно удачным видится применение в синтезаторах частот, где предъявляются высокие требования в чистоте выходного сигнала. Рассмотрен ряд микросхем, использующих данный принцип построения фазового детектора.

Ключевые слова: выборка-запоминание, синтезатор частот, устройства фазовой синхронизации, фазовый детектор.

ВВЕДЕНИЕ

Устройства фазовой синхронизации, в силу их эффективности, выделены в отдельный класс систем автоматического регулирования. Их развитию уделяется большое внимание, трудно назвать область техники, где бы они ни применялись, в частности в робототехнике с интеллектуальным управлением. На сегодняшний день в большинстве систем передачи информации используют устройства фазовой синхронизации (УФС) для синхронизации потоков при обмене информацией, для организации высокоточных синтезаторов частот. Использование некоторых режимов позволяет использовать УФС для шифрования информации [Шахтарин, 2006], [Шнайер, 2002]. Ведущие мировые производители разрабатывают различные функциональные блоки УФС, которые включены в стандартные микросхемы и имеют ряд основных характеристик. Таким образом, проектирование УФС на первом этапе сводится к выбору функциональных узлов и соединению их в структурную схему. Наличие методов синтеза и анализа УФС [Кузнецов, 1993] [Goldman, 2007], позволяет использовать классические топологические схемы, а стандартные функциональные микросхемы – придавать топологическим структурам вид электрических схем.

В УФС блоком, определяющим вид модуляции, основные динамические и шумовые характеристики, является фазовый детектор [Кузнецов, 1993] [Романов, 2010].

Автором предлагается при выборе фазовых детекторов, исходя из функционального назначения УФС, производить анализ характеристик на уровне электрических схем. При этом основным инструментом анализа предлагаются математические модели схем и пакет прикладных программ PSPICE.

ОСНОВНАЯ ЧАСТЬ

Необходимо определить три основных характеристики фазовых детекторов (ФД):

Статическую фазовую характеристику $\varepsilon(\varphi)$, т.е. зависимость постоянной составляющей на выходе детектора от фазового рассогласования входных сигналов;

Статический коэффициент передачи ФД:

$$K_{\text{ФД}} = \frac{d\varepsilon}{d\varphi}$$

Уровень помех на выходе ФД – $\varepsilon(K\omega_0)$, где $\omega_0 = 2\pi f_0 k$, $k = 1, 2 \dots$ – частота входного сигнала в установившемся режиме.

На рисунке 1 $y(t)$ – входной сигнал первого входа ФД; $u(t)$ – сигнал из цепи обратной связи УФС (второй вход ФД); $\varepsilon(t)$ – сигнал рассогласования (выход ФД).

Рассмотрим несколько основных видов фазовых детекторов и проведем анализ их характеристик. На рисунке 1 изображена структурная схема фазового детектора типа «выборка-запоминание».

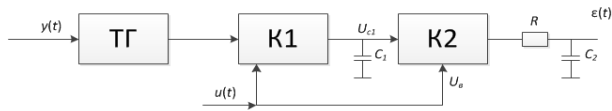


Рисунок 1 – Структурная схема фазового детектора типа «выборка-запоминание»

Где ТГ – токовый генератор; К₁ и К₂ – электронные ключи, работающие в противофазе; С₁ – конденсатор, на котором формируется пилообразное напряжение (или периодическое напряжение другой формы); С₂ – конденсатор, сохраняющий заряд на время между двумя импульсами выборки. Токовый генератор запускается входным сигналом у(t). Форма напряжения на конденсаторе С₁ определяет вид статической характеристики ФД – пилообразная, треугольная, трапецидальная и др. [Wolaver, 2006]. На рисунке 2 представлены временные диаграммы работы детектора.

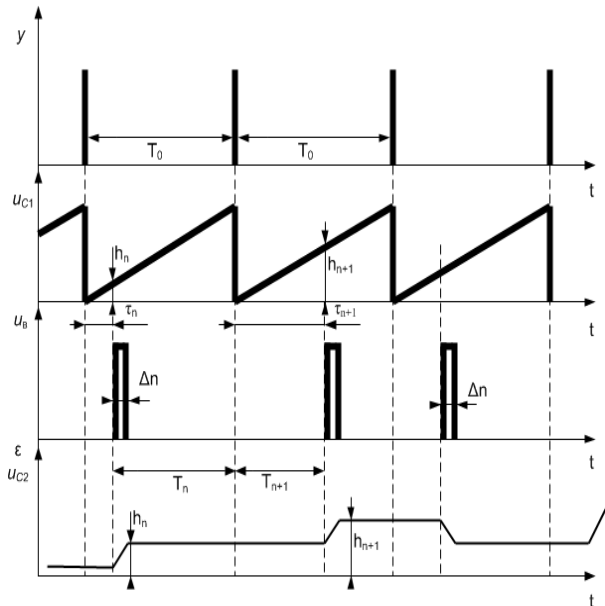


Рисунок 2 – Временные диаграммы работы фазового детектора типа «выборка-запоминание»

ФД работает следующим образом: сигнал выборки, сформированный из u(t), размыкает К₁ и замыкает К₂, что приводит к передаче напряжения на конденсатор С₂. Величина напряжения на С₂ соответствует фазовому рассогласованию сигналов у(t) и u(t). По окончании выборки, этот процесс определяется длительностью импульса выборки Δn, напряжение запоминается на конденсаторе С₂ до прихода следующего импульса выборки.

Математическая модель для данной схемы, при условии, что сопротивление ключей К₁ и К₂ малы, а входное сопротивление каскада, на который нагружен ФД – высокое, будет иметь вид. Пусть U_{C1}(t_n) = у(n), а U_{C2}(t_n) = ε(n-1), тогда на момент прихода импульса выборки (ключ К₁ разомкнут, К₂ – замкнут):

$$\frac{1}{C_2} \int idt + \frac{1}{C_1} \int idt = 0, \quad (1)$$

где i – ток в цепи C₁RC₂.

Обозначим

$$\tau_n = \frac{RC_1C_2}{C_1 + C_2},$$

тогда решение уравнения (1) будет иметь вид:

$$\varepsilon(t) = A_1 + A_2 e^{-\frac{t}{\tau_n}}, \quad (2)$$

где A₁ и A₂ – постоянные интегрирования.

Определение A₁ и A₂ позволяет получить математическую модель ФД «выборка-запоминание» в виде разностного уравнения:

$$\varepsilon[n] = y[n] \frac{(1 - e^{-\frac{\Delta n}{T_n}})}{1 + C_2/C_1} + \varepsilon[n-1] (1 - y[n] \frac{(1 - e^{-\frac{\Delta n}{T_n}})}{1 + C_2/C_1}). \quad (3)$$

Из последнего выражения определим передаточную функцию и статическую характеристику фазового детектора:

$$\varepsilon(\varphi) = \frac{U_n \varphi}{2\pi}. \quad (4)$$

Статический коэффициент передачи:

$$K_{\text{ФД}} = \frac{U_n}{2\pi}. \quad (5)$$

Следует отметить, что ε(φ) и КФД совпадают с аналогичными характеристиками триггерного фазового детектора.

Для анализа уровня проникновения в выходной сигнал побочных составляющих уравнение (3) рассмотрим для режима однократного синхронизма, условием которого является: ε[n] = ε[n-1] = ε*, у[n] = у[n-1] = у*. В этом случае выходной сигнал периодический, с периодом 2π.

Автором проведен анализ включения ФД на частоте сравнения 100 кГц. Фазовый детектор включен на буферный каскад, обладающий большим входным сопротивлением, который работает на низкочастотный фильтр (ФНЧ) с частотой среза f_{ср} = 200 кГц. Схема представлена на рисунке 3, ФНЧ представляет лестничный LC-фильтр с тремя элементами, нагружен на согласованное сопротивление R_н = 1*10³ Ом.

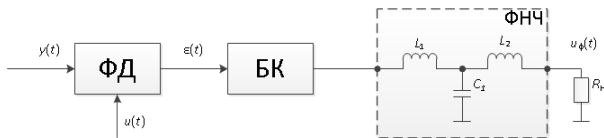


Рисунок 3 – Схема подключения низкочастотного фильтра к ФД «выборка-запоминание»

С помощью пакета PSPICE был произведен анализ включения ФД и выход его на рабочий режим синхронизации, оценен спектр сигнала для $\varepsilon(t)$ и $u_\phi(t)$. На рисунке 4 представлены временные диаграммы работы, на рисунке 4 – фазовое рассогласование между сигналом $y(t)$ и сигналом $u(t)$ пропорционально $0,2 T_0$, а на рис. 5 – $0,8 T_0$.

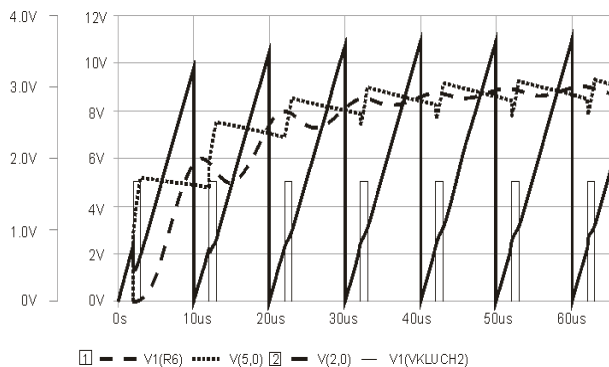


Рисунок 4 – Временные диаграммы работы ФД (фазовое рассогласование $y(t)$ и $u(t)$ пропорционально $0,2 T_0$)

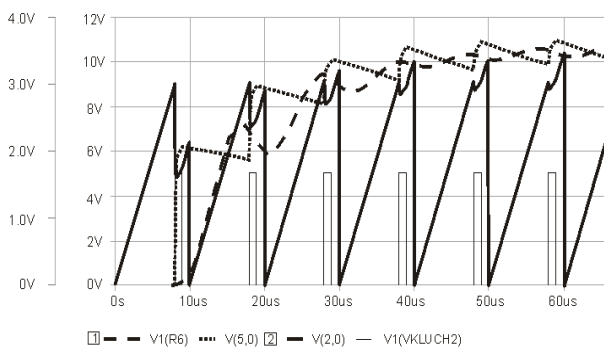


Рисунок 5 – Временные диаграммы работы ФД (фазовое рассогласование $y(t)$ и $u(t)$ пропорционально $0,8 T_0$)

где:

- сигнал пилообразного напряжения $U_{C1}(t)$;
- импульс выборки $U_B(t)$;
- переходное напряжение на конденсаторе $C_2 - U_{C2}(t) = \varepsilon(t)$;
- переходное напряжение на выходе фильтра $U_\phi(t)$.

Рассчитан спектр сигнала $\varepsilon(t)$ и $U_\phi(t)$, для установившегося режима рис. 5 ($\tau^* = 0,8 T_0$) спектр описывается следующим выражением:

$$\varepsilon(t) = 8.887 + 0.304 \sin(2\pi 10^5 t) + 0.145 \sin(4\pi 10^5 t) + 0.085 \sin(6\pi 10^5 t) + 0.054 \sin(8\pi 10^5 t) + 0.035 \sin(10\pi 10^5 t) \quad (6)$$

$$u_\phi(t) = 8.8286 + 0.184 \sin(2\pi 10^5 t) + 0.019 \sin(4\pi 10^5 t) + 0.14 \sin(6\pi 10^5 t) \quad (7)$$

Отношение сигнал полезной составляющей к уровню сигнала помехи будет:

для $\varepsilon(t)$:

$$SNR(dB) = 20 \log\left(\frac{U_1}{\sqrt{U_2^2 + U_3^2 + U_n^2}}\right) = 25 dB \quad (8)$$

где U_1 – действующее значение полезного сигнала; $U_2, U_3 \dots U_n$ – действующее значение сигналов помехи;

для $U_\phi(t)$:

$$SNR(dB) = 20 \log\left(\frac{U_1}{\sqrt{U_2^2 + U_3^2 + \dots}}\right) = 31.5 dB \quad (9)$$

Из выражений (6), (7) и результатов расчета (8), (9) видно, что изменяя схему и параметры ФНЧ можно добиться иных результатов подавления паразитных составляющих сигнала. В [Кузнецов, 1993], [Шилин, 2010] авторы предлагают методы синтеза, где определяются передаточные функции непрерывной линейной части (НЛЧ), в состав которой входит ФНЧ.

ЗАКЛЮЧЕНИЕ

Фазовые детекторы типа «выборка-запоминание» реализованы в микросхемах синтезаторов частот TSA6060 и TSA6060T фирмы Philips Semiconductors. Данная микросхема предназначена для построения цифровых синтезаторов с системой фазовой автоподстройки частоты. В состав микросхемы входят: генератор и делитель опорного сигнала, делитель входной частоты с программируемым коэффициентом деления, фазовый детектор, двухуровневый усилитель тока и контроллер обмена информацией по протоколу I2C. Работает в диапазоне $30 \div 200$ МГц.

С использованием данного принципа построения в микросхеме 5861ДФ1У (научно-технический центр ДЭЛС) выпускается фазовый компаратор, работающий на частоте до 45 МГц.

Таким образом, автором произведен анализ ФД «выборка-запоминание». Сделан вывод о возможности применения такого устройства в системах фазовой синхронизации в широком диапазоне частот, особенно удачным видится применение в синтезаторах частот, где

предъявляются высокие требования в чистоте выходного сигнала. Рассмотрен ряд микросхем, использующих данный принцип построения ФД.

Не секрет, что на сегодняшний день существует необходимость в повышении эффективности компьютерной поддержки проектирования подобных устройств, что позволит ускорить разработку УФС, а также более точно удовлетворить требования, предъявляемые к системе в целом. Таким образом, существует необходимость в создании интеллектуальных систем автоматизированного проектирования устройств с автоматическим регулированием, а в частности УФС, что позволит решить описанную выше задачу.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Кузнецов, 1993] Кузнецов, А.П. Анализ и параметрический синтез систем с фазовым управлением / А.П. Кузнецов, М.П. Батура, Л.Ю. Шилин // Мн.: Наука и техника, 1993. с. 224.
- [Goldman, 2007] Goldman, S. Phase-locked loop engineering handbook for integrated circuits / S. Goldman // Boston/London.: Artech House, 2007. с. 549.
- [Романов, 2010] Романов, С.К. Системы импульсно - фазовой автоподстройки в устройствах синтеза и стабилизации частот / С.К. Романов, Н.М. Тихомиров, А.В. Леньшин // М.: Радио и связь, 2010, с. 328.
- [Wolaver, 2006] Wolaver, Dan H. Phase-locked loop circuit design / Dan H. Wolaver // New Jersey.: Prentice Hall, 2006. с. 266.
- [Шилин, 2010] Шилин, Д.Л. Моделирование режима детерминированного хаоса в системах фазовой синхронизации / Д.Л. Шилин // Новополоцк: Вестник ПГУ, 2010, N 9.
- [Шахтарин, 2006] Шахтарин, Б.И. Случайные процессы в радиотехнике / Б.И. Шахтарин // М.: Гелиос АРВ, 2006, с. 448.
- [Шнайер, 2002] Шнайер, Б. Прикладная криптография: протоколы, алгоритмы, исходные тексты на языке С / Б. Шнайер // М.: Триумф, 2002, с. 815.

ANALYSIS OF SAMPLE-AND-HOLD PHASE DETECTOR

Shilin D.L.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus
dimashilin@gmail.com

In this paper, the author performed an analysis of "sample-and-hold" phase detector, discusses a method of phase detector designing which makes possible to use such devices in phase-locked loop in a broad range of frequencies, particularly successful application is seen in the frequency synthesizer, where high requirements in purity of output signal.

INTRODUCTION

Phase-locked loop device, because of their effectiveness, are grouped in a class of automatic control systems. Their development is given much attention; it's hard to call the technical field, where these devices are not applied. To date, the most of data exchange systems and devices using phase-locked loop devices (PLL) to synchronize the flows of information, for design of precision frequency synthesizer, in

particular in robotics with an intelligent control. Use of some modes allows applying PLL for information encrypting.

MAIN PART

The world's leading manufacturers are developing various functional blocks PLL, which are included in the standard chips and have a number of basic characteristics. Thus, the design of the PLL in the first stage reduces to the choice of functional units and their connection to the block diagram. Availability of methods of synthesis and analysis of the PLL can use the classical topologies, and basic, functional circuits - to give a topological structures form of electrical circuits.

The author offered the choice of the phase detector, based on the functional purpose of PLL, to analyze the characteristics at the level of electrical circuits. Thus, the basic tool for analyzing of mathematical models of circuits is software PSPICE.

CONCLUSION

Phase detectors, such as "sample-and-hold", implemented in the frequency synthesizer TSA6060 and TSA6060T developed by Philips Semiconductors. This chipsets is designed to build a digital synthesizer with a phase-locked loop. The structure of chips includes: generator and divider of reference signal, the input frequency divider with programmable division factor, phase detector, a two-level current amplifier and controller information exchange via I2C. It works in the frequency range of 30 ÷ 200 MHz.

Using this principle of designing was developed a chipset 5861DF1U (Science and Technology Center DELS). It's a phase comparator with working clock at 45 MHz, which producing till now.

It is concluded that there is possibility of using such a devices in phase-locked loop (PLL) in a broad range of frequencies, especially successful application is seen in the frequency synthesizer, where meet high requirements to the purity of the output signal. Given a number of chipsets that been designed using described principle.

It's no secret that today there is a need to increase the effectiveness of computer support of such devices design, which will accelerate the development of the PLL, as well as more accurately meet the requirements for the system as a whole. Thus, there is a need for intelligent CAD systems for automatic control devices designing, and in particular the PLL, which will solve the above problem.

AUTHOR INDEX

Abramov N.V.	129	Davydov A.G.	355
Adzinets D.	135	Dikovitsky V.V.	205
Alexeyev N.A.	103	Dyundyukov V.S.	257
Antipova O.V.	315	Eremeev A.A.	169
Bairak S.	135	Eremeev A.P.	169
Bakhtizin V.V.	453	Ergesh B.J.	397
Baranovich A.E.	91	Erchenko E.A.	513
Bekmanova G.T.	397	Efimenko I.V.	381
Birialtsev E.V.	245	Ivaniuk A.A.	123, 129
Bobkov A.S.	347, 351	Ivashenko V.P.	193
Bodyakin V.I.	83	Kadnikov D.S.	315
Borgest N.M.	335	Kalutskaya A.P.	267
Borovikova O.I.	181	Karabalaeva M.H.	397
Buribaeva A.K.	397	Kataev V.A.	207
Charnine M.M.	373	Khakhalin G.K.	327
Chernov R.V.	335	Kharlamov A.A.	385
Chernyahovskaya M.	473	Khoroshevsky V.F.	143
Curbatov S.S.	327	Kim Y.A.	413
Galimov M.R.	245	Kleschev A.S.	213, 437, 473
Gatiatullin A.R.	391	Kobrinский B.A.	53
Gavrilova T.A.	443	Kolb D.G.	111
Girnyk D.A.	489	Komartsova L.G.	315
Gladun A.	493	Koronchik D. N.	339
Globa L.S.	99, 103, 447, 501	Kozerenko E.B.	365, 373
Golenkov V.V.	23	Kozlov O.A.	507
Golovnja A.I.	419	Kuzin I.A.	501
Gorkina A.A.	175	Kuzmin A.A.	401
Gribova V.	213	Kuznetsov A.P.	531
Guliakina N.A.	23, 221	Kuznetsov I.P.	373
Hetsevich S.A.	401	Kuznetsov O.P.	159
Hetsevich Y.S.	401	Kyselov G.D.	359
Davydenko I.T.	457	Kyselov V.V.	355

Kyselova A.G.	359	Shmarlouski A.S.	531
Lavrenkov Ju.N.	315	Shtogrina O.S.	99
Lazurkin D.A.	221	Shunkevich D.V.	297
Lobzin A.P.	327	Shustova D.V.	335
Maiboroda Y.I.	469	Silivonets M.V.	531
Maklaev V.A.	427	Smetanin.M.Yu.	479
Malkovsky M.G.	77	Smetanin Yu.M.	289, 479
Matskevich A.G.	373	Soloviev S.Y.	77
Melnikov N.E.	443	Somin N.V.	373
Mikhailov J.F.	507	Sosnin P.I.	427
Mochalkina K.S.	501	Suleymanov D.Sh.	391
Molchanov Y.N.	103	Svyatkina M.N.	267
Morosanova N.A.	189	Tarassov V.B.	257, 267
Munoz M.	135	Tatur M.	135
Naidenova K.A.	327	Ternovoy M.Y.	99
Nevzorova O.A.	245	Timchenko V.A.	63
Niaborski S.N.	453	Tkachenja A.V.	355
Nikitin N.O.	91	Varlamov O.O.	279
Nikolayev V.G.	373	Volegov I.S.	229
Nikonov V.N.	531	Vovk A.I.	489
Novogrudska R.L.	447, 501	Yeliseyeva O.E.	401, 413, 513
Olizarovich E.V.	321	Yermolenko T.V.	385
Orlova Y.A.	347, 351	Zaboleeva-Zotova A.V.	347, 351
Parkalov A.V.	527	Zagorulko G.B.	181
Philipoff P.	135	Zagorulko Yu.A.	181
Pivovarchyk O.V.	221	Zalivako S.S.	297
Plesniewicz G.S.	163	Zamyatina E.B.	229
Rodchenko V.G.	321	Zhiltsov N.G.	245
Rogushina J.V.	239, 493	Zhilyakova L. Yu.	71
Rozaliev V.L.	347, 351	Zhitko V.A.	401
Ruban Y.Y.	489	Zhukevich A.I.	321
Samodumkin S.A.	512		
Shalfeeva E.A.	437, 473		
Sharipbaev A.A.	397		
Shilin D.L.	537		

АВТОРСКИЙ УКАЗАТЕЛЬ

Абрамов Н.В.	129	Дюндюков В.С.	257
Алексеев Н.А.	103	Елисеева О.Е.	401, 413, 513
Антипова О.В.	315	Ергеш Б.Ж.	397
Байрак С.А.	135	Еремеев А.А.	169
Баранович А.Е.	91	Еремеев А.П.	169
Бахтизин В.В.	453	Ермоленко Т.В.	385
Бекманова Г.Т.	397	Ерченко Е.А.	513
Биряльцев Е.В.	245	Ефименко И.В.	381
Бобков А.С.	347, 351	Жильцов Н.Г.	245
Бодякин В.И.	83	Жилякова Л.Ю.	71
Боргест Н.М.	335	Житко В.А.	401
Боровикова О.И.	181	Жукевич А.И.	321
Бурибаева А.К.	397	Заболеева-Зотова А.В.	347, 351
Варламов О.О.	279	Загорулько Г.Б.	181
Вовк А.И.	489	Загорулько Ю.А.	181
Волегов И.С.	229	Заливако С.С.	297
Гаврилова Т.А.	443	Замятина Е.Б.	229
Галимов М.Р.	245	Иванюк А.А.	123, 129
Гатиатуллин А.Р.	391	Ивашенко В.П.	193
Гецевич С.А.	401	Кадников Д.С.	315
Гецевич Ю.С.	401	Калуцкая А.П.	267
Гирнык Д.А.	489	Карабалаева М.Х.	397
Гладун А.Я.	493	Катаев В.А.	207
Глоба Л.С.	99, 103, 447, 501	Ким Ю.А.	413
Голенков В.В.	23	Киселёв В.В.	355
Головня А.И.	419	Киселёв Г.Д.	359
Горкина А.А.	175	Киселёва А.Г.	359
Грибова В.В.	213	Клещев А.С.	213, 437, 473
Гулякина Н.А.	23, 221	Кобринский Б.А.	53
Давыденко И.Т.	457	Козеренко Е.Б.	365, 373
Давыдов А.Г.	355	Козлов О.А.	507
Диковицкий В.В.	205	Колб Д.Г.	111

Комарцова Л.Г.	315	Родченко В.Г.	321
Корончик Д.Н.	339	Розалиев В.Л.	347, 351
Кузин И.А.	501	Рубан Ю.Я.	489
Кузнецов А.П.	531	Самодумкин С.А.	521
Кузнецов И.П.	373	Святкина М.Н.	267
Кузнецов О.П.	159	Силивонец М.В.	531
Кузьмин А.А.	401	Сметанин М.Ю.	479
Курбатов С.С.	327	Сметанин Ю.М.	289, 479
Лавренков Ю.Н.	315	Соловьев С.Ю.	77
Лазуркин Д.А.	221	Сомин Н.В.	373
Лобзин А.П.	327	Соснин П.И.	427
Майборода Ю.И.	469	Сулейманов Д.Ш.	391
Маклаев В.А.	427	Тарасов В.Б.	257, 267
Мальковский М.Г.	77	Татур М.М.	135
Марио Мунос	135	Терновой М.Ю.	99
Мацкевич А.Г.	373	Тимченко В.А.	63
Мельников Н.Е.	443	Ткаченя А.В.	355
Михайлов Ю.Ф.	507	Филипов Ф.	135
Молчанов Ю.Н.	103	Харламов А.А.	385
Моросанова Н.А.	189	Хахалин Г.К.	327
Мочалкина К.С.	501	Хорошевский В.Ф.	143
Найденова К.А.	327	Чернов Р.В.	335
Неборский С.Н.	453	Черняховская М.Ю.	473
Невзорова О.А.	245	Шалфеева Е.А.	437, 473
Никитин Н.О.	91	Шарипбаев А.А.	397
Николаев В.Г.	373	Шарнин М.М.	373
Никонов В.Н.	531	Шилин Д.Л.	537
Новогрудская Р.Л.	447, 501	Шмарловский А.С.	531
Одинец Д.Н.	135	Штогрин Е.С.	99
Олизарович Е.В.	321	Шункевич Д.В.	297
Орлова Ю.А.	347, 351	Шустова Д.В.	335
Паркалов А.В.	527		
Пивоварчик О.В.	221		
Плесневич Г.С.	163		
Рогушина Ю.В.	239, 493		

Научное издание

**ОТКРЫТЫЕ СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ
ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ**

OSTIS-2012

OPEN SEMANTIC TECHNOLOGIES FOR INTELLIGENT SYSTEMS

Материалы

II Международной научно-технической конференции

(Минск, 16 – 18 февраля 2012 года)

В авторской редакции

Ответственный за выпуск *В. В. Голенков*
Компьютерная верстка *С. А. Самодумкин*

Подписано в печать **01.02.2012**. Формат 60×84 1/8. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. . Уч.-изд. л. . Тираж 120 экз. Заказ **59**.

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



OSTIS

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-2013

III Международная научно-техническая конференция «Открытые семантические технологии проектирования интеллектуальных систем»

Open Semantic Technologies for Intelligent Systems

21 – 23 февраля 2013 г. Минск. Республика Беларусь

ПЕРВОЕ ИНФОРМАЦИОННОЕ СООБЩЕНИЕ

Приглашаем принять участие в III Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2013).

Конференция пройдет в период с *21 по 23 февраля 2013* 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)

ПРОГРАММНЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

Кузнецов О.П., д.т.н., проф., 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
Грибова В.В., д.т.н., РФ
Гулякина Н.А., к.ф.-м.н., доц., РБ
Еремеев А.П., д.т.н., проф., РФ
Ермолов П.П., к.т.н., Украина
Ефименко И.В., к. фил. н., РФ
Заболеева-Зотова А.В., д.т.н., РФ
Загорулько Ю.А., к.т.н., доц., РФ
Зубов А. В., д. фил. н., проф., РБ
Иванюк А.А., д.т.н., доц., РБ
Ижуткин В.С., д.ф.-м.н., проф., РФ
Клещев 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
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
Роберт И.В., д.п.н., проф., РФ
Родченко В.Г., к.т.н., доц., РБ
Сморозин В.С., д.т.н., РБ
Соснин П.И., д.т.н., проф., РФ
Стефанюк В.Л. д.т.н., проф., РФ
Сулейманов Д.Ш., академик АН Татарстана, РФ
Тарасов В.Б., к.т.н., доц., РФ
Харламов А.А., д.т.н., РФ
Хейдоров И.Э., к.ф.-м.н., доц., РБ
Хорошевский В.Ф., д.т.н., проф., РФ
Чернявский А.Ф., академик НАН Беларуси
Щербак С.С., к.т.н., доц., Украина

ДОКЛАДЫ НА КОНФЕРЕНЦИЮ БУДУТ ПРИНИМАТЬСЯ ПО СЛЕДУЮЩИМ НАПРАВЛЕНИЯМ:

1. *Базовые универсальные семантические модели представления и обработки знаний и их программная и аппаратная реализация.*
2. *Семантические технологии компонентного проектирования совместимых баз знаний, программ и пакетов программ, ориентированных на обработку знаний.*
3. *Семантические модели информационного поиска и решения задач. Семантические технологии компонентного проектирования совместимых интеллектуальных информационно-поисковых машин и решателей задач.*
4. *Семантические технологии компонентного проектирования совместимых мультимодальных и естественно-языковых интерфейсов интеллектуальных систем.*
5. *Методологии и менеджмент компонентного проектирования семантически совместимых интеллектуальных систем в рамках Open Source проектов.*
6. *Семантически совместимые прикладные интеллектуальные системы.*

ЦЕЛЬ И ФОРМАТ ПРОВЕДЕНИЯ КОНФЕРЕНЦИИ

Целью конференции является обсуждение проблем создания комплексной технологии проектирования семантически совместимых интеллектуальных систем. Этим определяется и формат её проведения, предполагающий

- (1) последовательное проведение секций,
- (2) ориентацию на круглые столы, посвященные обсуждению различных вопросов,
- (3) 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 МАТЕРИАЛОВ ДОКЛАДОВ

Материалы докладов (только по перечисленным выше направлениям) представляются в готовом для публикации виде. Текст доклада должен быть логически законченным и содержать новые научные и практические результаты.

Крайний срок получения материалов Оргкомитетом – **1 декабря 2012**.

Объем материалов доклада, включая краткий перевод, иллюстрации и список литературы – 6 или 8 полных страниц по правилам оформления.

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>.

Правила подготовки материалов докладов размещены на сайте конференции.

ФОРМИРОВАНИЕ ПРОГРАММЫ КОНФЕРЕНЦИИ

Программа конференции формируется Программным комитетом по результатам рецензирования, представленных материалов докладов и будет опубликована **15 января 2013** на сайте конференции <http://conf.ostis.net>.

ПУБЛИКАЦИЯ МАТЕРИАЛОВ КОНФЕРЕНЦИИ

Материалы докладов, включенных в программу конференции, печатаются в сборнике материалов конференции и публикуются на сайте конференции за неделю до начала конференции.

СВЯЗЬ С ОРГАНИЗАТОРАМИ КОНФЕРЕНЦИИ

Сайт: <http://conf.ostis.net>

Email: ostisconf@gmail.com