Информатика в техническом университете

А.И. Башмаков, И.А. Башмаков

Интеллектуальные информационные технологии

Издательство МГТУ имени Н.Э. Баумана

Серия основана в 2000 году

А.И. Башмаков, И.А. Башмаков

Интеллектуальные информационные технологии

Допущено Министерством образования
и науки Российской Федерации
в качестве учебного пособия для студентов
высших учебных заведений,
обучающихся по направлению
подготовки дипломированных специалистов
«Информатика и вычислительная техника»

Москва
Издательство МГТУ имени Н.Э. Баумана
2005

УДК 004.8:681.3.06(075.8)
ББК 32.813+32.973.26-018.2я73
Б336

Рецензенты:
д-р техн. наук, профессор И.П. Норенков
(Московский государственный технический университет им. Н.Э. Баумана);
кафедра «Компьютерные технологии и системы»
Московского государственного университета прикладной биотехнологии
(зав. кафедрой профессор Ю.А. Ивашкин);
кафедра «Вычислительные машины, системы и сети»
Московского энергетического института (технического университета)
(зав. кафедрой профессор И.И. Ладыгин)

Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии: Учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 304 с.: ил. — (Информатика в техническом университете).

Б336

ISBN 5-7038-2544-X

Интеллектуальные информационные технологии — одна из наиболее перспективных и быстро развивающихся научных и прикладных областей информатики. В учебном пособии рассматриваются ее основные направления: обработка текстов на естественном языке, моделирование знаний и базы знаний, управление знаниями, распознавание образов, нейротехнологии, интеллектуализация Internet, концептуальное программирование и др. Основное внимание уделяется математическим моделям, методам и инструментальным средствам разработки программного обеспечения интеллектуальных автоматизированных систем.

Содержание учебного пособия основано на материалах, используемых авторами в учебном процессе в МГТУ им. Н.Э. Баумана и МЭИ (ТУ).

Для студентов высших технических учебных заведений, изучающих информационные технологии и методы их интеллектуализации. Может быть полезно аспирантам и специалистам, занимающимся данной проблематикой.

УДК 004.8:681.3.06(075.8)
ББК 32.813+32.973.26-018.2я73

ISBN 5-7038-2544-X

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ 8

СПИСОК ОСНОВНЫХ СОКРАЩЕНИЙ 11

СТРУКТУРА ИССЛЕДОВАНИЙ В ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА 15

ВВЕДЕНИЕ 17

1. ТЕХНОЛОГИЯ КОНЦЕПТУАЛЬНОГО ПРОГРАММИРОВАНИЯ 21
1.1. Основы теории концептуального программирования 21
1.2. Инструментарий концептуального программирования 28
Вопросы для самопроверки 29

2. ТЕХНОЛОГИИ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ОБРАЗОВ 32
2.1. Основные понятия теории автоматического распознавания образов 32
2.2. Примеры программной реализации OCR-систем 35
Вопросы для самопроверки 40

3. АВТОМАТИЗАЦИЯ РАБОТЫ СО ЗНАНИЯМИ, ПРЕДСТАВЛЕННЫМИ В ТЕКСТОВОМ ВИДЕ 41
3.1. Основы гипертекстовой информационной технологии 41
3.1.1. Основные понятия гипертекстовой информационной технологии 42
3.1.2. Формализованная модель гипертекста 44
3.1.3. Условно-типовая модель гипертекста 47
3.1.4. Инструментальные средства для создания гипертекста 49
3.1.5. Гипертекстовые информационно-поисковые системы 55
3.1.6. Методы извлечения знаний для построения гипертекста 61
3.1.7. Автоматизация построения гипертекста 62
3.1.8. Место гипертекстовой информационной технологии среди технологий искусственного интеллекта 63
Вопросы для самопроверки 66
3.2. Автоматизированное извлечение знаний из текста 67
3.2.1. Проблема понимания текста на естественном языке 67

3.2.2. Компьютерные методы поиска в тексте ... 69
Вопросы для самопроверки ... 76
3.3. Автоматическое реферирование и аннотирование ... 77
Вопросы для самопроверки ... 89
3.4. Машинный перевод ... 90
Вопросы для самопроверки ... 96
3.5. Автоматическая классификация документов ... 96
Вопросы для самопроверки ... 99
3.6. Комплексные интеллектуальные программные системы для обработки текстов ... 100
3.6.1. Комплексный смысловой анализатор текста Text Analyst ... 100
3.6.2. Промышленная информационно-поисковая система Excalibur RetrievalWare ... 107
3.6.3. Пакет NeurOK Semantic Suite ... 113
Вопросы для самопроверки ... 118

4. МЕТАДАННЫЕ ДЛЯ ИНФОРМАЦИОННЫХ РЕСУРСОВ ... 120
4.1. Системы и модели метаданных ... 120
4.2. Семантический web и платформа XML ... 128
Вопросы для самопроверки ... 133

5. МОДЕЛИРОВАНИЕ ЗНАНИЙ О ПРЕДМЕТНЫХ ОБЛАСТЯХ КАК ОСНОВА ИНТЕЛЛЕКТУАЛЬНЫХ АВТОМАТИЗИРОВАННЫХ СИСТЕМ ... 135
5.1. Категория знания ... 135
Вопросы для самопроверки ... 146
5.2. Модели знаний ... 147
Вопросы для самопроверки ... 156
5.3. Сетевые модели знаний ... 157
5.3.1. Модель *M*1 — расширенные семантические сети ... 157
5.3.2. Модель *M*2 — неоднородные семантические сети ... 160
5.3.3. Модель *M*3 — нечеткие семантические сети ... 162
5.3.4. Модель *M*4 — обобщенная модель представления знаний о предметной области ... 164
Вопросы для самопроверки ... 172
5.4. Онтологический подход и его использование ... 173
5.4.1. Понятие онтологии ... 173
5.4.2. Основные задачи, решаемые с помощью онтологий ... 175
5.4.3. Модель онтологии ... 181
5.4.4. Методики построения онтологий и требования к средствам их спецификации ... 184
5.4.5. Обзор наиболее известных онтологических проектов ... 191
5.4.6. Примеры использования онтологий ... 194
Вопросы для самопроверки ... 197
5.5. Основы технологии баз знаний ... 198
5.5.1. Общие положения ... 198

5.5.2. Система операций для работы со знаниями в базе знаний 200
5.5.3. Элементарные операции 200
5.5.4. Комплексные операции 206
Вопросы для самопроверки 230

6. НЕЙРОННЫЕ СЕМИОТИЧЕСКИЕ СИСТЕМЫ 231
6.1. Общая характеристика направления 231
6.2. Нейропакеты 237
6.3. Модели сенсорных и языковой систем человека 246
Вопросы для самопроверки 251

7. СИСТЕМЫ УПРАВЛЕНИЯ ЗНАНИЯМИ 253
7.1. Общая характеристика направления 253
Вопросы для самопроверки 256
7.2. Технологии хранилищ данных и интеллектуального анализа данных 256
7.2.1. Основные понятия 256
7.2.2. Технология OLAP и многомерные модели данных 258
7.2.3. Глубинный анализ данных 262
Вопросы для самопроверки 265
7.3. Системы поддержки инновационной деятельности 266
Вопросы для самопроверки 279

ЗАКЛЮЧЕНИЕ 280

СПИСОК ОСНОВНОЙ ЛИТЕРАТУРЫ 282

СПИСОК ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ 283

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ 298

ПРЕДИСЛОВИЕ

Эволюция информационных технологий и систем все в большей степени определяется их интеллектуализацией. ***Интеллектуальные информационные технологии*** — одна из наиболее перспективных и быстро развивающихся научных и прикладных областей информатики. Она оказывает существенное влияние на все научные и технологические направления, связанные с использованием компьютеров, и уже сегодня дает обществу то, что оно ждет от науки, — практически значимые результаты, многие из которых способствуют кардинальным изменениям в сферах их применения.

Целями интеллектуальных информационных технологий являются, во-первых, расширение круга задач, решаемых с помощью компьютеров, особенно в слабоструктурированных предметных областях, и во-вторых, повышение уровня интеллектуальной информационной поддержки современного специалиста.

Ключевым компонентом научного фундамента интеллектуальных информационных технологий является ***искусственный интеллект (ИИ).*** Для создания и развития ИИ как научного направления за рубежом много сделали Н. Винер, У. Маккаллох, У. Питтс, Д. Маккарти (который впервые ввел термин «artificial intelligence»), Ф. Розенблат, А. Сазерленд, М. Минский, С. Пейперт, А. Ньюэлл, Г. Саймон, Дж. Шоу, Э. Фейгенбаум, А. Кольмероэ, Н. Хомский, Т. Виноград, М. Куиллиан, Р. Шенк, Н. Нильсон, П. Уинстон, Л. Заде, Р. Редди, Д. Ленат, Дж. Хинтон, Дж. Андерсон, Ж.-Л. Лорьер и многие другие. В СССР, а затем в России со становлением и развитием ИИ связывают имена А.А. Ляпунова, А.И. Берга, Г.С. Поспелова, М.Л. Цетлина, М.М. Бонгарда, М.А. Гаврилова, А.П. Ершова, В.Н. Пушкина, Л.Т. Кузина, А.С. Нариньяни, А.И. Половинкина, В.В. Чавчанидзе, В.К. Финна, Э.В. Попова, Э.Х. Тыугу, Н.Н. Непейводы, И.П. Кузнецова, О.И. Ларичева, А.И. Галушкина, А.Н. Горбаня, А.В. Чечкина и многих других. Следует отметить исключительную роль Д.А. Поспелова и его научной школы: В.Н. Вагина, Т.А. Гавриловой, А.П. Еремеева, Г.С. Осипова, В.Ф. Хорошевского и др.

Настоящее учебное пособие предназначено для студентов вузов, изучающих информационные технологии и методы их интеллектуализации, а также аспирантов и специалистов, занимающихся данной проблематикой. Оно основано на материалах, используемых авторами в учебном процессе в

МГТУ им. Н.Э. Баумана и МЭИ (ТУ). Его содержание в значительной мере охватывает вопросы, связанные с интеллектуализацией информационных технологий и систем, входящие в учебные программы дисциплин «Системы искусственного интеллекта», «Информационные технологии», «Интеллектуальные подсистемы САПР», «Представление знаний в информационных системах», «Технология разработки программного обеспечения интеллектуальных автоматизированных систем», «Интеллектуальные системы», «Основы искусственного интеллекта», «Интеллектуальные информационные системы» и других дисциплин в рамках направлений подготовки «Информатика и вычислительная техника» (ОКСО 230100), «Информационные системы» (ОКСО 230200), «Информационные технологии» (ОКСО 010400), «Прикладная математика и информатика» (ОКСО 010500), «Прикладная математика» (ОКСО 230400).

Основными задачами учебного пособия являются:

1) формирование представлений о классах и структуре программного обеспечения (ПО) интеллектуальных автоматизированных систем (ИАС), в особенности об инвариантном к предметной области ядре ПО ИАС;

2) создание представлений о методах, математическом аппарате и инструментальных средствах разработки ПО ИАС во взаимосвязи с обеспечивающими подсистемами ИАС: комплексом технических средств, а также математическим, лингвистическим и информационным обеспечениями;

3) приобретение знаний и умений, связанных с технологическим подходом к разработке ПО ИАС.

Использование технологического подхода к разработке ПО обеспечивает:

- концептуальное единство всех частей программного проекта;
- интеграцию и координацию деятельности отдельных исполнителей, в том числе программистов, в рамках единого проекта;
- совмещение разработки программной документации с ходом реализации проекта;
- повышение производительности труда программистов;
- повышение надежности и качества программного продукта;
- снижение стоимости разработки программного продукта;
- повышение границы сложности программных проектов.

Учебное пособие состоит из введения, семи глав и заключения.

Во введении представлены задачи учебного пособия, укрупненная функциональная модель интеллектуальной системы и структура исследований в области ИИ. Главы и параграфы пособия соотносятся с направлениями данной структуры.

В первой главе изложены теоретические основы технологии концептуального программирования и дана характеристика ее реализации в серии программных решателей пакета решения инженерных задач (ПРИЗ).

Во второй главе отражены основные понятия технологии автоматического распознавания образов. Вопросы ее реализации рассмотрены на примерах ведущих российских систем оптического чтения текстов.

Третья глава посвящена автоматизации работы со знаниями, представленными в текстовом виде. В ней описаны гипертекстовые модели и системы, методы извлечения знаний из текста и компьютерного поиска в тексте, технологии автоматического реферирования и аннотирования, машинного перевода и автоматической классификации документов. Завершает главу характеристика комплексных интеллектуальных программных систем для обработки текстов: комплексного смыслового анализатора текста Text Analyst, промышленной информационно-поисковой системы Excalibur RetrievalWare, пакета NeurOK Semantic Suite.

В четвертой главе отражено современное состояние работ в области метаданных для информационных ресурсов. Главное внимание уделено роли метаданных в обеспечении интеллектуализации WWW. Охарактеризованы универсальная система метаданных «Дублинское ядро» и модель RDF. Рассмотрены направления интеллектуализации Internet (концепция семантического web). Приведен перечень стандартов и спецификаций, составляющих ядро платформы XML, служащей технологической основой семантического web.

Пятая глава пособия является главной. Она посвящена вопросам моделирования знаний о предметных областях и роли этих моделей и методов в ИАС. Изложены современные представления о категории знаний. Приведен обзор базовых моделей знаний. Рассмотрены четыре модели семантических сетей. Дана развернутая характеристика онтологического подхода. Описаны концептуальные основы технологии баз знаний.

Шестая глава представляет технологии нейронных семиотических систем. Рассмотрены основные понятия нейротехнологий, структура работ в области нейрокибернетики, классификация, характеристики и примеры нейропакетов, а также подход к моделированию сенсорных и языковой систем человека искусственными нейронными сетями.

Седьмая глава посвящена системам управления знаниями. В ней дана характеристика технологий хранилищ данных и интеллектуального анализа данных, а также систем поддержки инновационной деятельности в технических областях.

СПИСОК ОСНОВНЫХ СОКРАЩЕНИЙ

АСНИ — автоматизированная система научных исследований
АСУ — автоматизированная система управления
АЯ — алгоритмическое ядро
БД — база данных
БЗ — база знаний
БНФ — нормальная форма Бэкуса—Наура
БСЭ — Большая советская энциклопедия
ВМ — вычислительная модель
ВНС — высшая нервная система (человека)
ГИПС — гипертекстовая информационно-поисковая система
ГИТ — гипертекстовая информационная технология
ГРНТИ — государственный рубрикатор научно-технической информации
ГТ — гипертекст
ЕЯ — естественный язык
ИАД — интеллектуальный анализ данных
ИАС — интеллектуальная автоматизированная система
ИИ — искусственный интеллект
ИНС — искусственная нейронная сеть
ИО — информационное обеспечение
ИПС — информационно-поисковая система
ИР — информационный ресурс
ИС — информационная система
ИСС — информационно-справочная статья
КРН — квазирецепторный нейрон
КТС — комплекс технических средств
ЛО — лингвистическое обеспечение
ЛП — лингвистический процессор
МВ — машина вывода
МО — математическое обеспечение
МП — машинный перевод
МПрО — модель предметной области
НИТ — новая информационная технология
НК — нейрокомпьютер
НОСС — нечеткая объектно-ориентированная семантическая сеть
НП — нейропакет
НСС — неоднородная семантическая сеть
ОЕЯ — ограниченный естественный язык
ОСС — объектно-ориентированная семантическая сеть

ПО — программное обеспечение
ПРИЗ — пакет решения инженерных задач (программный инструментарий, реализующий ТКП)
ПрО — предметная область
РСС — расширенная семантическая сеть
САПР — система автоматизированного проектирования
СН — символьный нейрон
СП — сетевая продукция
СУБД — система управления базами данных
СУБЗ — система управления базами знаний
СУЗ — система управления знаниями
ТКП — технология концептуального программирования
ТРИЗ — теория решения изобретательских задач
УДК — универсальная десятичная классификация
УТОПИСТ — универсальный транслятор описаний теорий (язык, используемый в ПРИЗ)
ФС — формальная система
ЭС — экспертная система
ЭСМ — элементарная сенсорная модель
ЭСС — элементарная сенсорная система
ЭФ — элементарный фрагмент
ЭЯС — элементарная языковая система
API — Application Programming Interface — интерфейс прикладного программирования
APRP — Adaptive Pattern Recognition Processing — адаптивное распознавание образов (технология, разработанная Convera Technologies Corp.)
CALS — Computer-Aided Acquisition and Lifecycle Support — компьютерная поддержка жизненного цикла (совокупность стандартов, унифицирующих спецификации технической системы на всех этапах ее жизненного цикла)
CASE — Computer Aided Software Engineering — автоматизированная разработка программного обеспечения
COM — Component Object Model — модель составных объектов (стандарт Microsoft, описывающий правила создания и взаимодействия программных объектов в среде Windows)
CRISP-DM — Cross Industry Standard Process for Data Mining — проект, направленный на унификацию и стандартизацию технологий DM
DAML — DARPA Agent Markup Language — язык разметки агентов, разработанный DARPA
DARPA — Defense Advanced Research Projects Agency — Агентство перспективных исследований Министерства обороны США
DM — Data Mining — глубинный анализ данных
DS — Description Subsumption — диаграмма строгой классификации (используется в IDEF5)
DTD — Document Type Definition — определение типа документа (язык описания модели XML-документа)

EL — Elaboration Language — язык доработок и уточнений (используется в IDEF5)
ERW — Excalibur RetrievalWare
FSNL — Fuzzy Semantic Network Language — язык описания нечеткой семантической сети
FTP — File Transfer Protocol — протокол передачи файлов
HOLAP — Hybrid OLAP — гибридная OLAP (способ хранения данных в OLAP)
HTML — HyperText Markup Language — язык гипертекстовой разметки
HTTP — HyperText Transport Protocol — протокол передачи гипертекста
KIF — Knowledge Interchange Format — формат обмена знаниями (один из языков представления знаний)
LOM — Learning Object Metadata — концептуальная схема метаданных для образовательных объектов (информационных ресурсов для сферы образования)
MDA — Model.Driven Architecture — архитектура, управляемая моделью (основана на объектно-ориентированной модели знаний)
MIME — Multipurpose Internet Mail Extensions — многоцелевые расширения почтовой службы Internet
MOLAP — Multidimensional OLAP — многомерная OLAP (способ хранения данных в OLAP)
NKC — Natural Kind Classification — диаграмма естественной (видовой) классификации (используется в IDEF5)
NLP — Natural Language Processing — обработка текстов на ЕЯ
OCR — Optical Character Recognition — оптическое распознавание символов
ODBC — Open DataBase Connectivity interface — открытый интерфейс взаимодействия с БД
ODP — Open Distributed Processing — открытая распределенная обработка (основана на объектно-ориентированной модели знаний)
OIL — Ontology Interchange Language — язык обмена онтологиями (один из языков описаний онтологий)
OLAP — On-Line Analytical Processing — интерактивная аналитическая обработка данных
OLE — Object Linking and Embedding — связывание и встраивание объектов (технология, обеспечивающая возможность включения в состав документа информационных объектов, имеющих разные форматы и обрабатываемых разными приложениями)
OLTP — On-Line Transaction Processing — оперативная обработка транзакций
OMG — Object Management Group — Консорциум OMG
QBE — Query-By-Example — запрос по образцу, язык запросов по образцу
RDF — Resource Description Framework — модель представления метаданных, описывающих ИР, и соответствующий ей язык, являющийся приложением XML
ROLAP — Relational OLAP — реляционная OLAP (способ хранения данных в OLAP)
SAO — (Subject — Action — Object) — (субъект — действие — объект)

SDK — Software Development Kit — инструментарий разработки ПО
SGML — Standard Generalized Markup Language — стандартный обобщенный язык разметки
SL — Schematic Language — схематический язык (используется в IDEF5)
SOAP — Simple Object Application Protocol — прикладной протокол передачи простых объектов (протокол передачи XML-данных)
SQL — Structured Query Language — язык структурированных запросов
STEP — STandard for Exchange of Product data — стандарт обмена спецификациями промышленных изделий (группа стандартов ISO 10303, лежащих в основе CALS-технологий)
SSR — Structural Synthesis Rules — система правил структурного синтеза программ
URI — Uniform Resource Identifier — унифицированный идентификатор ресурса
URL — Uniform Resource Locator — унифицированный указатель ресурса
W3C — World Wide Web Consortium — Консорциум WWW
WWW — World Wide Web — «всемирная паутина» (глобальная гипертекстовая система, использующая Internet в качестве транспортного средства)
XML — eXtensible Markup Language — расширяемый язык разметки
XMLP — XML Protocol — протокол передачи XML-данных

СТРУКТУРА ИССЛЕДОВАНИЙ В ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

1. Программы решения отдельных интеллектуальных задач
    1.1. Программы компьютерного доказательства теорем
    1.2. Игровые программы
    1.3. Распознающие и узнающие программы
    1.4. Программы для семантического анализа и обработки естественно-языковой информации
        1.4.1. Машинный поиск в базах данных естественно-языковых документов
        1.4.2. Машинный перевод
        1.4.3. Автоматическое реферирование
        1.4.4. Автоматическая классификация документов
        1.4.5. Генерация (синтез) текста
        1.4.6. Генерация (синтез) речи
    1.5. Программы, моделирующие поведение
    1.6. Программы для анализа и синтеза музыкальных произведений
2. Работа со знаниями
    2.1. Методы и средства представления знаний
        2.1.1. Модели знаний
            2.1.1.1. Логические модели
            2.1.1.2. Продукционные модели
            2.1.1.3. Фреймы
            2.1.1.4. Семантические сети
            2.1.1.5. Онтологии
            2.1.1.6. Объектно-ориентированные модели
        2.1.2. Системы представления знаний и базы знаний
    2.2. Методы и средства извлечения знаний из различных источников
        2.2.1. Приобретение знаний от экспертов
        2.2.2. Извлечение знаний из документов
        2.2.3. Согласование и интеграция знаний
    2.3. Методы обработки знаний
        2.3.1. Поиск знаний
        2.3.2. Верификация знаний
        2.3.3. Систематизация знаний
        2.3.4. Вывод на знаниях
        2.3.5. Обработка нечетких знаний
        2.3.6. Аргументация и объяснение на основе знаний

3. Интеллектуальное программирование
    3.1. Языки для интеллектуальных систем
        3.1.1. Языки логического программирования
        3.1.2. Объектно-ориентированные языки
        3.1.3. Языки представления знаний
        3.1.4. Языки семантической разметки
    3.2. Автоматический синтез программ
        3.2.1. Дедуктивные методы
        3.2.2. Индуктивные методы
    3.3. Инструментальные средства
    3.4. Интеллектуальные интерфейсы
    3.5. Мультиагентные технологии
4. Интеллектуальные автоматизированные системы
    4.1. Нейропакеты
    4.2. Интеллектуальные информационные системы
    4.3. Экспертные системы
    4.4. Интеллектуальные АСУ
    4.5. Интеллектуальные САПР
    4.6. Интеллектуальные АСНИ
    4.7. Интеллектуальные компьютерные средства обучения
    4.8. Интеллектуальные роботы
    4.9. Интеллектуальные консультирующие системы
    4.10. Системы управления знаниями
    4.11. Системы виртуальной реальности

Когда-то наш разум был вне конкуренции, но, возможно, придет день, когда вычислительные машины будут смеяться над нами и задавать вопрос о том, могут ли биологические информационные процессоры быть достаточно разумными.

П. Уинстон

ВВЕДЕНИЕ

В наше время преимущества в конкурентной борьбе уже не определяются ни размерами страны, ни ее природными ресурсами. Теперь все решают уровень образования и объем знаний, накопленных обществом. В будущем процветать будут государства, которые сумеют превзойти другие в создании и освоении новых знаний. Особую роль в этом играют *новые информационные технологии* (*НИТ*), а в них – методы и средства искусственного интеллекта (ИИ). Чтобы получить представления об основных технологиях ИИ, необходимо изучить, как его важнейшие концепции воплощаются в программных решениях.

Программы позволяют строить ясные описания разнообразных процессов. Их структуры могут отражать структуры тех задач, для решения которых они предназначены. Для изучающих ИИ программирование служит таким же средством, каким является математика для изучающих более старые области науки.

Под ***интеллектуальными системами*** понимают любые биологические, искусственные или формальные системы, проявляющие способность к целенаправленному поведению. Последнее включает свойства (проявления) общения, накопления знаний, принятия решений, обучения, адаптации и т. д.

В настоящее время существует устойчивая тенденция *интеллектуализации* компьютеров и их программного обеспечения (ПО). Основные функции будущих компьютеров — решение задач все в большей степени невычислительного характера, в том числе логический вывод, управление базами знаний (БЗ), обеспечение интеллектуальных интерфейсов и др. Интеллектуализация компьютеров осуществляется за счет разработки как специальной аппаратуры (например, нейрокомпьютеры), так и ПО (экспертные системы, базы знаний, решатели задач и т. д.).

Рабочее определение понятия «интеллектуальная система» предложено в [14]. Система считается интеллектуальной, если в ней реализованы следующие три базовые функции.

1. *Функция представления и обработки знаний.* Интеллектуальная система должна быть способна накапливать знания об окружающем мире, классифицировать и оценивать их с точки зрения прагматики и непротиворечивости, инициировать процессы получения новых знаний, соотносить новые знания со знаниями, хранящимися в базе знаний.

2. *Функция рассуждения.* Интеллектуальная система должна быть способна формировать новые знания с помощью логического вывода и механизмов выявления закономерностей в накопленных знаниях, получать обобщенные знания на основе частных знаний и логически планировать свою деятельность.

3. *Функция общения.* Интеллектуальная система должна быть способна общаться с человеком на языке, близком к естественному языку (ЕЯ) и получать информацию через каналы, аналогичные тем, которые использует человек при восприятии окружающего мира (прежде всего, зрительный и звуковой), уметь формировать «для себя» или по просьбе человека объяснения собственной деятельности (т. е. отвечать на вопросы типа «Как я это сделал?»), оказывать человеку помощь за счет знаний, которые хранятся в ее памяти, и логических средств рассуждения.

Функциональная модель интеллектуальной системы представлена на рис. В.1 [13].

В рамках этой функциональной модели:

• *интеллектуальный интерфейс* обеспечивает общение с внешней средой и преобразование информации из внешнего во внутреннее представление и обратно;

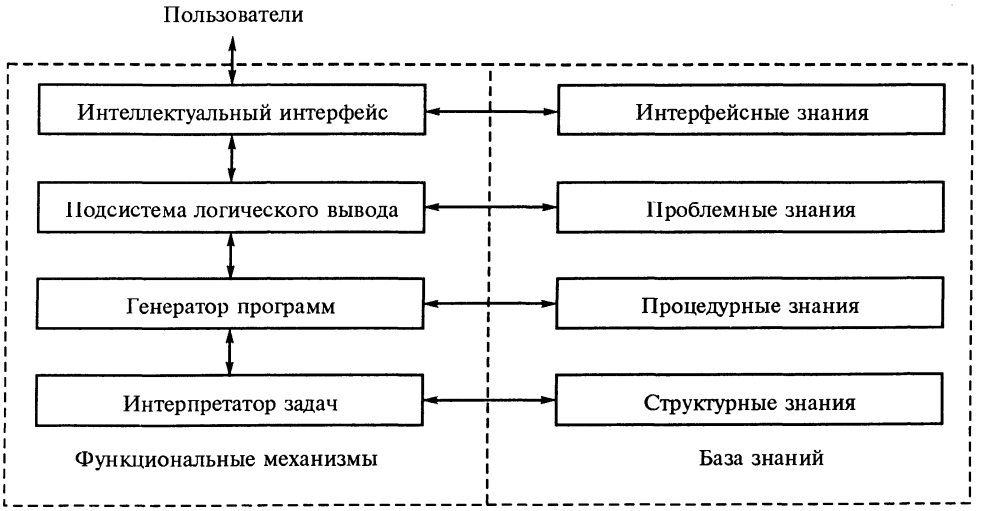


Рис. В.1. Функциональная модель интеллектуальной системы

• *подсистема логического вывода* на основе анализа семантики входных сообщений и имеющихся знаний формулирует постановку задачи, осуществляет поиск вариантов ее решения и выбирает из них наилучшие;

• *генератор программ* формирует программу решения, используя знания о методах решения задач;

• *интерпретатор задач* обеспечивает выполнение сгенерированных программ;

• *база знаний* обеспечивает хранение и доступ к различным видам знаний, используемым интеллектуальной автоматизированной системой (ИАС) при ее функционировании.

Выделяют следующие *виды знаний:*

• интерфейсные — знания о взаимодействии с окружающей средой;

• проблемные — знания о предметной области (ПрО);

• процедурные — знания о методах решения задач;

• структурные — знания об операционной среде;

• метазнания — знания о свойствах знаний.

В табл. В.1 представлено развитие операционной среды, базы знаний и интерфейса для пяти поколений ЭВМ.

Таблица В.1

Основные функциональные свойства	Поколение ЭВМ				
	1	2	3	4	5
Операционная среда					
интеллектуальная машина	0	0	0	0	1
объектная машина	0	0	0	1	x
виртуальная машина	0	0	1	x	x
процедурная машина	0	1	x	x	x
реальная машина	1	x	x	x	x
База знаний					
метазнания	0	0	0	0	1
проблемные знания	0	0	0	1	1
структурные знания	0	0	1	1	1
интерфейсные знания	0	1	1	1	1
процедурные знания	1	1	1	1	1
Интерфейс					
естественные языки	0	0	0	0	1
языки спецификаций	0	0	0	1	1
языки управления	0	0	1	1	1
процедурные языки	0	1	1	1	1
машинные языки	1	1	1	1	1

Примечание. 1 — присутствует; 0 — отсутствует; x — неопределяющее свойство.

Попытки определения структуры исследований в области ИИ предпринимались неоднократно. Одна из наиболее известных точек зрения по этому вопросу изложена в [12]. Согласно ей исследования в области ИИ включают два базовых направления:

• *бионическое*, занимающееся проблемами искусственного воспроизведения структур и процессов, характерных для человеческого мозга и лежащих в основе решения задач человеком;

• *программно-прагматическое*, занимающееся созданием программ решения задач, считающихся прерогативой человеческого интеллекта (поиск, классификация, обучение, принятие решений, распознавание образов, рассуждения и др.).

В рамках первого направления в учебном пособии будут рассмотрены проблемы создания ПО, использующего модели искусственных нейронных сетей (ИНС).

В силу сложности целей и задач бионического направления до последнего времени доминирующим в ИИ являлось программно-прагматическое направление, хотя в будущем бионическое направление, вероятно, будет определяющим. Поэтому в пособии основное внимание уделяется общим для обоих направлений концепциям технологического подхода к созданию ПО, а также базовым методам программно-прагматического направления.

В программно-прагматическом направлении выделяют три подхода:

• *локальный,* или *задачный* — создание для каждой задачи, присущей интеллектуальной деятельности человека, специальной программы, дающей результат не хуже того, что достигает человек (например, программы для игры в шахматы);

• *системный,* или *основанный на знаниях* — создание средств автоматизации построения программ для решения интеллектуальных задач на основе знаний о ПрО; в настоящее время этот подход является преобладающим;

• *использующий метапроцедуры программирования* для составления интеллектуальных программ по описаниям задач на ЕЯ.

Структура исследований, относящихся к программно-прагматическому направлению ИИ, приведена на с. 15–16. Первые три области исследований соответствуют названным выше подходам. Четвертая область представляет основные классы прикладных ИАС, использующих результаты, получаемые в рамках программно-прагматического направления.

Материал учебного пособия соотносится с направлениями исследований из данной структуры. Ссылки на номера направлений указаны в преамбулах глав и сносках к названиям параграфов.

Не существует фактов,
есть лишь их интерпретация.

Ф. Ницше

1. ТЕХНОЛОГИЯ КОНЦЕПТУАЛЬНОГО ПРОГРАММИРОВАНИЯ

Технология концептуального программирования ориентирована на хорошо структурированные предметные области. Ее сущность заключается в автоматическом синтезе программ решения прикладных задач по их описанию на ограниченном естественном языке.

В главе изложены теоретические основы данной технологии. Приведена краткая характеристика ее инструментария — программных решателей пакета решения инженерных задач (ПРИЗ).

Содержание главы соответствует направлениям исследований в области ИИ 1.1 и 3.2.

1.1. Основы теории концептуального программирования

Приходится порой простые мысли
доказывать всерьез, как теоремы.

О. Сулейменов

Технология концептуального программирования (ТКП) — одна из старейших и наиболее развитых в ИИ как в теоретическом, так и в практическом аспектах. Она разработана советскими учеными и сейчас ведущие позиции в ней занимают ученые России и Эстонии. Технология концептуального программирования предназначена для *синтеза программ решения задач по их описанию на ограниченном естественном языке (ОЕЯ) при некоторых ограничениях.* Эти ограничения требуют, во-первых, точного указания ПрО, к которой относится решаемая задача, и, во-вторых, фиксации класса решаемых задач. Последние получили название *вычислительных* или *расчетно-логических задач.* В общем случае их описание на ОЕЯ имеет вид:

$$\text{Зная } M, \text{ вычислить } (y_1, \ldots, y_n) \text{ по } (x_1, \ldots, x_m). \qquad (1.1)$$

В выражении (1.1) M идентифицирует ПрО (например, тригонометрию, кинематику и т. д.). Кортеж $(x_1, \ldots, x_m)$ содержит идентификаторы пе-

ременных с известными значениями, а кортеж $(y_1, \ldots, y_n)$ — идентификаторы переменных, значения которых требуется определить.

Такая постановка допускает широкую трактовку понятия ПрО. Рассмотрим примеры интерпретации (1.1).

Пример 1. Зная *треугольник,* вычислить S по a, b, c.

Здесь ПрО — раздел геометрии, в котором определяются понятие треугольника и его свойства; S — площадь треугольника с вершинами a, b и c, координаты которых считаются известными.

Пример 2. Зная *теория,* вычислить *доказательство* по *формула.*

Здесь ПрО задана некоторой формальной системой *теория.* Требуется доказать истинность или ложность указанной формулы.

Пример 3. Зная *кадры,* вычислить *фамилии_молодых_сотрудников.*

Здесь ПрО представляет база данных (БД) с описанием кадров. Предполагается, что система располагает критерием отнесения сотрудника к категории молодых сотрудников.

Существенным ограничением ТКП является предположение, что в компьютере имеется модель ПрО (МПрО), с которой можно манипулировать. В технологии концептуального программирования для представления МПрО используются семантические сети специального вида, называемые ***вычислительными моделями (ВМ).*** Они будут описаны ниже.

Известны четыре *подхода к синтезу программ*:

1) дедуктивный — построение программы выполняется на основе доказательства, что решение задачи существует;

2) индуктивный — программа строится по примерам, каждый из которых определяет ответ для некоторого подкласса исходных данных;

3) трансформационный — программа синтезируется путем преобразования исходного описания задачи по правилам, совокупность которых представляет знания о ее решении;

4) утилитарный — программа строится из практических потребностей на основе частных закономерностей и приемов.

В технологии концептуального программирования используются первые два подхода (дедуктивный и индуктивный).

Основная идея ТКП состоит в следующем. Пусть существует постановка задачи в виде (1.1). Необходимо:

- перейти от (1.1) к теореме существования решения данной задачи;
- построить доказательство теоремы существования;
- извлечь из доказательства программу решения задачи.

При реализации этого метода получаем два важных результата:

1) программа точно соответствует описанию задачи;

2) вместо отладки программы выполняется «отладка» описания задачи.

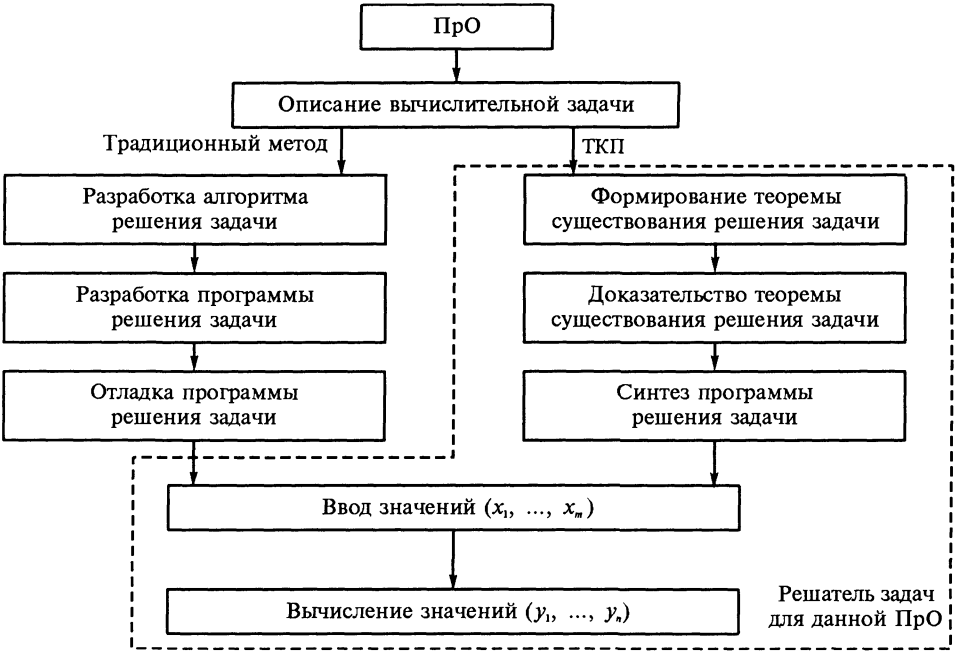


Рис. 1.1. Сравнение традиционного метода разработки программы решения задачи и ТКП

Сравнение традиционного метода разработки программы решения задачи и ТКП иллюстрирует рис. 1.1.

Процесс перехода от описания ПрО на ОЕЯ к точной спецификации этого описания на некотором формальном языке, ориентированном на компьютерное представление, называется *концептуализацией.* Отсюда и пошло название рассматриваемой технологии — ТКП.

В качестве математического аппарата концептуализации в рамках ТКП разработаны, как отмечалось выше, так называемые ВМ. Они являются разновидностями семантических сетей. ***Семантическая сеть*** S в общем виде определяется следующим образом:

$$S = (O, R) = (\{o_i | i = 1, 2, \ldots, k\}, \{r_j | j = 1, 2, \ldots, l\}), \qquad (1.2)$$

где O — множество объектов ПрО ($|O| = k$); R — множество отношений между объектами ПрО ($|R| = l$); o_i — i-й объект ПрО; r_j — j-е отношение между объектами ПрО.

ВМ для заданной ПрО определяется как кортеж:

$$(\{p_i\}, \{f_j\}, \{u_k\}), \qquad (1.3)$$

где p_i — имя понятия ПрО; f_j — функциональное отношение между понятиями; u_k — управляющая структура.

Функциональное отношение f_j задается тройкой

$$f_j = (X_j, F_j, Y_j), \tag{1.4}$$

где $X_j = (x_{j1}, \ldots, x_{jm_j})$ — набор входных переменных для f_j (их типы и значения должны быть известны); F_j — ссылка на процедуру (программный модуль), реализующую вычисление $Y_j = F_j(X_j)$; $Y_j = (y_{j1}, \ldots, y_{jn_j})$ — набор выходных переменных для f_j (их типы известны, а значения должны вычисляться по X_j).

Входные и выходные переменные соответствуют понятиям ПрО. Управляющие структуры u_k реализуют отображения X_j и Y_j в множество разрешенных типов данных. Кроме того, они позволяют приписывать переменным как известные, так и вычисленные значения.

Функциональное отношение может реализоваться только тогда, когда все переменные из X_j имеют допустимые значения.

Заметим, что тройки (1.4) получили в научной литературе название *плекс-элементов,* а формальные грамматики, в терминальные и нетерминальные словари которых могут входить плекс-элементы, — *плекс-грамматик.*

Графически концептуализация ПрО в рамках ВМ изображается графом G:

$$G = (V, U) = (\{x_i\} \cup \{y_j\} \cup \{F_l\}, \{u_k\}). \tag{1.5}$$

Процесс доказательства теоремы существования решения задачи (1.1) отображается на графе G как «волновой процесс», начинающийся в вершинах $(x_1, \ldots, x_m)$ и заканчивающийся, когда «волна» достигнет всех вершин $(y_1, \ldots, y_n)$.

При волновой интерпретации можно детализировать постановку задачи (1.1) и выделить четыре класса задач.

1. Задачи на доказательство. Дано: теория M и имена X_i. Доказать, можно ли определить значения переменных с именами Y_j.

Зададим на графе G отображение Γ: $V \to V$, а также отображения старших порядков $\Gamma^{\pm k}$ и транзитивные замыкания $\hat{\Gamma}^{\pm}$. При этих обозначениях решение задачи существует, если $Y_j \subseteq \hat{\Gamma} X_i$.

2. Задачи на вычисление значений переменных. Дано: теория M, имена X_i, значения $\tilde{X}_i$, имена Y_j. Найти значения $\tilde{Y}_j$.

Решение сводится к решению предыдущей задачи, но с вычислением значений переменных по мере распространения «волны».

3. Задачи на прогнозирование. Дано: теория M и имена X_i. Найти, что можно определить при этих условиях.

Ответом служит множество имен $\hat{\Gamma}X_i$.

Задачи третьего класса полезны, например, при обработке результатов эксперимента, когда требуется найти все, что можно определить по экспериментальным данным.

4. Задачи планирования эксперимента. Дано: теория M, имена Y_j, L — критерий оценки трудоемкости определения значений переменных с именами X. Найти $X_i \subset X | X_i \vdash Y_j$ & $L(X_i) = \text{extr}$.

В задачах четвертого класса минимизируются затраты на постановку эксперимента, в результате которого вычисляются значения переменных с именами Y_j.

Рассмотрим ***теорему существования решения задачи*** в постановке (1.1). Обозначим $P(x)$ предикат входных условий, а $R(x, y)$ — предикат выходных условий; $x = (x_1, \ldots, x_m)$, $y = (y_1, \ldots, y_n)$. Запишем теорему существования в виде

$$\forall x \, (P(x) \Rightarrow \exists y \, R(x, y)). \tag{1.6}$$

Будем рассматривать только конструктивные логические теории, в которых под «существовать» понимается «быть построенным». Другими словами, конструктивное доказательство теоремы существования решения содержит описание процесса построения искомого решения, так как только существование реализуемых объектов может быть конструктивно доказано.

Впервые Н.Н. Непейвода доказал, что различные определения реализуемости эквивалентны [18]. Он же показал, что существует реализуемость, при которой формулам вида $\exists y \, R(y)$ будет соответствовать либо программа вычисления y, либо само значение y [19]. Тогда *любой доказуемой формуле будет соответствовать программа.* Предполагается, что реализации всех аксиом заданы априорно. Для каждого правила вывода $\prod \frac{A_1, \ldots, A_k}{A}$ (или просто $\frac{A_1, \ldots, A_k}{A}$) заданы правила построения реализации выводимой по этому правилу формулы A по реализациям формул $A_1, \ldots, A_k$. Тогда реализация любой выводимой формулы может быть построена прямо по выводу формулы.

Обычно в качестве конструктивной логической теории используют *интуиционистскую логику*, в которой неприменимы законы снятия двойного отрицания и закон исключенного третьего. Для каждого правила вывода в ней записываются программные конструкции, дающие реализации формул, выводимых по этому правилу [19].

Конструктивные доказательства имеют следующие особенности:

- на каждом шаге доказательства применяется некоторое правило вывода;
- в качестве посылок используются только аксиомы или ранее доказанные формулы;
- в доказательстве отсутствуют циклы;
- некоторые шаги доказательства могут использовать леммы, для которых строятся вспомогательные доказательства.

Важно отметить, что каждый шаг доказательства преобразуется во фрагмент программы отдельно от других шагов. Однако, структуру доказательства можно сохранить и в программе, так как «поток фактов в доказательстве» удовлетворяет требованиям «потока данных в программе». Построенные таким способом программы являются хорошо структурированными: в них отсутствуют операторы *goto*.

Существуют два способа *извлечения программы из доказательства.* При первом реализации формул используются непосредственно, поэтому программой является реализация теоремы существования решения. Программа строится в функциональной форме. Шаг вывода: $\prod \frac{F_1, ..., F_k}{F}$. Посылки $F_1, \ldots, F_k$ являются либо аксиомами, либо уже выведенными формулами, поэтому их реализации определены. Реализация следствия F строится по реализациям посылок и по номеру правила вывода.

Любой вывод рассматривается как дерево, ребра которого определяют логическую зависимость шагов вывода, расположенных в вершинах. Важно заметить, что вместо полного вывода теоремы существования (1.6) используется следующее правило: при выполнении предусловий P программы следует выполнение ее постусловий R, т. е при добавлении к системе аксиом формулы P доказывается R: $(P \vdash R) \vdash (P \Rightarrow R)$.

В терминальных вершинах дерева вывода располагаются аксиомы, в корне — последняя выведенная формула. На каждом шаге вывода применяется одно и то же правило:

$$\frac{F_1, ..., F_k, F_1 \,\&\, ... \,\&\, F_k \Rightarrow F}{F}, \quad k \geqslant 1. \tag{1.7}$$

Второй способ извлечения программы заключается в составлении ее оператор за оператором из шагов доказательства теоремы существования (так называемый линейный вывод). В этом случае программа состоит из операторов присваивания и операторов вызова процедур.

Отметим, что:

- рассмотренная система правил вывода не содержала правил для индукции, поэтому в программах не было циклов;

• применяя разные схемы индукции, можно получить разные схемы циклов (в [18] описаны схемы индукции для синтеза цикла *while* и доказана завершимость вычислений).

Сформулируем краткие общие замечания к процессам построения доказательства теоремы существования и извлечению из него программы решения задачи.

1. Только малая часть информации, используемой при синтезе программы, задается в постановке задачи (1.1). Целесообразно хранить знания о ПрО в памяти решателя и использовать их для решения всего множества задач ПрО (а не одной конкретной задачи).

2. Знания должны быть представлены в виде аксиом теории. Таким образом, язык представления знаний определяется программой поиска доказательства. Правила вывода почти всегда фиксированы (хотя Н. Нильсон в [6] приводит примеры изменения набора правил вывода).

3. Знания о ПрО или об отдельной задаче образуют теорию.

4. Число аксиом в практически полезных теориях достигает десятков тысяч.

5. Первой удачной системой, в которой используется дедуктивный синтез программ, является ПРОЛОГ.

6. Универсальные методы синтеза программ требуют длинных доказательств. Однако, к счастью, теории, в которых строятся доказательства разрешимости вычислительных задач, всегда являются в некотором смысле простыми.

7. Для общего метода резолюции количество шагов минимального вывода может превышать экспоненту от числа переменных пропозиционной формулы. Этот результат получен Г.С. Цейтиным еще в 1968 г.

8. В системе ПРОЛОГ, как правило, применяется единичная линейная гиперрезолюция, которая часто обеспечивает приемлемое время вывода.

9. В продукционных системах при дедуктивном выводе следствие выводится из совокупности фактов и правил, причем факты выступают как аксиомы, а правила используются как правила вывода.

10. В технологии концептуального программирования применяется класс теорий, в которых почти отсутствует перебор при построении доказательства существования решения задачи. Такой класс теорий получил название SSR (Structural Synthesis Rules — структурный синтез программ, точнее, система правил структурного синтеза). Минц Г.Е. показал, что система SSR полна в том смысле, что по приведенным правилам из любой системы аксиом в виде предложений вычислимости выводимы точно те формулы, эквиваленты которых выводимы в интуиционистской логике [17].

1.2. Инструментарий концептуального программирования

Мысли не сохраняются,
их надо во что-то воплотить.

А. Уайтхед

Технология концептуального программирования программно реализована в серии программных решателей ПРИЗ: Микро-Приз, Эксперт-Приз. Общим для них является язык УТОПИСТ (Универсальный Транслятор ОПИСаний Теорий). В решателях накоплена значительная база описаний ПрО (теорий): элементарная математика, физика, электротехника, механика и др.

В Эксперт-Приз ТКП объединена с еще одной эффективной технологией ИИ — экспертными системами (ЭС). На рис. 1.2 представлена укрупненная схема решения задачи в ПРИЗ, а на рис. 1.3 — архитектура этой системы.

Эксперт-Приз предоставляет средства для формирования набора понятий ПрО, с помощью которых описываются объекты и отношения, фигурирующие в прикладной задаче. Таким образом, модель задачи состоит из двух разделов: списка объектов и списка уравнений (рис. 1.4).

Запрос на решение задачи содержит перечень искомых параметров объектов. Результаты моделирования выводятся в окне Results (рис. 1.5).

Основные выводы

1. На основе ТКП разрабатываются решатели задач для хорошо определенных (структурированных) ПрО.

2. Черты естественного интеллекта, присущие ТКП:

- дедуктивный вывод;
- ВМ как средства концептуализации для хорошо структурированных ПрО;
- интуиционистская логика.

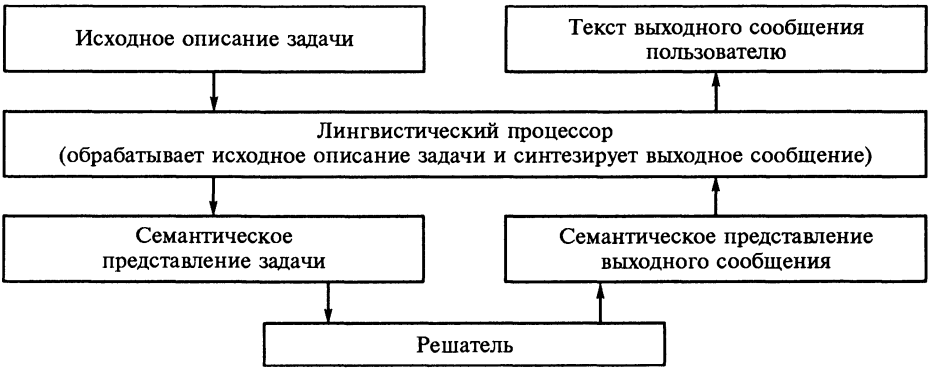


Рис. 1.2. Укрупненная схема решения задачи в ПРИЗ

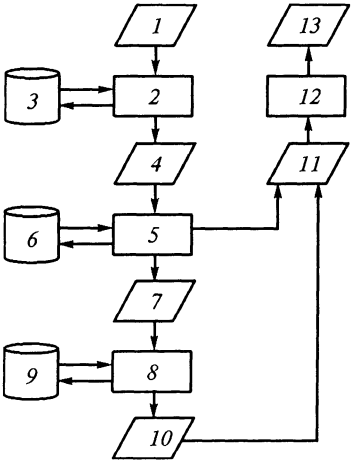


Рис. 1.3. Архитектура ПРИЗ:

1 — исходное описание задачи на языке УТОПИСТ; *2* — макропроцессор; *3* — база макроопределений; *4* — полное описание задачи на языке УТОПИСТ; *5* — транслятор с языка УТОПИСТ; *6* — база ВМ, используемая при трансляции; *7* — построенный путем доказательства алгоритм решения задачи; *8* — генератор (синтезатор) программы по алгоритму; *9* — фрагменты программ, соответствующих отдельным шагам доказательства существования решения задачи; *10* — программа решения задачи; *11* — семантическое представление выходного сообщения; *12* — языковый процессор; *13* — выходное сообщение пользователю

3. Текущее состояние ТКП:

- полностью отработана, доведена до активного практического использования в пакетах типа ПРИЗ;
- инвариантна к ПрО (инвариантность связана с используемым аппаратом ВМ, языком УТОПИСТ, архитектурой программных пакетов).

Вопросы для самопроверки

1. Каково назначение ТКП?
2. Что такое вычислительные или расчетно-логические задачи?
3. Назовите подходы к синтезу программ.
4. В чем состоит основная идея ТКП?
5. Дайте определение понятия «концептуализация».
6. Что понимается под вычислительными моделями и как они описываются?
7. Как определяется функциональное отношение в ВМ?
8. Как графически представляется концептуализация ПрО в рамках ВМ?
9. Какие классы задач можно выделить при волновой интерпретации процесса их решения на графе концептуализации?
10. Сформулируйте теорему существования решения задачи в ТКП.
11. Какой тип логики используется в ТКП и почему?

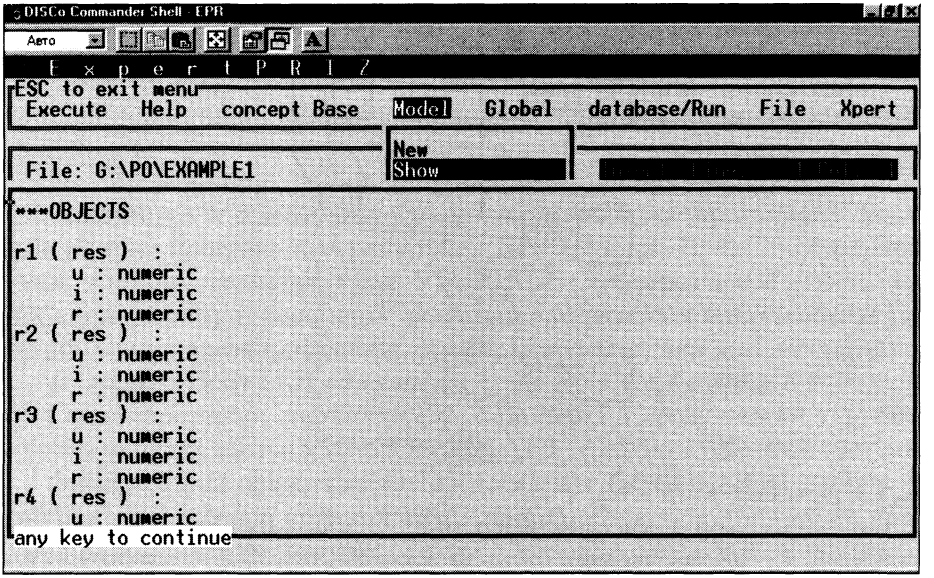


Рис. 1.4. Представление модели задачи в пакете Эксперт-Приз

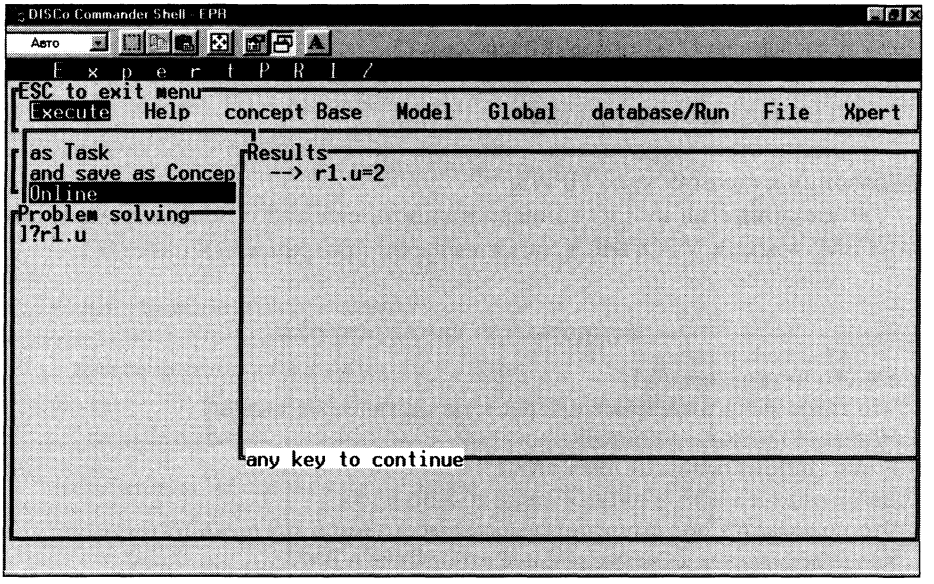


Рис. 1.5. Представление запроса на решение и результатов моделирования в пакете Эксперт-Приз

12. Какие способы извлечения программы решения задачи из доказательства теоремы его существования Вы знаете?
13. Какие знания о ПрО используются в ТКП?
14. Какой класс теорий используется в практических реализациях ТКП и почему?
15. Назовите программные реализации ТКП.
16. Почему ТКП относится к методам ИИ?
17. Каковы перспективы развития ТКП?

Гораздо легче найти ошибку,
нежели истину.

Гёте

2. ТЕХНОЛОГИИ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ОБРАЗОВ

Рассмотрены основные понятия и ключевые принципы автоматического распознавания образов. Реализация данной технологии представлена на примерах ведущих российских систем оптического чтения текстов. Содержание главы соответствует направлению исследований в области ИИ 1.3.

2.1. Основные понятия теории автоматического распознавания образов

Сотри случайные черты,
и ты увидишь — мир прекрасен.

А. Блок

Методы автоматического распознавания образов и их реализация в ***системах оптического чтения текстов*** (OCR-системах — Optical Character Recognition) — одна из самых плодотворных технологий ИИ. В развитии этой технологии российские ученые и разработчики занимают ведущие позиции в мире.

В приведенной трактовке OCR понимается как автоматическое распознавание с помощью специальных программ изображений символов печатного или рукописного текста (например, введенного в компьютер с помощью сканера) и преобразование его в формат, пригодный для обработки текстовыми процессорами, редакторами текстов и т. д.

Сокращение OCR иногда расшифровывают как Optical Character Reader. В этом случае под OCR понимают устройство оптического распознавания символов или автоматического чтения текста. В настоящее время такие устройства при промышленном использовании обрабатывают до 100 тыс. документов в сутки. Промышленное использование предполагает ввод документов хорошего и среднего качества. Это соответствует задачам обработки бланков переписи населения, налоговых деклараций и т. п.

Отметим следующие особенности ПрО, существенные с точки зрения OCR-систем [27]:

- шрифтовое и размерное разнообразие символов;
- искажения в изображениях символов (разрывы образов символов, например, при увеличении изображения; слипание соседних символов и др.);
- перекосы при сканировании;
- посторонние включения в изображениях;
- сочетание фрагментов текста на разных языках;
- большое разнообразие классов символов, которые могут быть распознаны только при наличии дополнительной контекстной информации (дуальные символы, имеющие одно и то же начертание в строчном и прописном вариантах, например, «W» и «w», «S» и «s»; эквивалентные символы, принадлежащие разным алфавитам и имеющие одинаковое начертание, например, «О» в кириллице, латинице и ноль; толерантные символы, т. е. символы, близкие по начертанию, например, «ъ» и «ь», «*l*», «1» и «i»).

Автоматическое чтение печатных и рукописных текстов является частным случаем автоматического визуального восприятия сложных изображений. Многочисленные исследования показали, что для полного решения этой задачи необходимо интеллектуальное распознавание, т. е. «распознавание с пониманием». Однако в настоящее время в технически реализуемых OCR-системах рассматриваемая проблема значительно упрощена и сведена к задаче *классификации по признакам простых объектов*. Эта задача описывается хорошо разработанным математическим аппаратом пороговых отделителей — разделяющими плоскостями [25].

В лучших OCR-системах используется технология распознавания, свойственная человеку. У человека распознавание образа является многоступенчатым (рис. 2.1).

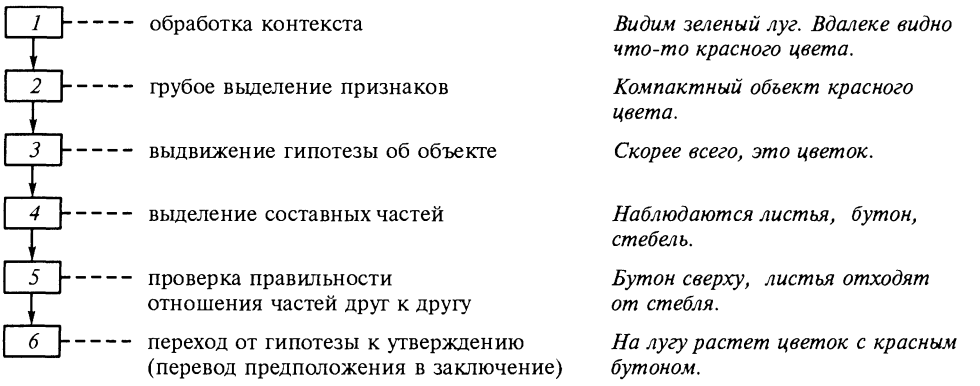


Рис. 2.1. Многоступенчатое распознавание образов человеком

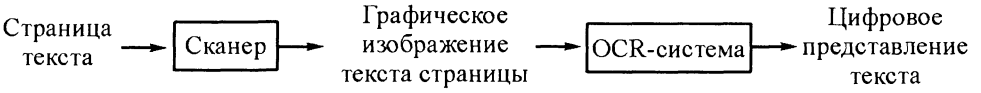


Рис. 2.2. Общая схема распознавания текста

Выделяются *три принципа, на которых основаны все OCR-системы* [27].

1. Принцип целостности образа: в исследуемом объекте всегда есть значимые части, между которыми существуют отношения. Результаты локальных операций с частями образа интерпретируются только совместно в процессе интерпретации целостных фрагментов и всего образа в целом.

2. Принцип целенаправленности: распознавание является целенаправленным процессом выдвижения и проверки гипотез (поиска того, что ожидается от объекта).

3. Принцип адаптивности: распознающая система должна быть способна к самообучению.

На рис. 2.2 представлена общая схема распознавания текста.

Графический образ символа на выходе сканера имеет вид *шейпа,* представляющего собой матрицу из точек, которую можно редактировать поэлементно. На рис. 2.3 приведен пример шейпа буквы «л» или «п». Он ближе к букве «л», но без контекстной обработки утверждать это со 100%-ной уверенностью нельзя.

При контекстной обработке для распознавания «сомнительного» шейпа привлекается информация о результатах распознавания соседних элементов текста. В простейшем случае контекстом служит слово. Например, шейп, изображенный на рис. 2.3, входящий в трехбуквенное слово «е*ь» (обозначен звездочкой), соответствует букве «л», а не «п», так как в словаре системы есть слово «ель», а не «епь».

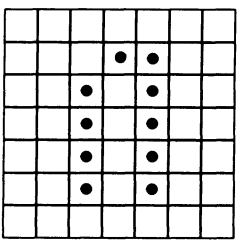


Рис. 2.3. Пример шейпа

Информация об отдельном слове не всегда достаточна для принятия решения. Например, в слове «сто*» в позиции звездочки может располагаться как «л», так и «п». В таких случаях анализируемый контекст включает предложение или несколько предложений (фрагмент текста). Реализация соответствующих механизмов связана с решением проблемы понимания текста на естественном языке (см. § 3.2).

2.2. Примеры программной реализации OCR-систем

В действительности все выглядит иначе, чем на самом деле.

Станислав Ежи Лец

Ведущие российские OCR-системы:

- Fine Reader, Fine Reader Рукопись и Form Reader фирмы ABBYY Software House (http://www.abbyy.ru), позволяющие распознавать как печатные, так и рукописные многоязычные тексты;
- CuneiForm (http://www.cuneiform.ru) фирмы Cognitive Technologies;
- Cognitive Forms фирмы Cognitive Technologies (http://www.cognitive.ru), предназначенная для массового ввода структурированных документов (например, налоговых деклараций, бухгалтерских форм, платежных документов и т. д.).

Работа системы типа Fine Reader включает два крупных этапа.

1. Анализ графических изображений:
   - выделение таблиц, картинок;
   - определение областей распознавания;
   - выделение строк, символов.
2. Распознавание отдельных символов.

Рассмотрим второй этап. Ранее мы определили, что система распознавания реализуется как классификатор. Существуют *три типа классификаторов*:

1) шаблонные (растровые);
2) признаковые;
3) структурные.

Схема классификатора первого типа показана на рис. 2.4. В нем с помощью критерия сравнения определяется, какой из шаблонов выбрать из базы. Самый простой критерий — минимум точек, отличающих шаблон от исследуемого изображения. К достоинствам *шаблонного классификатора* относятся хорошее распознавание дефектных символов («разорванных» или

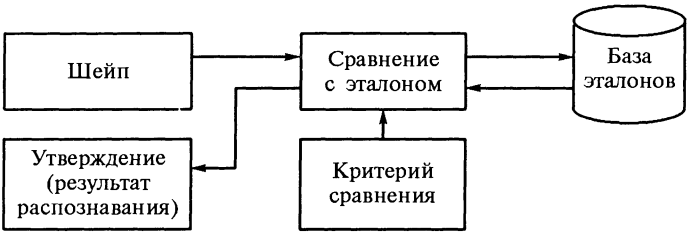


Рис. 2.4. Шаблонный классификатор

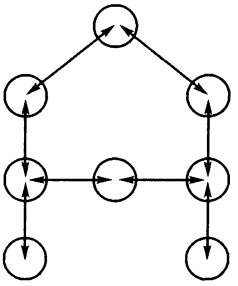


Рис. 2.5. Структурно-пятенный эталон

«склеенных»), простота и высокая скорость распознавания. Недостатком является необходимость настройки системы на типы и размеры шрифтов.

Наиболее распространены *признаковые классификаторы.* Анализ в них проводится только по набору чисел или признаков, вычисляемых по изображению. Таким образом, происходит распознавание не самого символа, а набора его признаков, т. е. производных данных от исследуемого символа. Это неизбежно вызывает некоторую потерю информации.

Структурные классификаторы переводят шейп символа в его топологическое представление, отражающее информацию о взаимном расположении структурных элементов символа. Эти данные могут быть представлены в графовой форме. Такой способ обеспечивает инвариантность относительно типов и размеров шрифтов. Недостатками являются трудность распознавания дефектных символов и медленная работа.

В Fine Reader применяется так называемый *структурно-пятенный эталон* и его фонтанное (от англ. font — шрифт) представление (рис. 2.5). Оно имеет вид набора пятен с попарными отношениями между ними. Подобную структуру можно сравнить со множеством шаров, нанизанных на резиновые шнуры, которые можно растягивать. При этом обеспечиваются все достоинства шаблонного и структурного классификаторов. Также данное представление нечувствительно к различным начертаниям и дефектам символов.

В современных OCR-системах обычно используются все три типа классификаторов, но основным является структурный. Для ускорения и повышения качества распознавания применяются растровый и признаковый классификаторы.

На рис. 2.6 изображена укрупненная схема работы системы Fine Reader.

Особенности распознавания рукописных текстов:

- использование структурно-пятенного эталона с учетом особенностей траектории движения пишущего инструмента (выделяются кольца, дуги, точки, отрезки и другие топологические признаки);
- основным механизмом является выдвижение и подтверждение гипотез;
- использование методов оптимизации при управлении перебором вариантов.

Пользовательский интерфейс Fine Reader иллюстрирует рис. 2.7. В левом дочернем окне представлено исходное изображение распознаваемой страницы, разбитое на блоки текста (1, 3, 5) и рисунков (2, 4). Распознаванию подлежат блоки текста и таблицы, рисунки включаются в формируе-

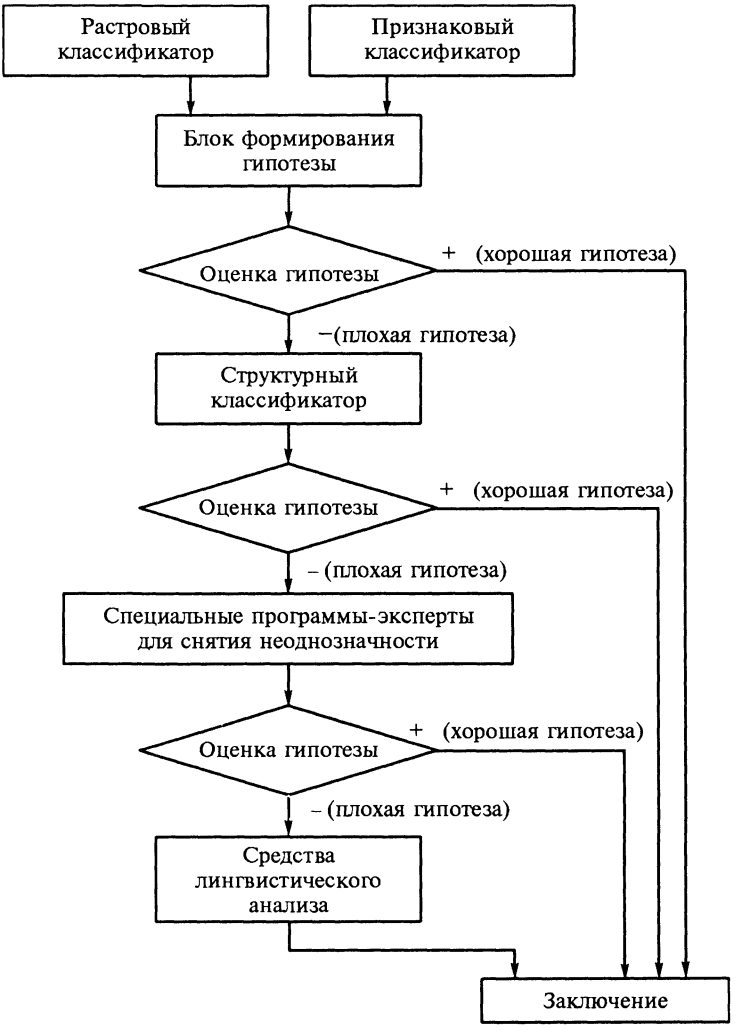


Рис. 2.6. Укрупненная схема работы системы Fine Reader

мый документ без распознавания. Результаты распознавания выводятся в правом дочернем окне. Фрагменты текста, по которым у системы возникли сомнения, выделены фоном.

Fine Reader 7.0 распознает документы на 177 языках. Система может обрабатывать документы, содержащие текст на разных языках. Для 34 языков предусмотрена возможность проверки орфографии.

Fine Reader 7.0 также поддерживает выделение в документах и распознавание штрих-кодов (в том числе двухмерных).

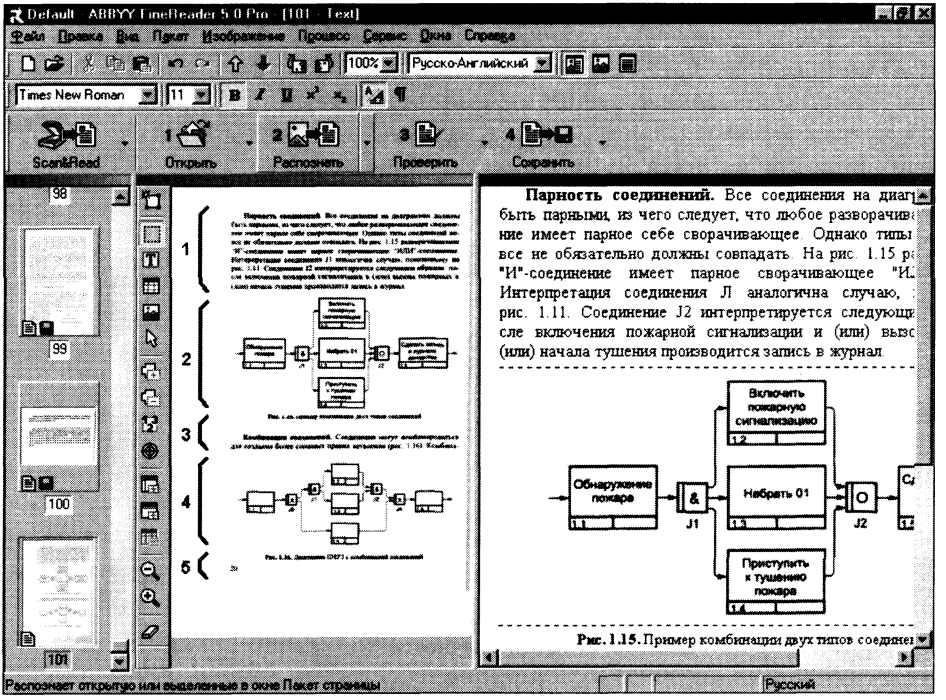


Рис. 2.7. Пользовательский интерфейс системы Fine Reader

В системах типа Fine Reader реализуются интеллектуальные механизмы, характерные для распознающей системы человека: целостное, целенаправленное, адаптивное (настройка на внешние условия и самообучение) восприятие. Экспериментальная проверка на рукописных текстах, написанных более 40 тыс. человек и имеющих суммарный объем более 3 млн изолированных символов, показала, что такие системы дают 1–2 ошибки на 3000 машиночитаемых знаков.

OCR-система Cognitive Forms представляет собой программный комплекс для массового ввода документов, имеющих стандартизованные формы. Его модули, установленные на компьютерах локальной вычислительной сети, способны взаимодействовать друг с другом, образуя конвейер обработки данных, производительность которого может составлять более 10 тыс. страниц в сутки.

Технология ввода документов в стандартизованных формах включает две стадии: подготовительную и основную. На первой стадии создаются шаблоны документов, которые планируется вводить. *Шаблон* описывает свойства документа и входящих в него элементов данных: структуру документа, размер страниц, состав элементов данных, размеры и расположение

соответствующих им полей, типы данных, форматы их представления, наборы допустимых значений и др. Шаблон может быть построен на основе графического представления документа-образца. Для создания и редактирования шаблонов предназначено средство Cognitive Forms Designer.

Основная стадия состоит из шести этапов.

1. Сканирование. Перевод бумажных документов в цифровое графическое представление. Управление данным процессом обеспечивают модуль пакетного сканирования Cognitive Forms ScanPack и модуль постраничного сканирования Cognitive Forms AutoScan.

2. Сортировка и комплектация. Документ может состоять из нескольких страниц, ассоциируемых с разными шаблонами. На этом этапе выполняется группирование полученных ранее графических образов страниц в наборы, соответствующие документам. Указанная задача решается в автоматическом режиме модулем Cognitive Forms Processor, который осуществляет:

- предварительную обработку графического представления и выделение графических примитивов (границ полей, строк текста и др.);
- выбор наиболее релевантного шаблона документа;
- выделение и распознавание элементов данных, значимых с точки зрения оценивания комплектности документа;
- контроль комплектности на основе соответствия последовательности типов страниц структуре, указанной в шаблоне.

3. Корректировка результатов сортировки. Этот этап выполняет оператор, к которому поступают некомплектные документы. Он выясняет причины возникших проблем и устраняет их.

4. Распознавание основной информации. Процесс реализуется модулем Cognitive Forms Processor. Графические представления страниц и распознанные значения элементов данных записываются в БД системы. Для повышения точности распознавания осуществляется логический контроль и контекстный анализ получаемых результатов.

5. Верификация результатов распознавания. Документы, содержащие элементы данных, которые не распознаны либо распознаны не однозначно (например, из-за низкого качества документа или нарушения правил его заполнения), направляются оператору. Для верификации и корректировки результатов распознавания служит модуль Cognitive Forms Editor.

6. Экспорт распознанных документов для передачи внешним приложениям.

Основные выводы

1. OCR-технологии доведены до активного практического использования. Основным направлением их развития является дальнейшая интеллектуализация.

2. Общее решение задачи автоматического распознавания образов должно основываться на организации процесса с такими интеллектуальными составляющими, как целостность восприятия, целенаправленность, предвидение (выдвижение гипотез), максимальное использование контекста и знаний о среде (в пределе — использование модели мира), т. е. учете и реализации интеллектуальных механизмов зрительного восприятия человека.

Важнейшей стороной многоуровневого процесса восприятия является выдвижение гипотез на основе иерархической модели ПрО. В знакомой среде восприятие идет на уровне обобщений (частное — общее), укрупнений (часть — целое) и состоит в подтверждении гипотез на этих уровнях.

3. Автоматическое зрительное восприятие на сегодняшний день не достигает совершенства человеческого восприятия образов. Главная причина этого заключается в неумении строить достаточно полные и семантически выразительные компьютерные модели ПрО.

4. Среди OCR-технологий важное значение имеют специальные технологии решения отдельных классов задач автоматического распознавания образов:

- поиск людей по фотографиям;
- поиск месторождений полезных ископаемых и прогнозирование погоды по данным аэрофотосъемки и снимкам со спутников в различных диапазонах светового излучения;
- составление географических карт по исходной информации, используемой в предыдущей задаче;
- анализ отпечатков пальцев и рисунков радужной оболочки глаза в криминалистике, охранных и медицинских системах.

Для решения этих задач созданы специальные методы и алгоритмы, рассмотрение которых выходит за рамки данного учебного пособия.

Вопросы для самопроверки

1. Дайте определение технологии OCR.
2. Какие особенности ПрО являются существенными для OCR-систем?
3. Что свойственно процессу распознавания образов человеком?
4. Какие принципы лежат в основе технологии OCR?
5. Что такое шейп?
6. Какие OCR-системы Вы знаете?
7. Какие типы классификаторов используются в OCR-системах? Какие достоинства и недостатки присущи классификаторам каждого типа?
8. Что такое структурно-пятенный эталон?
9. В чем заключаются особенности распознавания рукописных текстов?
10. Постройте укрупненную схему работы OCR-системы Fine Reader.
11. Почему OCR-технологии относят к ИИ?
12. Каковы перспективы развития OCR-технологий?

3. АВТОМАТИЗАЦИЯ РАБОТЫ СО ЗНАНИЯМИ, ПРЕДСТАВЛЕННЫМИ В ТЕКСТОВОМ ВИДЕ

Текст является универсальным средством представления, накопления и передачи знаний в человеческом обществе. Поэтому технологии работы с естественно-языковыми текстами (а также с текстами на ограниченном естественном языке) всегда считались важнейшими для ИИ.

Глава посвящена технологиям работы со знаниями, выраженными в текстовом виде. В ней рассмотрены гипертекстовые модели и системы, проблема понимания текста на естественном языке и подходы к ее решению, модели и методы информационного поиска (в том числе показатели эффективности, особенности поиска в Internet, направления интеллектуализации поиска), технологии автоматического реферирования и аннотирования, машинного перевода, автоматической классификации документов. Приведена характеристика открытой справочной лексической системы WordNet, используемой в качестве лингвистического обеспечения интеллектуальных программ, обрабатывающих тексты на естественном языке. Описаны примеры комплексных интеллектуальных программных систем для обработки текстов.

Содержание главы соответствует направлениям исследований в области ИИ 1.4, 2.2.1, 2.2.2, 2.3.1, 2.3.3 и 4.2.

3.1. Основы гипертекстовой информационной технологии*

... И указывают тысячами пальцев
тысячи дорожек для скитальцев.

Гарсия Лорка

Гипертекст (ГТ) — одна из фундаментальных моделей представления знаний, выраженных в текстовом виде. Обычный (одномерный) текст рассматривается как длинная строка символов, читаемая в одном направлении. Многомерный текст (ГТ) включает *точки ветвления,* в которых чтение можно продолжать в нескольких направлениях в зависимости от информационных потребностей читателя.

---

* Содержание параграфа соответствует направлениям исследований в области ИИ 1.4.1, 2.2.1, 2.2.2 и 4.2.

Современные гипертекстовые системы позволяют пользователю самостоятельно формировать альтернативные траектории навигации по ГТ, максимально отвечающие его текущим интересам.

3.1.1. Основные понятия гипертекстовой информационной технологии

В основе ГТ лежат следующие основные идеи.

1. Текст разбивается на фрагменты, представляющие его *семантические единицы* (*сеты*). Между ними устанавливаются *связи,* которые могут наделяться именами.

2. В отличие от обычного текста, который читается последовательно (в порядке, определенном его автором), ГТ можно читать, *двигаясь по разным траекториям,* образованным связанными сетами.

3. Активируемые переходы выбираются читателем (пользователем). Имена (типы) связей облегчают решение задачи выбора перехода. Например, «раздел *А*», «аргументы за...», «определение термина...», «замечания», «детализация положения...» и др.

Под гипертекстом понимается форма организации семантической информации, предусматривающая ее разделение на фрагменты, для каждого из которых заданы переходы к родственным фрагментам. Исторически первым гипертекстовым документом можно считать Библию.

Заметим, что гипертекстовый документ может быть как электронным, так и бумажным. Например, обязательным элементом энциклопедий являются ассоциативные ссылки между статьями или терминами (понятиями). Однако в полной мере функциональность ГТ реализуется лишь в электронных гипертекстовых документах.

В настоящее время под ГТ также понимают многоцелевой информационный фонд, характеризующийся полнотой изложения сведений по определенной тематике и наличием ссылок между статьями.

В гипертекстовом документе может быть представлено несколько *уровней детализации материала.* Такие документы моделируются деревьями или сетями. Если в обычном тексте автором или экспертом расставлены точки ветвления (ссылки), позволяющие читать его, двигаясь по разным траекториям, то текст превращается в ГТ. В графовой модели ГТ вершины соответствуют вычлененным фрагментам текста, а ребра — возможным переходам между ними. Каждый путь на графе представляет отдельную линию прочтения текста.

Таким образом, ГТ как информационная модель интегрирует положительные стороны энциклопедий, монографий и тезаурусов. От энциклопедий ГТ наследует возможности детального представления понятий, быстро-

го просмотра материала (без использования ссылок), алфавитного поиска; от монографий — возможности представления материала с разной степенью глубины и детальности, поиска по оглавлению; от тезаурусов — раскрытие объема и содержания понятий, а также связей между понятиями.

Гипертекстовая информационная технология (ГИТ) — технология обработки семантической информации, основанная на использовании ГТ. Она относится к проблематике ИИ, так как ее содержанием является представление, поиск и обработка семантической информации, выраженной в текстах.

Области применения ГИТ весьма разнообразны:

- информационные ресурсы и технологии Internet;
- гипертекстовые информационно-поисковые системы;
- гипертекстовые информационные модели экономических систем;
- базы данных с гипертекстовой организацией;
- представление электронной документации (в том числе, контекстно-зависимой и ситуативно-зависимой справки по программным средствам);
- электронные записные книжки;
- электронные картотеки, словари, энциклопедии, справочники;
- обучающие системы;
- экспертные системы;
- организация пользовательского интерфейса и др.

Коротко поясним основные аспекты *применения ГИТ в Internet.* Информационные ресурсы Internet разнородны и динамичны. Их невозможно представить в виде единой БД. Гипертекст в Internet применяется с 1993 г. в рамках технологии World Wide Web (WWW) — «всемирной паутины», позволяющей перемещаться по сети гипертекстовых документов. В соответствии с протоколом передачи гипертекста HyperText Transport Protocol (HTTP) минимальной неделимой единицей данных, предназначенной для межмашинного обмена, является текст, записанный на языке разметки гипертекста HyperText Markup Language (HTML). Файл с этим текстом представляет собой гипертекстовый документ, называемый *HTML-страницей* или *web-страницей.* HTML-страница содержит описание структуры документа, в тело которого в виде унифицированного указателя ресурса (Uniform Resource Locator — URL) могут входить ссылки на фрагменты данного документа и других документов.

Взаимосвязанная совокупность HTML-страниц, расположенных на одном web-сервере, образует *web-сайт.*

Гипертекстовый документ, представленный на HTML, может включать не только текст, но и таблицы, фрагменты исполняемого сервером или компьютером пользователя программного кода (скрипты, апплеты), а также ссылки на цифровые объекты (графические изображения, звук, видео, ани-

мацию и др.). Отметим, что возможности HTML как языка описания данных выходят за рамки только лишь включения в документ гипертекстовой разметки. В частности, язык HTML позволяет:

- определять структуру документа (заголовки и области различных уровней);
- представлять собственно содержимое документа;
- устанавливать оформление содержимого (способ представления информации — отступы, шрифты, цвета, выравнивание, параметры таблиц и т. д.);
- задавать ссылки для вставки внешних компонентов — рисунков, элементов пользовательского интерфейса, программных объектов и др. (их вставка или активация происходит на этапе загрузки страницы);
- включать в документ фрагменты программного кода (скрипты);
- определять гиперссылки, ассоциируемые с различными информационными элементами документа для организации переходов и вызова функций.

Логически единая система HTML-страниц может быть физически рассредоточена по сети. Система URL позволяет как размещать, так и собирать ресурсы, на которые ссылается ГТ.

Далее будут рассмотрены следующие проблемы и задачи, связанные с ГИТ:

- модели ГТ;
- инструментальные средства для создания ГТ;
- гипертекстовые информационно-поисковые системы (ГИПС): классификация, методы поиска, критерии смыслового соответствия;
- методы извлечения знаний для гипертекстовых систем;
- автоматизация построения ГТ;
- место ГИТ среди технологий ИИ.

3.1.2. Формализованная модель гипертекста

В основе моделей ГТ лежит понятие ***информационно-справочной статьи (ИСС),*** выступающей в качестве информационной единицы ГТ.

В формализованной модели ГТ ИСС соответствует информационному объекту, содержание которого характеризуется смысловым единством и логической целостностью. В конкретных технологиях ИСС называют поразному: страница, статья, тема и др. Она может включать информацию, представленную в разных формах: текст, таблицы, фрагменты программного кода (макросы, скрипты), внедренные цифровые объекты, а также ссылки на подобные объекты (графика, звук, видео, управляющие элементы пользовательского интерфейса и т. д.), включаемые в ИСС при ее загрузке.

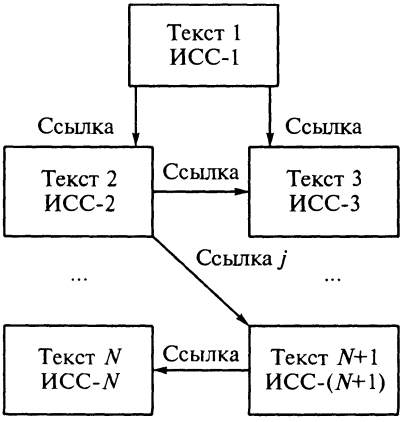


Рис. 3.1. Графовая модель ГТ

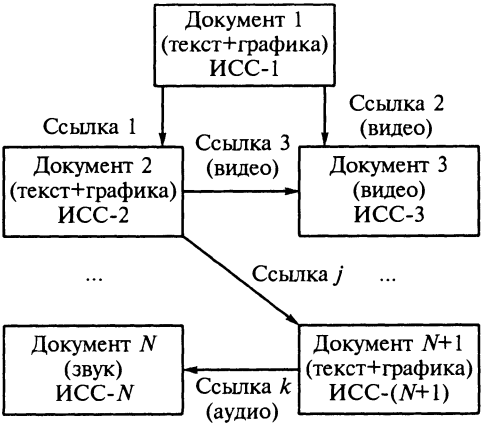


Рис. 3.2. Графовая модель гипермедиа

Элементам ИСС могут быть присвоены метки, уникальные в рамках ИСС. Кроме того, элементы (слова, фразы, предложения, ячейки таблиц, пиктограммы, фрагменты изображений, кнопки и др.) могут наделяться интерактивным поведением. Такие элементы называются ***гиперссылками.*** При воздействии на гиперссылку (например, щелчке на ней мышью) инициируется переход:

- к началу другой ИСС;
- фрагменту другой ИСС, начинающемуся с элемента, который имеет указанную метку;
- фрагменту данной ИСС, начинающемуся с элемента, который имеет указанную метку.

Таким образом, гиперссылки задают направления переходов между ИСС и фрагментами ИСС, т. е. фактически соответствуют точкам ветвления при чтении документа. Сказанное иллюстрируют рис. 3.1 и 3.2. На рис. 3.1 все ИСС являются текстовыми фрагментами, на рис. 3.2 представлен общий случай.

Гиперссылка содержит указатель на ИСС и, возможно, ее фрагмент. В Internet подобные указатели представляются в виде URL, задающих адреса соответствующих ресурсов. Гиперссылки, указывающие на фрагменты текущей ИСС, называются локальными. Гиперссылки, указывающие на другие ИСС, называются глобальными.

Графические иллюстрации и мультимедийные представления, содержащие интерактивные элементы, называются ***гиперграфикой*** и ***гипермедиа*** соответственно*. Эти же понятия часто используются по отношению к до-

* Считается, что гипермедиа охватывает также гипертекст и гиперграфику.

кументам, включающим гиперграфику и гипермедиа. На данном уровне рассмотрения гиперграфика и гипермедиа описываются теми же моделями, что и гипертекст.

С точки зрения программной реализации формализованная модель ГТ состоит из двух слоев. Первый слой представляет отображаемое на экране содержимое документа, в котором гиперссылки по умолчанию выделены цветом, подчеркиванием или изменением шрифта. Адреса переходов (идентификаторы ИСС и метки их фрагментов) хранятся во втором, скрытом слое модели.

Для работы с гипертекстовой системой, включающей множество связанных документов, не требуется «сборка» интегрального документа. Входящие в систему документы могут храниться на одном или множестве компьютеров (узлах сети). При этом физически распределенная система является логически единой.

В формализованной модели ИСС описывает кортеж:

$$(x_0, x_1, \ldots, x_{11}), \tag{3.1}$$

где x_0 — имя ИСС; x_1 — заголовок ИСС; x_2 — аннотация ИСС; x_3 — точка входа в ИСС; x_4 — множество текстовых фрагментов, входящих в ИСС; x_5 — множество цифровых информационных объектов, входящих в ИСС (графические изображения, видео и т. д.); x_6 — множество программных объектов, входящих в ИСС; x_7 — справка по ИСС; x_8 — признак ускоренного просмотра ИСС; x_9 — признак детального просмотра ИСС; x_{10} — список гиперссылок внутри ИСС; x_{11} — список гиперссылок между ИСС.

На рис. 3.3 условно представлена структура ГТ, созданного по модели (3.1). В ней выделены три ИСС: *A*, *B* и *C*. Во всех ИСС обязательными являются точка входа, имя, заголовок и аннотация. Остальные элементы являются необязательными.

Имя служит формальным идентификатором ИСС и используется для ее адресации программными средствами. В рамках ГТ все ИСС должны иметь уникальные (т. е. несовпадающие) имена. *Заголовок* представляет содержательное название ИСС.

Если на ИСС не указывают гиперссылки из других ИСС, то она становится главной темой и включается в *список главных тем ГТ*. Если ИСС не имеет исходящих внешних ссылок, то на текущий момент времени эта ИСС заканчивает один или множество путей навигации по ГТ.

Деление основных элементов содержимого ИСС на три группы (x_4, x_5, x_6) обусловлено удобствами программной реализации гипертекстовых редакторов и скрыто от пользователей.

Ускоренный просмотр помогает пользователю оперативно ознакомиться с ИСС. Часто линию ускоренного просмотра ИСС образуют элементы x_1 и x_2 (заголовок и аннотация, отражающие основные идеи, представленные в ИСС).

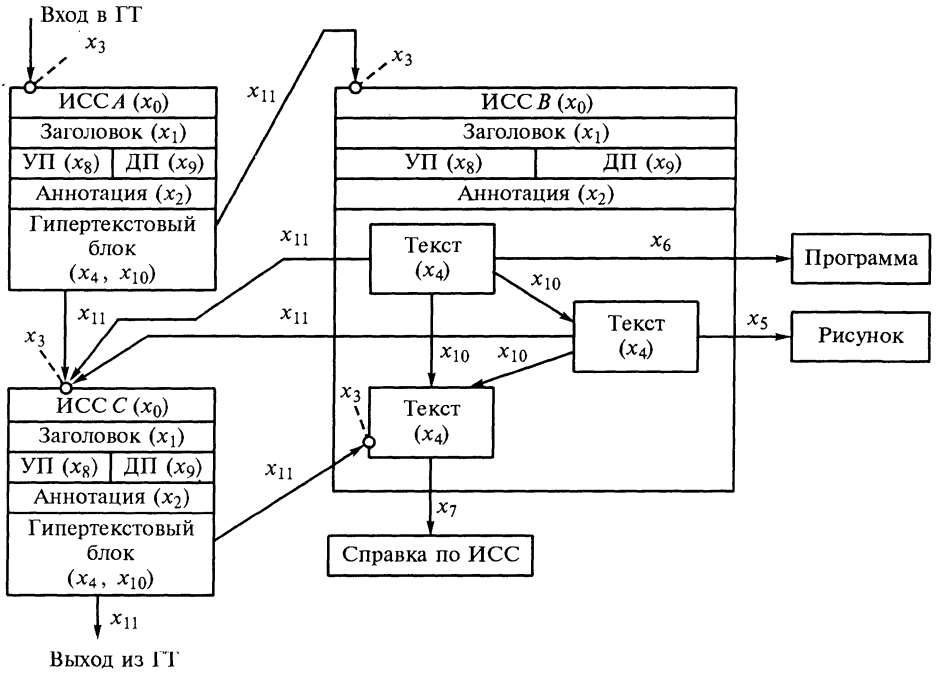


Рис. 3.3. Структура гипертекста, описываемого (3.1):
УП — признак ускоренного просмотра ИСС; ДП – признак детального просмотра ИСС

Активация признака *детального просмотра* обеспечивает представление всего содержимого ИСС. В данном режиме пользователь может пройти по любому пути, включающему элементы x_4, x_5, x_6 и x_7. Поскольку объем ИСС в принципе не ограничивается, предусмотрена *справка* x_7, которая представляет дополнительную информацию, связанную с содержанием ИСС.

Элементы x_7, x_8, x_9, x_{10} и x_{11} реализуются через интерактивные компоненты пользовательского интерфейса, обеспечивающие навигацию по ГТ.

3.1.3. Условно-типовая модель гипертекста

Один из недостатков формализованной модели ГТ связан с отсутствием в ней возможности явного определения типов гиперссылок. В условно-типовой модели *все гиперссылки имеют явно указанный тип.* Данная модель ГТ включает тезаурус, список главных тем и совокупность указателей. Обязательным компонентом является ***тезаурус*** ПрО, к которой относится информационная система (ИС). Приведем три определения понятия «тезаурус»:

1) тезаурус — упорядоченный перечень терминов, в котором отражены семантические отношения между ними [29];

2) тезаурус — свод знаков, терминов, кодов и отношений между ними, которые используются в процессе обмена сведениями, сообщениями [39];

3) тезаурус — автоматизированный словарь, отображающий семантические отношения между лексическими единицами дескрипторного информационно-поискового языка и предназначенный для поиска слов по их смысловому содержанию*.

Каждый термин в тезаурусе снабжается его текстовой характеристикой (статьей). Тезаурус позволяет пользователю ГТ уточнять как содержание (смысл), так и объем интересующего его термина.

Для упрощения работы с ГТ, а также повышения эффективности поиска по нему как в полуавтоматическом режиме (с участием человека), так и в автоматическом режиме (в ГИПС) в условно-типовую модель ГТ включаются список главных тем и указатели.

Список главных тем делит ГТ на сегменты, соответствующие более или менее независимым частям (срезам или аспектам) ПрО. Таким образом, он отражает самое общее представление о тематике ГТ.

Указателем называется упорядоченная установленным образом последовательность информационных объектов (понятий, выражений, обозначений и т. п.), ссылающихся на ИСС, в которых эти объекты упоминаются. В зависимости от характера объектов указатели подразделяются на предметные, именные, событийные, библиографические и др. По принципу упорядочения различают алфавитные, хронологические, систематические и прочие виды указателей. Гипертекст может включать один или несколько указателей.

В лингвистике выделено около 200 *семантических типов отношений* [3]. Наиболее часто употребляются 10 типов, используемых в условно-типовой модели. Их обозначения расшифрованы в табл. 3.1.

Таблица 3.1

Тип связи	Обозначение
синоним	СН
род—вид	РВ
вид—род	ВР
часть—целое (укрупнение)	ЧЦ
целое—часть (декомпозиция)	ЦЧ
процесс—надпроцесс	ПН
процесс—подпроцесс	ПП
причина—следствие	ПС
следствие—причина	СП
ассоциация	АС

* *Першиков В.И., Савинков В.М.* Толковый словарь по информатике. — М.: Финансы и статистика, 1991. — С. 395.

Графовой интерпретацией условно-типовой модели является *семантическая сеть*. Теория семантических сетей будет рассмотрена в § 5.3. Здесь же отметим, что можно построить несколько вариантов формального описания условно-типовой модели. Один из таких вариантов в нотации БНФ выглядит следующим образом:

ГТ ::= <тезаурус>[список главных тем][список указателей]
тезаурус ::= <ИСС>[ИСС...]
ИСС ::= <имя><заголовок><текстовая информация>
[список ИСС, связанных с данной ИСС]
имя ::= <строка символов, служащая уникальным идентификатором ИСС>
заголовок ::= <строка символов>
список ИСС, связанных с данной ИСС ::= <тип родства><имя>
[(<тип родства><имя>)...]
тип родства ::= СН | РВ | ВР | ЧЦ | ЦЧ | ПН | ПП | ПС | СП | АС
список главных тем ::= <имя ИСС, не имеющей входящих
гиперссылок типов РВ, ЦЧ, ПП>
[(имя ИСС, не имеющей входящих
гиперссылок типов РВ, ЦЧ, ПП)...]
список указателей ::= <указатель>[указатель...]
указатель ::= <понятие><список ИСС>[(<понятие><список ИСС>)...]
<понятие> ::= <строка символов>
<список ИСС> ::= <имя>[имя...]

В рамках условно-типовой модели ИСС включает имя, заголовок, собственно текст (содержимое) и список ссылок на ИСС, связанные с данной ИСС различными типами отношений. При этом ссылки относятся только к «ближайшим» родственникам. Такой список ссылок образует *локальный справочный аппарат ИСС*. Он может быть организован тремя способами. Первый способ — в виде списка. При втором способе ссылки внедряются в текст (как в энциклопедиях). Третий способ является комбинированным. Часть ссылок помещаются после заголовка статьи в виде списка, оставшаяся часть — в самом тексте.

При отображении ГТ на экране имена ИСС заменяются их заголовками.

3.1.4. Инструментальные средства для создания гипертекста

Существует большое число инструментальных средств для создания ГТ. Благодаря широкому использованию ГТ в ИС практически любой инструментарий разработки ИС включает функции для построения ГТ. В частности, данные функции реализуются в средствах разработки электронной документации (например, Adobe Acrobat), авторских системах, редакторах презентаций, издательских системах, редакторах web-страниц и др. Для формирования представлений о возможностях гипертекстового инструмен-

тария дадим характеристику четырем системам, специально предназначенным для создания ГТ:

- Microsoft Windows Help (WinHelp);
- HTML Help;
- HyperRef;
- АСФОГ.

WinHelp и *HTML Help* представляют собой стандартные технологии построения и работы с гипертекстовыми справочниками для платформы Windows. Технология WinHelp была реализована фирмой Microsoft еще в Windows 3.0. В дальнейшем на смену ей пришла технология HTML Help. Современные версии операционных систем семейства Windows содержат средства для работы с гипертекстовыми справочниками, созданными с помощью обеих технологий, однако базовой технологией является HTML Help*.

Системы WinHelp и HTML Help позволяют формировать самые разнообразные ГТ: электронные руководства, справочники, энциклопедии, пособия и др. Однако главное назначение данных технологий — реализация контекстно-зависимых гипертекстовых справочников по программным продуктам. Такие справочники являются неотъемлемым компонентом прикладных программных систем. По умолчанию они вызываются клавишей F1 или через меню «Справка». Информация, отображаемая в окне справочника после его вызова, зависит от текущего режима работы приложения, с которым он связан. Поэтому подобные справочники называются *контекстно-зависимыми.*

Создание гипертекстового справочника по программному продукту состоит из шести основных этапов.

1. Определение структуры справочника и его разделов.

2. Подготовка текста и графических иллюстраций справочника. Определение гипертекстовых ссылок. Формирование файлов тем (ИСС) и графических файлов, включая задание контактных областей для гиперграфики.

3. Создание файла проекта справочника.

4. Компиляция исходных файлов тем, графических файлов и файла проекта с формированием файла справочника.

5. Программная реализация модуля приложения, обеспечивающего доступ к справочнику.

6. Тестирование и отладка справочника.

Первый этап является наиболее сложным и трудно формализуемым. В рамках него специфицируются:

- назначение продукта, для которого создается справочник;
- категории пользователей продукта;

* Начиная с версии 2.0, эта технология называется Microsoft Help и входит в состав Microsoft Visual Studio.NET.

- рыночный сектор, на который ориентирован продукт;
- функции и характеристики продукта, представляемые в справочнике;
- основные разделы справочника и их примерное содержание;
- соглашения, фиксирующие стиль, дизайн и оформление справочника.

Гипертекст в формате WinHelp реализуется в виде файла с расширением HLP (help-файла). Представление и взаимодействие со справочником обеспечивает программа WINHELP.EXE, входящая в состав Windows.

Информационно-справочная статья в WinHelp называется темой (topic). Тема самого верхнего уровня представляет содержание справочника. С каждой темой может быть ассоциирован перечень ключевых слов для поиска. Темы связываются друг с другом гиперссылками. Другой механизм задания связей между ними — определение просмотровых последовательностей, по которым можно перемещаться с помощью кнопок «вперед» и «назад».

Доступ к нужной информации в справочнике обеспечивают следующие способы:

- выбор темы в содержании;
- переход по гиперссылке;
- переход по просмотровой последовательности;
- возврат назад к предыдущей теме в списке пройденных тем;
- выбор темы в списке пройденных тем (истории работы со справочником);
- поиск темы по ключевому слову;
- полнотекстовый поиск в справочнике;
- переход к теме, на которой установлена закладка;
- обращение к теме по контексту из вызывающего приложения.

HLP-файл формируется на основе файлов с текстом в формате RTF с помощью специального компилятора. Для вызова справочника из приложения служит функция Windows API WinHelp().

Гипертекст в формате HTML Help реализуется в виде файла с расширением CHM. Представление и взаимодействие со справочником обеспечивают программные компоненты браузера Internet Explorer (начиная с версии 4.0). Таким образом, для использования CHM-справочника обязательно наличие данного браузера.

Справочники HTML Help могут включать ГТ, графические изображения в форматах GIF, JPEG и PNG, компоненты ActiveX, а также скрипты на Java и Visual Basic. Информация в CHM-файле хранится в сжатом виде. Степень компрессии составляет примерно 8:1. При сжатии графики используются алгоритмы компрессии без потери информации.

Информационно-справочная статья в HTML Help называется страницей. Способы доступа к нужным сведениям в HTML Help и WinHelp анало-

гичны. Окно CHM-справочника может включать следующие навигационные панели:

- «Содержание», на которой представлена иерархическая структура справочника;
- «Указатель» для поиска по ключевым словам;
- «Избранное» для определения закладок;
- «Поиск», содержащую средства для полнотекстового поиска.

CHM-файл формируется на основе файлов в формате HTML с помощью специального компилятора. Для вызова справочника из приложения служит функция HTML Help API HtmlHelp().

К достоинствам HTML Help относятся:

- мощные средства языка HTML, включая каскадные таблицы стилей;
- возможности использования компонентов ActiveX и скриптов;
- тесная интеграция с технологиями Internet;
- возможность создания составных гипертекстовых справочников, объединяемых во время выполнения.

Гипертекст в формате HTML Help может быть разработан с помощью различных инструментальных средств. Наиболее популярными из них являются HTML Help Workshop фирмы Microsoft и KeyTools фирмы KeyWorks Software. Система Anet Help Tool российской фирмы Anet Soft позволяет создавать ГТ в формате как HTML Help, так и WinHelp.

Инструментальная среда HyperRef предназначена для построения электронных гипертекстовых изданий большого объема. Она разработана в МЭИ (ТУ) под руководством А.И. Тихонова. HyperRef поддерживает следующие типы информационных объектов: текстовые экранные страницы, графические изображения, исполняемые модули. Объекты объединяются как в линейные последовательности, метафорой которых является глава или раздел книги, так и в гипертекстовую сеть. В визуальных объектах могут быть определены интерактивные элементы, используемые для организации гиперссылок. HyperRef поддерживает типизацию гиперссылок и содержит средства навигации по ГТ с учетом ограничений, обусловленных типами ссылок.

В состав HyperRef входят:

- диалоговый инструментарий автора (конструктор);
- пользовательская программа для работы с ГТ (исполнитель);
- набор утилит, позволяющих осуществлять поточный ввод информации, контролировать и восстанавливать целостность электронных гипертекстовых документов и т. д.

В HyperRef предусмотрены средства, присущие фактографическим и полнотекстовым БД: словари ключевых слов, оглавления, средства выполнения сложных запросов и автоматической индексации текстов.

Автоматизированная система формирования и обработки гипертекстов (АСФОГ) создана в МЭСИ и предназначена для моделирования экономических объектов и процессов на основе представления информационного фонда ПрО в виде ГТ [39]. АСФОГ целесообразно использовать для моделирования слабоструктурированных ПрО, когда поиск текстовой информации в традиционных линейных и иерархических структурах неэффективен из-за их неадекватности реальной сетевой структуре информационных объектов, представляющих эти ПрО.

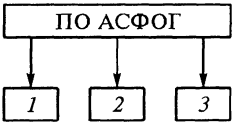


Рис. 3.4. Функциональная структура ПО АСФОГ:

1 — работа с тезаурусом; *2* — работа с информационными статьями (текстом); *3* — работа с алфавитным словарем

Программное обеспечение АСФОГ реализовано в трех подсистемах (рис. 3.4). Первая подсистема выполняет четыре функции:

1.1 — поиск в тезаурусе;

1.2 — поддержка ускоренного просмотра;

1.3 — формирование отчетов;

1.4 — поддержка формирования и корректировки тезауруса.

Функция 1.1 обеспечивает:

- поиск по связям с учетом их типов;
- контекстный поиск по связям.

Вторая подсистема выполняет пять функций:

2.1 — создание ИСС с помощью текстового редактора типа Word;

2.2 — коррекция ИСС;

2.3 — доступ к ИСС;

2.4 — формирование и печать отчетов по ИСС;

2.5 — импорт и экспорт файлов, содержащих ИСС.

Третья подсистема выполняет четыре функции:

3.1 — алфавитная сортировка (лексико-графическое упорядочение) заголовков ИСС;

3.2 — контекстный поиск ИСС по заголовку;

3.3 — поддержка ускоренного просмотра словаря;

3.4 — печать информации из словаря.

Сравнение основных возможностей рассмотренных гипертекстовых технологий представлено в табл. 3.2.

Таблица 3.2

Технология	Возможности технологии									
	Точки входа в ГТ через	Типизация гиперссылок	Функции поиска	Возврат назад по пройденному пути в ГТ	Просмотр пройденного пути в ГТ (истории работы с ГТ)	Установление закладок	Создание аннотаций	Поддержка просмотровых последовательностей	Включение в ГТ мультимедийных компонентов	Включение в ГТ программных компонентов или вызов внешних программных модулей
Windows Help	Список главных тем; словарь (указатель); средства поиска	–	По заголовкам ИСС; по ключевым словам; по тексту ИСС	+	+	+	+	+	+	+
HTML Help	То же	–	То же	+	+	+	+	+	+	+
Hyper-Ref	»	+	По заголовкам ИСС; по ключевым словам; по тексту ИСС; по типам гиперссылок, связывающих искомые ИСС с текущей ИСС	+	+	+	+	+	+	+
АСФОГ	»	+	По заголовкам ИСС; по ключевым словам; по типам гиперссылок, связывающих искомые ИСС с текущей ИСС	+	+	–	–	+	–	–

Примечание. «+» — наличие возможности; «–» — отсутствие возможности.

3.1.5. Гипертекстовые информационно-поисковые системы

Гипертекстовая информационная технология используется при организации больших массивов текстовых документов и реализации методов поиска информации в них.

Информационный поиск — совокупность операций, методов и процедур, направленных на отбор данных, хранящихся в ИС и соответствующих заданным условиям.

Информационно-поисковые системы (ИПС) подразделяются на три класса:

- документальные;
- фактографические;
- гипертекстовые (ГИПС).

Документальные ИПС хранят и выдают сведения о документах, основное содержимое которых представлено в виде связанного текста на ЕЯ. Признаки документа, отражающие его содержание в ИПС, называют *поисковым образом,* а признаки запроса к ИПС — *поисковым предписанием.* Процедура перевода документа и запроса в форму представления, принятую в ИПС, называется *индексированием.* При сопоставлении поискового образа и поискового предписания используется тот или иной *критерий смыслового соответствия (релевантности).*

Первые ИПС были предназначены для поиска книг в библиотеках и получили название *библиографических.* Позже их стали применять и для поиска документов в больших хранилищах и стали называть документальными.

Основным объектом информационного фонда документальной ИПС является аннотация (реферат) и библиографическое описание документа (книги, события, предмета). Реферат (аннотация) выражается на ЕЯ и отражает основные характеристики документа, представляющие интерес для пользователей. Предполагается, что в подобном описании можно выделить ряд слов и словосочетаний, число которых значительно меньше общего числа слов в описании. В то же время выделенная информация достаточно точно характеризует описание. Такие слова и словосочетания называются *ключевыми словами* или *дескрипторами.* Запрос к документальной ИПС формулируется в виде перечня дескрипторов, которые по мнению пользователя характеризуют искомый документ.

При вводе в ИПС нового объекта (реферата) его дескрипторы автоматически включаются в словарь дескрипторов. Каждому дескриптору присваивается номер, называемый индексом дескриптора. Совокупность индексов, соответствующих полному набору дескрипторов реферата, составляет его поисковый образ. Новый поисковый образ снабжается уникальным идентифика-

тором (регистрируется) и включается в массив поисковых образов. Тем же идентификатором помечается новый реферат, заносимый в массив рефератов.

Поиск в дескрипторной ИПС организуется следующим образом. Запрос, сформулированный на ЕЯ, подвергается анализу, в рамках которого в нем выделяются дескрипторы, входящие в словарь дескрипторов. Их совокупность образует поисковое предписание, соответствующее запросу. Оно сопоставляется с поисковыми образами, в результате чего определяется их релевантность. Если поисковый образ и предписание релевантны, то из поискового образа извлекается идентификатор реферата, выдаваемого пользователю. Ответом на запрос является множество рефератов, соответствующих отобранным в процессе поиска идентификаторам.

В целях ускорения поиска для каждого дескриптора в словаре дескрипторов указывается список идентификаторов рефератов, в которых он встречается. Такая информационная структура ИПС называется *индексом.*

Заметим, что с помощью дескрипторов можно лишь приблизительно отразить смысл документов. Это же относится к переводу запросов в поисковые предписания. Сказанное обусловливает то, что документальная ИПС может выдать рефераты, не относящиеся к поисковому запросу, или не найти рефераты, которые соответствуют ему.

Документальный поиск относится к числу сложных информационных процессов, поскольку он связан с проблемой оценивания смыслового соответствия документа и запроса. Из-за субъективности и неоднозначности подобного оценивания этот вид поиска в принципе не может быть исчерпывающе точным и полным, в нем всегда будет присутствовать элемент нечеткости.

Развитием поиска по дескрипторам является *полнотекстовый поиск,* реализуемый, например, в поисковых машинах Internet (см. § 3.2). В системах, использующих данный вид поиска, индекс формируется на основе всех слов и словосочетаний, содержащихся в документах, за исключением служебных неинформативных слов (союзов, предлогов, местоимений и т. п.). При индексировании с помощью словарей и средств морфологического анализа слова приводятся к базовой грамматической форме (именительный падеж, единственное число и т. д.).

В ***фактографических ИПС*** хранятся не документы, а собственно сведения (факты) об объектах ПрО. Подобные ИПС реализуются, в частности, на основе реляционных БД. С точки зрения обеспечения релевантности результатов поиска (выборки данных) запросу фактографический поиск в отличие от документального является точным и полным.

В ***гипертекстовых ИПС*** кроме содержимого документов отражается их семантическая структура. Поэтому по глубине формализации ГИПС занимают промежуточное положение между документальными и фактографическими ИПС.

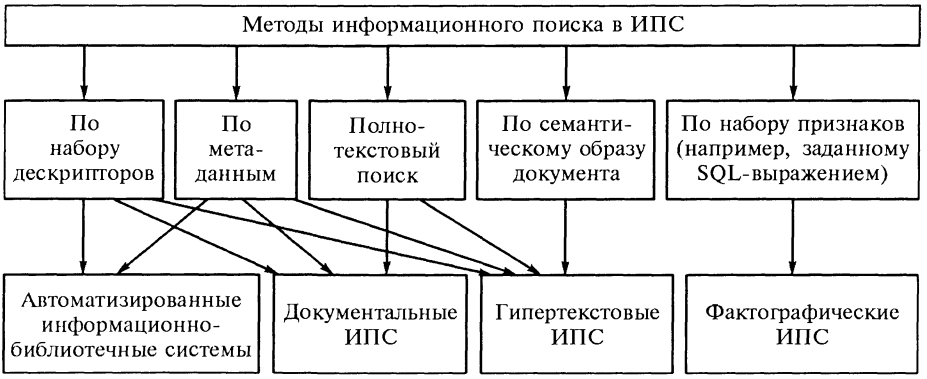


Рис. 3.5. Классификация методов информационного поиска в ИПС

Еще одно направление развития технологии документальных ИПС связано со структуризацией и унификацией сведений о документах. Такие сведения по отношению к исходным документам играют роль *метаданных* (см. гл. 4). Примером метаданных служит библиографическое описание, содержащее информацию об авторах документа, дате его создания, объеме, форме представления и т. д. Ключевые слова также относят к метаданным.

Поиск по метаданным сближает технологии документальных и фактографических ИПС. С одной стороны, метаданные представляют документы. С другой стороны, некоторые элементы метаданных допускают четкое определение релевантности запроса и записи в БД (экземпляра метаданных, ассоциируемых с конкретным документом), что характерно для фактографических ИПС. В настоящее время хранилища метаданных обычно реализуются на основе реляционных и XML-ориентированных БД и используют механизмы поиска, воплощаемые в соответствующих системах управления БД (СУБД).

Классификация методов информационного поиска в ИПС представлена на рис. 3.5.

Введем следующие обозначения: D — множество документов в информационном хранилище, $d_i \in D$ — i-й документ, $D_j \subseteq D$ — подмножество документов. В данном контексте под документом будем понимать как собственно текстовый или гипертекстовый документ, так и отдельную запись в БД.

Зададим на D оценку смысловой близости пары документов $r(d_i, d_j) \geqslant 0$. При $r = 0$ документы d_i и d_j эквивалентны по смыслу. Для семантически несопоставимых документов r не определена. Также введем оценки ряда важных свойств документов: $S = (S_1, S_2, \ldots, S_k)$, $k > 0$. Пусть

оценка каждого свойства S_f выражается действительным числом, принадлежащим некоторому интервалу. Для определенности примем, что чем больше значение S_f, тем важнее для пользователя документ.

Поисковый запрос может рассматриваться как виртуальный документ z. В идеальном случае ($r(z, d_i) = 0$) ему точно соответствует документ d_i.

Используя введенные обозначения, определим следующие *виды поиска.*

1. Найти $(D_j \subseteq D) \mid r(z, d_i \in D_j) \to \min$. Если $D_j = \varnothing$, то в D нет документов, релевантных запросу. При $|D_j| = 1$ есть единственный подходящий документ. Если же $|D_j| > 1$, то таких документов несколько.

2. Найти $(D_j \subseteq D) \mid r(z, d_i \in D_j) \leqslant \Delta$, где Δ — оценка наибольшего допустимого расхождения смыслов запроса и искомых документов.

3. Найти $(D_j \subseteq D) \mid S_f(d_i \in D_j) \to \max$. Результатом поиска служит подмножество документов, которым приписана наибольшая оценка важности f-го свойства. Обобщением этого варианта является векторный поиск, учитывающий оценки нескольких свойств.

4. Комбинированный поиск: найти $(D_j \subseteq D) \mid r(z, d_i \in D_j) \leqslant \Delta$ & & $S_f(d_i \in D_j) \to \max$.

Интеллектуальные возможности ИПС в части функций информационного поиска обусловлены способами задания и вычисления r и S.

Эффективность информационного поиска документов, обеспечиваемая ИПС, оценивается по ***информационной полноте*** и ***информационному шуму.*** Названные показатели выражаются коэффициентами полноты $k_п$ и шума $k_ш$ соответственно. Коэффициенты $k_п$ и $k_ш$ принимают значения в интервале от 0 до 1. В некоторых источниках эти коэффициенты выражают в процентах.

Пусть ИПС предъявлен i-й запрос. Информационно-поисковая система содержит множество документов D_i, релевантных этому запросу. В результате поиска получено множество D_i^0. Возможны следующие варианты.

1. $D_i^0 = D_i$. Идеальный вариант: полнота максимальна ($k_п = 1$), а шум нулевой ($k_ш = 0$).

2. $D_i^0 \subset D_i$. Имеет место неполнота ($0 \leqslant k_п < 1$), а шум отсутствует ($k_ш = 0$).

3. $D_i^0 \supset D_i$. Неполнота исключается ($k_п = 1$), но есть шум ($0 < k_ш \leqslant 1$).

4. $D_i^0 \cap D_i = \varnothing$ & $D_i^0 \neq \varnothing$ & $D_i \neq \varnothing$. Худший вариант: нулевая полнота (ни один релевантный документ не найден; $k_п = 0$) и максимальный шум (все, что выделено, не соответствует запросу; $k_ш = 1$).

5. $D_i^0 \cap D_i \neq \varnothing$ & $D_i^0 \not\subset D_i$ & $D_i \not\subset D_i^0$ & $D_i^0 \neq D_i$. Имеют место и неполнота ($0 < k_п < 1$), и шум ($0 < k_ш < 1$).

Определим коэффициенты полноты и шума:

$$k_{\text{п}} = \lim_{k\to m}\frac{1}{k}\sum_{i=1}^{k}\frac{|D_i \cap D_i^0|}{|D_i|}; \quad (3.2)$$

$$k_{\text{ш}} = \lim_{k\to m}\frac{1}{k}\sum_{i=1}^{k}\frac{|D_i^0 \backslash D_i|}{|D_i^0|}, \quad (3.3)$$

где m — достаточно большое число, чтобы по теореме о больших числах обеспечить требуемую достоверность результата эксперимента по определению $k_{\text{п}}$ и $k_{\text{ш}}$.

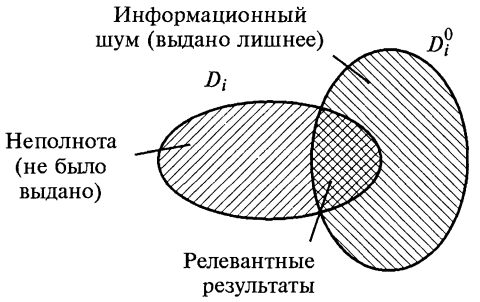


Рис. 3.6. Смысл коэффициентов полноты и шума

Смысл коэффициентов полноты и шума на теоретико-множественном уровне иллюстрирует рис. 3.6. Анализируя этот рисунок, нетрудно заметить, что успешность поиска формально определяется степенью совпадения множеств D_i и D_i^0 (см. вар. 1: в идеале, при $D_i^0 = D_i$ выборка содержит все релевантные документы и ни одного не релевантного). Это дает возможность ввести оценку эффективности информационного поиска E_1 на основе мощностей множеств D_i, D_i^0 и $D_i^0 \cap D_i$:

$$E_1 = 2\lim_{k\to m}\frac{1}{k}\sum_{i=1}^{k}\frac{|D_i \cap D_i^0|}{|D_i| + |D_i^0|}. \quad (3.4)$$

Эффективность информационного поиска E_1 выражается через коэффициенты $k_{\text{ш}}$ и $k_{\text{п}}$, что позволяет рассматривать ее в качестве *интегрального показателя эффективности информационного поиска ИПС*. В литературе в функции $E_1(k_{\text{ш}}, k_{\text{п}})$ вместо $k_{\text{ш}}$ принято использовать обратный ему показатель — *коэффициент точности* $k_{\text{т}}$:

$$k_{\text{т}} = 1 - k_{\text{ш}} = \lim_{k\to m}\frac{1}{k}\sum_{i=1}^{k}\frac{|D_i \cap D_i^0|}{|D_i^0|}. \quad (3.5)$$

Таким образом, запишем данную функцию в виде:

$$E_1 = \frac{2k_{\text{т}}k_{\text{п}}}{k_{\text{т}} + k_{\text{п}}}. \quad (3.6)$$

В теории информационного поиска предложен *обобщенный комплексный показатель эффективности* E_β (мера Ван Ризбергена), позволяющий учитывать предпочтение, отдаваемое пользователем ИПС точности или полноте:

$$E_\beta = \frac{(\beta^2 + 1)k_{\text{т}}k_{\text{п}}}{\beta^2 k_{\text{т}} + k_{\text{п}}}, \quad (3.7)$$

где β — параметр, отражающий предпочтение пользователя ИПС одному из показателей эффективности, входящих в E_β (точности, полноте), над другим.

При $\beta = 1$ точность и полнота одинаково важны. На интервале $\beta \in [0; 1[$ приоритет имеет точность, а на интервале $\beta \in]1; \infty[$ — полнота.

Отметим важные частные случаи:

- $E_{\beta=1} = E_1$ (т. е. E_1 выводится из (3.7));
- $E_{\beta=0} = k_т$ (значима только точность, полнота не важна);
- $E_{\beta\to\infty} = k_п$ (значима только полнота, точность не важна).

Сравнение документальных, фактографических и гипертекстовых ИПС по ряду показателей представлено в табл. 3.3.

Таблица 3.3

Характеристика ИПС	Виды ИПС		
	Документальные	Фактографические	Гипертекстовые
Полнота и шум	$k_{п\ max} = 0{,}5$ $k_{ш\ max} = 1$	$k_{п\ max} = 1$ $k_{ш\ max} = 0$	$k_{п\ max} = 0{,}9 \div 1{,}0$ $k_{ш\ max} = 0{,}1 \div 0{,}2$
Систематизирующая информация	Поисковые образы документов, метаданные	Значения атрибутов объектов ПрО	Гипертекстовое представление документов, метаданные
Тип поискового аппарата	Информационно-поисковые языки с развитой грамматикой	Языки реляционного типа	Гипертекстовый тезаурус
Трудоемкость подготовки информационного массива	Требуется специальная лингвистическая подготовка сотрудника	Требуется высокая квалификация сотрудника	Относительно несложная подготовка по типам семантических связей
Структуры данных	Прямые и инверсные списки	Иерархические или реляционные структуры	Семантическая сеть: вершины – понятия, ребра – отношения
Математический характер критериев поиска	Логические и алгебраические выражения	Логические и алгебраические выражения	Семантические признаки
Тип собственного языка системы	Специальные информационные языки (например, Сетка-5)	Специальные языки (SQL, QBE)	ОЕЯ ПрО

3.1.6. Методы извлечения знаний для построения гипертекста

Рассмотрим классификацию методов извлечения знаний для построения ГТ [33, 39].

Существуют два класса *источников знаний:*

- эксперты (специалисты в ПрО, для которой формируется ГТ);
- текстовые документы на ЕЯ.

Соответственно методы извлечения знаний подразделяются на два больших класса:

1) приобретение знаний от экспертов (коммуникативные методы);

2) обработка документов (текстологические методы).

Первый класс методов извлечения знаний имеет следующую структуру.

1.1. Пассивные методы.

1.1.1. Наблюдение за работой эксперта. Инженер по знаниям наблюдает за экспертом, который выполняет или имитирует выполнение своей профессиональной деятельности. Эксперт может комментировать совершаемые им действия. В ходе процесса ведется протокол (на бумаге, аудио- или видеоносителе).

1.1.2. Запись и анализ лекций.

1.1.3. Запись и анализ вербальных отчетов. Как и в методе 1.1.1, эксперт выполняет или имитирует выполнение своей профессиональной деятельности. Отличие заключается в том, что на каждом ее шаге он объясняет принимаемые им решения, рассуждая вслух (почему совершается именно это, а не иное действие; как было получено данное решение и т. п.). Вербальный отчет («мысли вслух») фиксируется на бумаге или аудионосителе и впоследствии анализируется инженером по знаниям.

1.2. Активные методы.

1.2.1. Работа с группой экспертов.

1.2.1.1. Метод «мозгового штурма». Этот метод является одним из наиболее известных и широко применяемых. Его цель — активизация творческого мышления за счет запрета критики высказываемых идей. Для проведения «мозгового штурма» формируется группа экспертов. Членам группы предлагается высказывать любые идеи, связанные с решением определенной проблемы. Выступления протоколируются. Обсуждение и критика идей исключаются. Последующий анализ и оценивание предложенных идей, как правило, выполняют эксперты, не участвовавшие в «мозговом штурме».

1.2.1.2. Метод «круглого стола». Метод заключается в организации обсуждения некоторой проблемы группой экспертов, наделенных равными правами. На первом этапе эксперты выступают по очереди, на втором проводится свободная дискуссия. Содержание обсуждения записывается на аудионоситель и впоследствии анализируется инженером по знаниям.

1.2.1.3. Ролевые игры. В рамках рассматриваемой проблемной ситуации каждому эксперту приписывается определенная роль (тип действующего лица в этой ситуации). Игра заключается в имитации совместной деятельности, направленной на разрешение проблемы.

1.2.2. Индивидуальная работа с экспертом.

1.2.2.1. Анкетирование.

1.2.2.2. Интервьюирование.

1.2.2.3. Свободный диалог. Суть свободного диалога – беседа инженера по знаниям с экспертом, для которой заранее не составляется план интервью или перечень вопросов.

1.2.2.4. Исследовательская игра с одним экспертом. В игре участвуют эксперт и инженер по знаниям. Последний может играть одну из ролей в рамках рассматриваемой проблемной ситуации.

Структура второго класса методов извлечения знаний приведена ниже.

2.1. Обработка текстов на ОЕЯ.

2.1.1. Анализ специализированной документации.

2.1.2. Анализ специализированных инструктивных и нормативных материалов (должностных и производственных инструкций, методик и др.).

2.2. Обработка текстов на ЕЯ.

2.2.1. Анализ учебной литературы.

2.2.2. Анализ научной и научно-практической литературы.

2.2.3. Анализ периодических изданий.

2.2.4. Анализ технической документации.

Технологии автоматизированной обработки текста будут рассмотрены в § 3.2.

3.1.7. Автоматизация построения гипертекста

Ручное формирование ГТ на основе объемного текстового материала — весьма трудоемкий процесс. Для его упрощения служат средства, позволяющие:

- автоматически определять позиции, в которых нужно устанавливать гиперссылки;
- автоматически выявлять связи между документами.

Среди российских программных продуктов можно отметить следующие средства автоматизации построения ГТ:

- авторскую систему HyperMethod (разработчик — компания «ГиперМетод»), включающую компонент HyperText Assistant, выполняющий автоматическую расстановку гиперссылок в формируемом электронном издании на основе системы настраиваемых правил;

• комплексную систему анализа текстов TextAnalyst (разработчик — научно-производственный инновационный центр «Микросистемы»).

Автоматизация расстановки гиперссылок в HyperText Assistant основана на использовании базы правил. Каждое правило содержит условие выделения фрагмента текста, от которого должна быть установлена гиперссылка, и идентификатор целевого кадра, на который эта ссылка должна указывать. Например, правило, представленное в табл. 3.4, предписывает, что все вхождения в текст слов «гиперссылка», «гиперсвязь» и «ссылка» должны быть оформлены как гиперссылки, ведущие в кадр «Определение гиперссылки».

Таблица 3.4

Условие выделения фрагмента текста	Идентификатор целевого кадра
«гиперссылка» ИЛИ «гиперсвязь» ИЛИ «ссылка»	«Определение гиперссылки»

Разработчик ГТ может создавать, изменять и удалять правила. Каждому правилу приписан признак активности, позволяющий запретить его применение, не исключая из базы правил.

HyperText Assistant автоматически выделяет фрагменты текста, удовлетворяющие условиям активных правил, и преобразует их в гиперссылки. За человеком остается принятие решения: устанавливать гиперссылку или нет. В пределе такое средство перерастает в специализированную ЭС: база правил учитывает специфику языка, на котором представлен текст, блок объяснения обосновывает выбор фрагмента для реализации гиперссылки, блок вывода может опираться на средства синтаксического и семантического анализа текста.

Характеристика TextAnalyst приведена в § 3.6.

3.1.8. Место гипертекстовой информационной технологии среди технологий искусственного интеллекта

Основоположником гипертекстового подхода принято считать Ванневара Буша [45]. Им был предложен проект MEMEX (Memory Extender), в рамках которого предполагалось создать автоматизированную систему доступа к большим слабоструктурированным информационным массивам, обеспечивающую быстрый просмотр хранимых сведений путем перемещения по заранее определенным связям между информационными единицами. Сам термин ГТ ввел Тед Нельсон [40], под руководством которого была создана первая гипертекстовая система Xanadu. Первые коммерческие ги-

пертекстовые системы (Guide, Hypercard) появились в середине 80-годов XX века. Тогда началось широкое проникновение ГИТ во все сферы информационной деятельности.

По мнению Теда Нельсона основные преимущества ГТ состоят в том, что читатель может не просто выбирать ту или иную траекторию изучения текста, но и создавать новый текст на основе содержащейся в ГТ информации [46]. Главное различие между традиционными и гипертекстовыми ИПС заключается в том, что традиционные ИПС обычно формируются на основе структурированных данных, в то время как в ГИПС может быть представлена слабо формализованная совокупность текстов, иллюстраций, аудио и видеодокументов и т. д.

Различие между ГТ и традиционной ИС подобно различию между БД и БЗ [41]. Из базы данных можно извлечь данные, перенести в другую БД, и они при этом не потеряют своих свойств. В свою очередь, элементы знаний не могут быть произвольно перенесены из одной БЗ в другую БЗ, поскольку их интерпретация в общем случае зависит от всего содержимого БЗ. Аналогично, смысл и ценность элемента ГТ зависит от содержания связанных с ним прочих элементов ГТ, а также от возможностей читателя увидеть и эксплицировать новые связи между этим элементом и остальными.

Человеку свойственны две стратегии обработки информации. Левое полушарие мозга отвечает за формально-логическую сторону мышления (создание концептуального пространства), а правое — за образную (создание перцептивного пространства). Исходя из этих представлений пользователей ГТ можно условно разделить на три класса. В первый входят люди с доминирующим левым полушарием. Они склонны к логическому типу мышления, использующему наиболее «сильные», логически обусловленные связи, отраженные в тексте. Ко второму классу относятся люди с преобладающим правополушарным мышлением, которые действуют, руководствуясь интуицией. Они могут не учесть «сильные» связи. В то же время для них характерна возможность выявления «слабых», неочевидных связей, что нередко приводит к формированию новых, неожиданных идей. Третий класс включает людей, у которых работа обоих полушарий уравновешена. Гипертекстовое представление информации соответствует ассоциативному характеру мышления человека, способствует осознанию целей читателя, обеспечивает высокую степень свободы его мышления.

Гипертекстовая информационная технология базируется на основных парадигмах ИИ: использовании БЗ, логическом выводе и общении с пользователем на ОЕЯ. Рис. 3.7 иллюстрирует соотношение структур гипертекстовой и экспертной систем. На рисунке видно, что данные системы имеют аналогичные блоки пользовательского интерфейса, БЗ, БД и приобретения

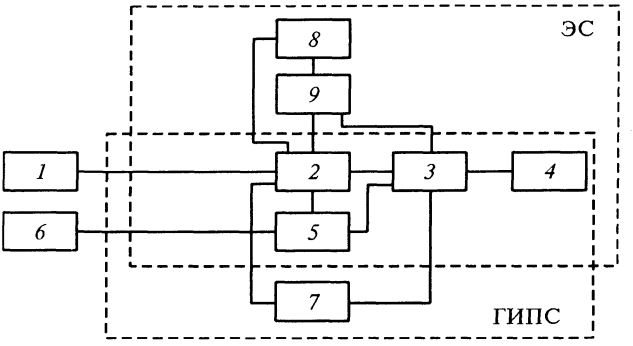


Рис. 3.7. Соотношение структур гипертекстовой и экспертной систем:

1 — пользователь; *2* — блок пользовательского интерфейса; *3* — БЗ; *4* — БД; *5* — блок приобретения знаний; *6* — тексты документов (для ГИПС) и знания экспертов (для ЭС); *7* — блок организации навигации и поиска по данным и знаниям; *8* — подсистема объяснения; *9* — блок логического вывода

знаний. Экспертную систему отличает наличие блоков объяснения и логического вывода с базой правил вывода. В свою очередь, для гипертекстовой ИС характерно наличие блока фиксации навигации при поиске, который в какой-то степени является прототипом блока объяснения в ЭС.

Гипертекст расширяет возможности человека, связанные с поиском и обработкой информации, за счет установления ассоциаций, построения обобщений, формирования целостного представления о содержании документа и т. д.

В настоящее время существует тенденция интеграции гипертекстовых ИС со специализированными пакетами прикладных программ. При этом возникают гибридные ИС, предназначенные для решения различных классов трудноформализуемых задач. В ряде источников гипертекстовые ИС рассматриваются как представители систем, доставляющих знания [29].

Основные выводы

1. ГИТ является одной из основных технологий ИИ, доведенной до широкого практического применения. Лучшей демонстрацией возможностей ГИТ служит WWW. Средства для построения ГТ — обязательный компонент инструментария специалиста по НИТ.

2. Текст, ГТ и гипермедиа являются обобщенными моделями представления знаний. Гипертекстовая информационная технология позволяет формировать интегрированные модели представления ПрО для решения трудноформализуемых задач.

3. Фиксация в ГТ множества траекторий изучения документа позволяет адаптировать его к интересам читателей, имеющих разные уровни профессиональной подготовки.

4. К чертам естественного интеллекта, отражаемым в ГИТ, относятся ассоциативный характер мышления, а также умение выделять семантические связи в тексте и формировать целостное представление о его содержании.

5. Отражение в ГИПС семантической структуры документов расширяет возможности и повышает эффективность информационного поиска. Наряду с особыми методами в ГИПС реализуются поисковые процедуры, используемые в документальных ИПС.

6. Существует тенденция интеграции ГИТ с другими технологиями обработки текстов на ЕЯ и ЭС нового поколения.

Вопросы для самопроверки

1. Какие идеи лежат в основе ГТ, и какие новые свойства ИС они обеспечивают?
2. Как ГИТ используется в Internet?
3. Что такое HTTP и HTML?
4. Назовите основные области применения ГИТ. Что обеспечивает ГИТ в каждой из них?
5. Охарактеризуйте формализованную модель ГТ.
6. Чем различаются ГТ, гиперграфика и гипермедиа?
7. Опишите условно-типовую модель ГТ.
8. Дайте определение понятия «тезаурус».
9. Что включается в список главных тем ГТ?
10. Для чего предназначен указатель?
11. Перечислите основные виды указателей.
12. Какой компонент условно-типовой модели ГТ представляет семантические отношения ИСС?
13. Охарактеризуйте известные Вам инструментальные средства для создания ГТ.
14. Какие способы доступа к информации предусматривают в гипертекстовых справочниках?
15. Назовите основные классы ИПС.
16. Что такое поисковый образ, поисковое предписание, дескриптор, индексирование, индекс?
17. Для чего предназначен критерий смыслового соответствия?
18. Какие существуют методы информационного поиска в ИПС?
19. Какое место занимает поиск по метаданным среди методов поиска, используемых в документальных и фактографических ИПС?
20. В чем состоят особенности полнотекстового поиска по сравнению с традиционным поиском по дескрипторам?
21. Назовите показатели эффективности информационного поиска документов в ИПС. Поясните их смысл.
22. Сравните основные характеристики ИПС разных классов.

23. Каковы причины неполноты и информационного шума при поиске по дескрипторам? Может ли такой вид поиска быть исчерпывающе полным и точным?
24. Какие методы извлечения знаний используются при построении ГТ?
25. Какое место занимает ГИТ среди технологий ИИ? Почему ГИТ относят к ИИ?
26. Каковы перспективы развития ГИТ?

3.2. Автоматизированное извлечение знаний из текста*

Знания — это способность действовать в соответствии с контекстом.

Д-р Карл-Эрик Свейби

Автоматизированное извлечение знаний из текста становится одной из центральных задач ИИ. Этому способствует исключительно быстрое развитие Internet и электронных библиотек, в которых знания представляются, в основном, в текстовом виде.

Ограниченные естественные языки играют роль *языков деловой прозы* или *языков специалистов в ПрО.* Исследования показали, что ОЕЯ, к сожалению, присущи большинство трудностей ЕЯ [52]. Поэтому использование ОЕЯ вместо ЕЯ не обеспечивает существенного упрощения обработки текстов.

Создание методов автоматизированного извлечения знаний из текста сопряжено с фундаментальной проблемой ИИ, связанной с пониманием текста на ЕЯ.

3.2.1. Проблема понимания текста на естественном языке

Общепризнанная схема анализа монологического текста на ЕЯ изображена на рис. 3.8, заимствованном из [58].

Предредактор выделяет в исходном тексте слова и фразы и проверяет выполнение принятых ограничений. Обычно недопустимыми являются сложноподчиненные предложения, включающие рекурсивно вложенные определительные предложения.

Блок морфологического анализа выделяет в словах неизменные части (основы) и приписывает словам ряд грамматических характеристик (часть речи, род, число, падеж, склонение, вид и т. п.).

Программная реализация предредактора и блока морфологического анализа не вызывает трудностей за исключением отмеченных выше ограни-

* Содержание параграфа соответствует направлениям исследований в области ИИ 1.4.1, 2.2.2 и 4.2.

Рис. 3.8. Схема анализа монологического текста на ЕЯ

чений для предредакторов и немногих случаев морфологической омонимии. Последняя проблема разрешается в *блоке синтаксического анализа.* Он строит дерево синтаксического разбора, используя базу синтаксических правил. Реализация этого блока также не вызывает трудностей.

Цель *семантического анализа* состоит в определении для каждого слова и фразы в целом некоторых смысловых характеристик. Сложности возникают из-за семантической неоднозначности. Для ее снятия используются тезаурусные статьи, связанные друг с другом в рамках семантической сети. Анализ отношений в ней позволяет получить информацию, в явном виде отсутствующую во фразе, но без которой адекватное понимание фразы невозможно. Трудности реализации этого этапа связаны с большим размером требуемых семантических сетей и многовариантностью анализа.

После семантического анализа выполняется *перевод анализируемого текста во внутреннее представление.* Обычно для этого также используются семантические сети. Содержание текста на ЕЯ отображается во фрагмент семантической сети, связанный дугами соответствующего типа с той семантической сетью, которая уже хранится в системе. Воплощение этого этапа не вызывает трудностей.

Внутреннее представление служит основой для реализации феномена понимания естественно-языкового текста. Именно с этим процессом связа-

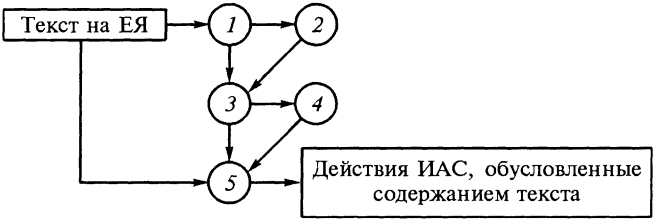


Рис. 3.9. Взаимосвязь уровней понимания естественно-языковых текстов:
1—5 — уровни

ны основные теоретические проблемы. Во многом они обусловлены отсутствием точного определения термина «понимание».

Понимание — многоуровневый процесс. На первом, простейшем уровне все сведения о содержании рассматриваемого текста извлекаются в результате его анализа без привлечения дополнительных знаний, известных системе. На втором уровне с помощью процедур логического пополнения информации осуществляется доопределение временно́й, пространственной и причинно-следственной структур событий.

На третьем уровне к сформированному представлению содержания текста добавляется информация, релевантная этому содержанию и известная системе. На четвертом уровне к нему присоединяются сведения, извлеченные из БЗ и связанные с анализируемым текстом только отношениями ассоциации.

На пятом уровне понимания из анализируемого текста извлекается его прагматическое содержание. При этом система выполняет все обусловленные им действия, например, решает задачу, для которой есть готовая или генерируемая программа, а в исходном тексте выражены исходные данные для нее. Ясно, что наибольший практический интерес представляют системы, реализующие пятый уровень понимания, и именно они называются интеллектуальными. Вместо модели «текст — смысл — текст» такие системы реализуют модель «текст — действительность — текст».

Взаимосвязь уровней понимания естественно-языковых текстов показана на рис. 3.9. Уровни не образуют строгой иерархии, и порядок их прохождения может быть разным.

До реализации в полном объеме такой схемы понимания еще далеко.

3.2.2. Компьютерные методы поиска в тексте

При реализации НИТ в ИИ принято начинать с анализа особенностей решения аналогичных задач человеком.

Методы поиска в тексте, используемые человеком, представлены следующими формами:

- поиск «сверху» (по оглавлению с аннотациями глав и, возможно, менее крупных разделов);
- поиск «снизу» (с помощью различных указателей);
- поиск с помощью гипертекстовых связей (перекрестных ссылок);
- полнотекстовый поиск путем просмотра всего текста.

Компьютерные методы поиска реализуются в ИПС, БД, БЗ и поисковых машинах Internet.

В информационно-поисковых системах применяются следующие методы поиска:

1) индексирование текстов и поиск по ключевым словам (по индексу);

2) поиск, включающий морфологический разбор и отождествление различных грамматических форм слов;

3) поиск с ранжированием документов по степени релевантности запросу;

4) использование формальных поисковых языков;

5) комплексные методы.

В технологиях БД и БЗ наряду с перечисленными применяются следующие методы поиска:

- использование формальных языков запросов, позволяющих описывать условия совместного вхождения ключевых слов в документ (это направление представляют SQL-подобные языки);
- методы семантического анализа текста.

Средства автоматического извлечения знаний из текстовых ресурсов Internet реализуются в поисковых машинах. При этом различают:

1) методы итеративного поиска;

2) методы поиска по выборке;

3) методы, использующие каталоги (рубрикаторы и классификаторы, организующие множество документов в деревья или лес);

4) семантические методы поиска, использующие подходы ИИ.

Создание средств построения и выполнения запросов на ЕЯ является серьезной проблемой. К числу подобных поисковых механизмов, например, относятся:

- средства, реализующие поиск типа «где»;
- средства поиска особых контекстных явлений;
- средства выполнения фактографических запросов и др.

Среди интеллектуального поискового инструментария для WWW следует упомянуть утилиту Echo Search фирмы Iconovex Corp. Она написана на языке Java и работает на базе платформ Windows и Macintosh. Эта утилита избавляет пользователя от необходимости разбираться в результатах поиска: она анализирует содержимое web-серверов, найденное несколькими средствами поиска (используются шесть популярных поисковых машин), и затем

формирует описания исследованных узлов. Утилита Echo Search создает текстовые копии страниц, соответствующих ключу поиска, и записывает их на жесткий диск. Проводя лингвистический анализ содержимого этих страниц, Echo Search автоматически генерирует указатель, кратко представляющий содержание каждой страницы.

Internet растет взрывообразно, поэтому вероятность наличия в нем необходимой информации постоянно увеличивается. Для поиска информации в Internet служат различные классы поисковых средств:

- каталоги (directories);
- подборки ссылок (bookmarks);
- поисковые машины (search engines);
- БД адресов электронной почты (email addresses databases);
- средства поиска в архивах Gopher (Gopher archives);
- системы поиска файлов (FTP search);
- системы поиска новостей (usenet news).

Каталог ресурсов Internet — постоянно обновляемая и пополняемая система ссылок на ресурсы, распределенные по иерархической структуре категорий. На верхнем уровне каталога представлены самые общие категории (рубрики), например, «наука», «бизнес», «развлечения» и т. д. На нижележащих уровнях эти рубрики декомпозируются на подчиненные рубрики, имеющие более частный характер. Например, верхний уровень каталога mail.ru содержит рубрики: «Автомобили», «Бизнес и финансы», «Вокруг света», «Государство российское», «Домашний очаг», «Интернет», «Компьютеры», «Культура/Искусство», «Медицина и здоровье», «Непознанное», «Образование/Наука», «Отдых», «Работа и заработок», «СМИ», «Спорт», «Справки», «Товары и услуги», «Юмор».

На нижнем уровне каталога указываются ссылки на конкретные ресурсы Internet (сайты и web-страницы), снабженные краткими описаниями их содержимого.

Каталоги ресурсов Internet незаменимы, когда человек имеет недостаточно точное представление о цели поиска. Некоторые из них позволяют выполнять поиск по ключевым словам в кратком описании содержимого ресурсов.

Каталоги облегчают поиск за счет упорядоченности ссылок на ресурсы. Все интеллектуальные функции остаются за человеком. То же можно сказать о *подборках ссылок* на информационные ресурсы Internet. Такие подборки представляют собой отсортированные по темам адреса ресурсов.

Формирование и актуализация каталогов и подборок ссылок выполняются вручную персоналом соответствующих ИС. Подобная работа требует высокой квалификации и достаточно трудоемка.

Ниже перечислены некоторые универсальные каталоги ресурсов Internet:

- Yahoo! (http://www.yahoo.com);
- MSN (http://search.msn.com);
- AOL (http://search.aol.com);
- About (http://www.about.com);
- Search (http://www.search.com);
- Яndex (http://www.yandex.ru);
- Rambler (http://www.rambler.ru);
- Апорт (http://www.aport.ru);
- Город-ОК! (http://link.cid.ru);
- Пингвин (http://www.able.ru) и др.

Наряду с универсальными существуют и специализированные каталоги, систематизирующие сведения о ресурсах Internet, имеющих определенную тематическую направленность. Например:

- каталог общественных ресурсов Интернет (некоммерческих и общественных организаций, средств массовой информации, электронных библиотек и изданий, БД, ИС, Internet-проектов и т. д.; http://www.ngo.ru);
- каталог вузов России (http://www.5ballov.ru/universities);
- каталог образовательных сайтов (http://www.allbest.ru).

Поисковые машины (или *поисковые системы*) позволяют находить ресурсы Internet непосредственно по их текстовому содержимому. Функционирование поисковой машины включает два базовых процесса: 1) индексирование ресурсов Internet (автоматическое построение и обновление индекса); 2) поиск по индексу по запросам пользователей.

В Международном каталоге поисковых машин (Search Engine Colossus — http://www.searchenginecolossus.com) зарегистрировано свыше 2300 систем из 232 стран. По данным этого каталога более 80 % пользователей Internet находят информационные ресурсы с помощью поисковых машин, 57 % пользователей ежедневно применяют поисковые машины, каждый день выполняется до 450 млн поисковых запросов, поисковые машины служат источником сведений для 55 % всех покупок в on-line.

К наиболее известным поисковым машинам относятся:

- AltaVista (http://www.altavista.com);
- Excite (http://www.excite.com);
- HotBot (http://www.hotbot.com);
- Lycos (http://www.lycos.com);
- Yahoo! (http://www.yahoo.com);
- AOL (http://search.aol.com);
- MSN (http://search.msn.com);
- Infoseek (http://infoseek.go.com);
- Google (http://www.google.com.ru);
- About (http://www.about.com);
- Search (http://www.search.com);

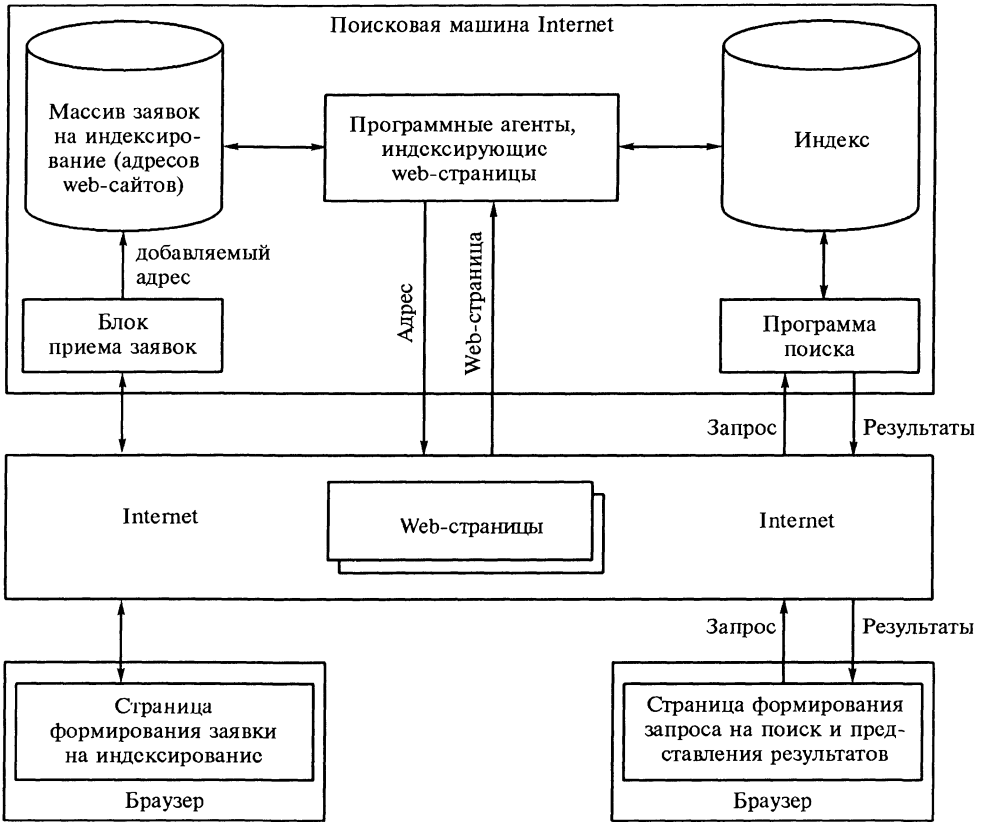


Рис. 3.10. Упрощенная структура типовой поисковой машины

- Dogpile (http://www.dogpile.com);
- Яndex (http://www.yandex.ru);
- Rambler (http://www.rambler.ru);
- Апорт (http://www.aport.ru);
- Rundex (http://www.rundex.ru).

Следует отметить, что многие поисковые машины включают и каталоги ресурсов Internet.

Упрощенная структура типовой поисковой машины показана на рис. 3.10. Ее главными компонентами являются:

- программный агент, «перемещающийся» по сети и индексирующий ресурсы (web-страницы);
- БД (индекс), содержащая информацию, собираемую агентом;
- программа поиска, применяемая пользователями для поиска информации в БД.

На этапе индексирования поисковые машины реализуют следующий *примерный алгоритм работы.*

1. Адреса web-узлов, включаемые в обрабатываемую область, определяются по гиперссылкам, ведущим из страниц данного web-узла. При этом используются различные модификации волнового алгоритма (например, с вычислением профилей узлов).

2. Агент либо переходит к индексированию очередного web-узла из сформированного списка, либо выполняет так называемое зеркалирование (дублирование) его содержимого на свой web-узел.

3. Производится собственно индексирование. Оно может быть полнотекстовым (обрабатывается весь текст) и неполнотекстовым (обрабатываются наиболее значимые части текста: заголовки, названия, ключевые поля, начальные слова разделов и т. д.).

4. Полученные данные о ключевых словах добавляются в БД.

5. Если был сделан зеркальный дубль, он стирается.

6. Пункты 2—5 повторяются для каждого адреса, полученного в п. 1.

Изложенный алгоритм соответствует некоторой канонической структуре поисковой машины. Конкретные их реализации различаются по многим параметрам:

- поддержке простого и сложного поиска;
- учету различий строчных и прописных символов;
- возможности поиска по частям слов и словосочетаниям;
- поддержке обработки запросов, содержащих логические операторы И, ИЛИ, НЕ;
- использованию специальных языков поиска информации, значительно сокращающих его время (к сожалению, такие языки не стандартизованы, поэтому в разных поисковых машинах реализуются разные поисковые языки).

Применение поисковых машин для поиска в Internet эффективно, если пользователь представляет, какие ключевые слова характеризуют нужные ресурсы.

Дополнительные возможности предоставляет режим расширенного поиска, в котором можно задавать правила поиска. Часто это значительно увеличивает вероятность нахождения требуемой информации.

Агент – самый интеллектуальный из компонентов поисковой машины. Он обладает автономностью, имеет блоки навигации, управляющие «перемещением» по сети, и механизмы индексации, основанные на некоторой базе правил. Агенты реализуются как простые программные системы, запрашивающие информацию с узлов Internet. Физически по сети агенты не перемещаются. Они индексируют полученные страницы и заносят результаты в БД.

Поисковые механизмы отличаются разнообразием. Некоторые агенты следуют по каждой ссылке на каждой найденной странице и затем, в свою очередь, исследуют каждую ссылку на новой странице и т. д. Как правило, агенты игнорируют ссылки к графическим и мультимедийным файлам, файлам с данными (например, архивам), БД и др. Ряд агентов просматривают страницы с учетом их популярности.

Одной из проблем является реализация алгоритма перемещения (навигации) по сети. Учитывая, что большинство web-серверов организовано иерархически, перемещение вширь по ссылкам от исходной вершины при ограниченной глубине вложенности с большей вероятностью приводит к нахождению документов с высоким уровнем релевантности, чем при перемещении в глубину. Поскольку это подтверждается статистикой работы поисковых машин, данный метод (сначала вширь, затем вглубь) принят как предпочтительный для индексирования web-ресурсов.

Разновидностями агентов являются кроулеры, «роботы» и «пауки». Кроулеры (crawlers) просматривают заголовки страниц и возвращают поисковой машине только первую найденную ссылку. «Роботы» проходят по ссылкам различной глубины и вложенности. «Пауки» (spiders) сообщают о содержании найденного документа, индексируют его и пересылают извлеченную информацию в БД поисковой машины.

Системой правил для всего этого сообщества автономных программ управляют администраторы поисковых машин. Они же устанавливают параметры алгоритмов определения степени релевантности документа и запроса. Обычно в этих алгоритмах учитываются:

- количество слов запроса в текстовом содержимом документа (т. е. в HTML-коде);
- теги, в которых эти слова встречаются;
- местоположение искомых слов в документе;
- удельный вес слов, относительно которых определяется релевантность, в общем количестве слов документа;
- время существования web-сайта;
- индекс цитируемости web-сайта и др.

Средством полнотекстового поиска, ориентированным на локальные информационные массивы и корпоративные сети, служит интеллектуальная система «Следопыт» (разработчик — компания «МедиаЛингва»; http://www.sledopyt.ru). Она позволяет сформировать дерево областей поиска в виде иерархии папок с документами. Для каждой области может быть построен отдельный индекс.

Система обрабатывает документы в форматах DOC, DOT, RTF, XLS, PPT, TXT, HTML и PDF, документы данных форматов в ZIP-архивах, сообщения электронной почты Microsoft Outlook, а также вложения в эти сооб-

щения в перечисленных форматах (в том числе в ZIP-архивах). При индексировании учитывается морфология русского и английского языков. Предусмотрена возможность автоматического обновления индексов.

«Следопыт» ведет поиск как по содержимому, так и по атрибутам документов. Запрос представляется в виде фразы на ЕЯ либо выражается на формальном языке с использованием логических операторов. В запрос могут одновременно входить термины русского и английского языков.

Другие примеры реализации функций интеллектуального поиска в корпоративных ИС описаны в § 3.6 и § 7.3.

Основные выводы

1. Возможности поисковых машин Internet в целом аналогичны поисковым возможностям ГИПС. Отличие заключается в колоссальных размерах сети и пространственной распределенности ее узлов, а также параллельности процессов поиска. Основа функционирования этих систем одна и та же: гипертекстовые связи, ключевые слова, критерии смыслового соответствия, активная роль человека в управлении их формой и в оценке релевантности результатов поиска запросу.

2. Поисковые системы, использующие базы правил, имеют тенденцию к перерастанию в специализированные ЭС.

3. Фундаментальной проблемой является создание методов понимания текста. Очевидно, что ее решение обеспечило бы основу для реализации эффективных средств извлечения знаний из естественно-языкового текста и полнотекстового поиска.

4. К сожалению, еще длительное время методы автоматического индексирования документов будут давать худшие по качеству результаты по сравнению с их индексированием авторами. Для повышения эффективности и качества индексирования к документам добавляют метаданные, представляющие общую характеристику их семантики. В частности, язык HTML позволяет вводить в документ ограниченный набор метаданных. Более эффективные решения связаны с использованием языка XML. Об этом пойдет речь в гл. 4.

Вопросы для самопроверки

1. В чем состоит проблема понимания текста на ЕЯ?
2. Каковы основные задачи морфологического и синтаксического анализа текста на ЕЯ?
3. Каковы основные задачи семантического анализа текста на ЕЯ?
4. Сколько уровней понимания естественно-языкового текста выделяют в ИАС? Охарактеризуйте эти уровни.

5. Назовите основные методы поиска в тексте, используемые в ИПС, технологиях БД и БЗ, Internet.
6. Что такое каталоги ресурсов Internet?
7. Какие основные компоненты включает типовая поисковая машина Internet и каков алгоритм ее работы?
8. Какие факторы могут учитываться поисковой машиной Internet при определении степени релевантности документа и запроса?

3.3. Автоматическое реферирование и аннотирование*

Всякое слово уже обобщает.

И.Н. Горелов

Рефератом называют:

- доклад на определенную тему, включающий обзор соответствующих литературных и других источников;
- изложение содержания научной работы, книги и т. д.

Далее будем опираться на вторую трактовку.

Под *аннотацией* понимается краткая характеристика произведения печати или рукописи. Обычно аннотация приводится после библиографического описания источника.

Аннотацию от реферата отличают:

- существенно меньший объем;
- обязательная констатация назначения аннотируемого произведения (для каких категорий читателей оно предназначено).

Автоматические реферирование и аннотирование получили значительную актуальность в связи с развитием Internet и каталогов информационных ресурсов. Для экономии времени поиска пользователям предлагаются каталоги аннотаций и рефератов источников.

Формирование рефератов и аннотаций вручную требует колоссальных человеческих ресурсов, поэтому и возникла задача создания методов автоматического реферирования и аннотирования.

Автоматическое реферирование и аннотирование — одно из направлений компьютерной обработки естественно-языковых текстов**. И в этом качестве оно относится к фундаментальным технологиям ИИ.

Основные тенденции для данной области:

---

* Содержание параграфа соответствует направлению исследований в области ИИ 1.4.3.

** Системы, обрабатывающие тексты на ЕЯ, в зарубежной литературе называют NLP-системами (natural language processing).

- аннотированные каталоги перерастают в гипертекстовые (с их минусами и плюсами);
- на всех крупных сайтах Internet предусматривают оглавления (карта сайта — sitemap) и функции поиска по сайту;
- использование онтологических словарей-тезаурусов общего и специализированного назначения, а также методов ИИ.

Потребности в средствах автоматического реферирования и аннотирования испытывают: корпоративные системы документооборота; поисковые машины и каталоги ресурсов Internet; автоматизированные информационно-библиотечные системы; каналы вещания; службы рассылки новостей и др.

Методы автоматического реферирования и аннотирования подразделяются на поверхностные и глубинные.

Поверхностные методы базируются на «экстрагировании» текста, т. е. извлечении из него фрагментов, оцениваемых системой как важнейшие, и объединении их в реферат или аннотацию. Важность фрагментов определяется:

- по маркерам важности (оборотам типа «идея ... состоит в...», «главным результатом ... является...», «в заключении нужно сказать, что...» и т. д.);
- по количеству заданных в запросе ключевых слов, входящих во фрагмент, и др.

При объединении выделенных предложений в реферат или аннотацию учитываются их зависимости друг от друга (удаленность выделяемых мыслей). «Стыки» между предложениями (фрагментами) «сглаживаются».

Глубинные методы, развиваемые в настоящее время, базируются на применении тезаурусов и развитых механизмов синтаксического разбора текста.

К традиционным системам автоматического реферирования и аннотирования, реализующим поверхностные методы, можно отнести:

- Microsoft Word (начиная с версии 7 имеется функция автоматического реферирования);
- ОРФО 5.0 (разработчик — компания «Информатик»), включающую функцию автоматического аннотирования русских текстов;
- «Либретто» (разработчик — компания «МедиаЛингва»), обеспечивающую автоматическое реферирование и аннотирование русских и английских текстов (система встраивается в Word);
- пакет «МедиаЛингва Аннотатор SDK 1.0», служащий инструментарием для реализации функций автоматического реферирования и аннотирования в прикладных ИАС;
- поисковую систему «Следопыт», включающую средства автоматического реферирования и аннотирования документов;

- поисковую машину «Золотой Ключик» компании Textar, обеспечивающую составление рефератов и аннотаций;
- Intelligent Text Miner (IBM);
- Oracle Context;
- программные компоненты для разработки систем управления знаниями Inxight Summarizer фирмы Inxight Software, Inc.

Перечисленные средства обеспечивают выбор оригинальных фрагментов из исходных документов и соединение их в короткий текст.

Сделаем два замечания. Во-первых, источниками информации для рефератов и аннотаций могут служить не только тексты, но и видеозаписи, разнообразные табличные документы и т. д. Во-вторых, краткое изложение предполагает передачу основной мысли не обязательно теми же словами.

Основные *требования к реферату*:

- сжатие (объем реферата должен составлять от 5 до 30 % от объема исходного документа);
- возможность использования нескольких источников;
- выражение всех основных мыслей оригинала.

Выделяют *три вида рефератов*:

1) повествовательные, формирующие общее представление об источнике;

2) информационные, заменяющие источник (содержат основную или новую фактическую информацию);

3) критические (обзоры), отражающие не только суть источника, но и мнение о нем (т. е. содержащие дополнительные выводы, которых нет в оригинале).

Построение реферата человеком включает следующие этапы:

- анализ источника;
- выделение в источнике наиболее важных и информативных фрагментов;
- формирование выводов.

В теории автоматического реферирования различают три основных подхода [65]. Первый из них не предполагает опору на знания, связанные с текстом на ЕЯ. В системах такого типа применяется универсальная база правил, не зависящая от ПрО и языка текста. Второй подход предусматривает выделение различных уровней понимания текста, что требует использования наряду с универсальными правилами БЗ о ПрО и базы лингвистических правил, зависящих от языка. Третий подход является гибридным. Он сочетает лучшие стороны первых двух.

В *системах первого типа* (т. е. воплощающих первый подход) применяется *метод составления выдержек*. Он реализуется в два этапа. На первом проводится сопоставление текста и фразовых шаблонов, в результате

чего выделяются блоки наибольшей лексической и статистической релевантности. На втором — путем соединения выделенных фрагментов формируется итоговый документ.

Для реализации первого этапа используют *модель линейных весовых коэффициентов.* В соответствии с ней каждому блоку U текста оригинала автоматически (на основании определенных правил) приписываются весовые коэффициенты:

- k_1, зависящий от расположения блока U в оригинале;
- k_2, зависящий от частоты появления блока в оригинале;
- k_3, зависящий от частоты использования блока в ключевых предложениях;
- k_4, отражающий показатели статистической значимости блока.

Затем по значениям k_1, k_2, k_3 и k_4 и коэффициентам настройки программы реферирования α_1, α_2, α_3 и α_4 вычисляется коэффициент важности блока $B(U) = \alpha_1 k_1 + \alpha_2 k_2 + \alpha_3 k_3 + \alpha_4 k_4$. По коэффициентам важности выполняется отбор блоков в реферат.

Для вычисления каждого весового коэффициента используется своя группа правил. Для k_1 они учитывают расположение блока:

- во всем тексте или некотором разделе;
- в начале, середине или конце текста;
- во вводной части, заключении и т. д.

Для k_2 правила учитывают результаты автоматической индексации документа (например, соотношение между частотой появления термина в документе и в наборе документов).

Для k_3 учитывается наличие в блоке таких ключевых фраз и выражений, как «в заключение...», «в данной статье...», «согласно результатам анализа...», «отличный от...», «малозначащий...» и т. п.

Для k_4 правила учитывают вхождение термина в заголовки, колонтитулы, первый параграф текста, пользовательский профиль запроса и т. п.

Настройка с помощью коэффициентов α_1, α_2, α_3 и α_4 позволяет управлять степенью сжатия.

На рис. 3.11 изображена обобщенная архитектура системы автоматического реферирования первого типа.

Главное достоинство описанной модели линейных весовых коэффициентов заключается в простоте ее реализации, а главный недостаток связан с возможностью формирования бессвязных рефератов, не учитывающих контекст. Для его устранения вводится этап ручного редактирования результатов.

Схема автоматического определения критериев адекватного выбора фрагментов оригинала для реферата используется в системе Inxight Summarizer (рис. 3.12). Обучение (настройка) системы осуществляется на наборах

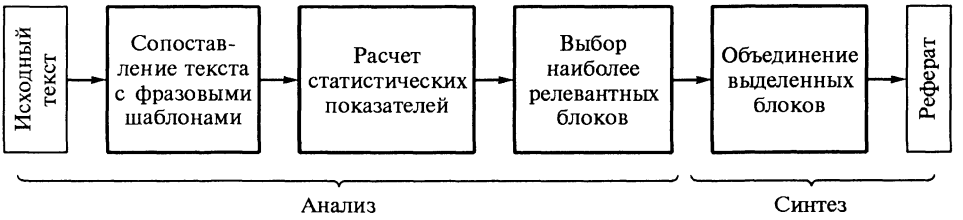


Рис. 3.11. Обобщенная архитектура системы автоматического реферирования

текстов и рефератов, составленных для них вручную при различных критериях сжатия.

Человеку, уловившему общий смысл информации, легче выделить главное и кратко изложить содержание. Это и обусловливает создание *реферирующих систем второго типа.* Для таких систем требуются:

- мощные вычислительные ресурсы;
- развитые грамматики и словари;
- развитые средства синтаксического разбора;
- средства генерации естественно-языковых конструкций;
- онтологические справочники.

В этих системах реализуются три подхода:

1) традиционный метод синтаксического разбора;
2) подход с опорой на понимание ЕЯ;
3) комбинированный подход.

В первом случае для построения деревьев разбора используется син-

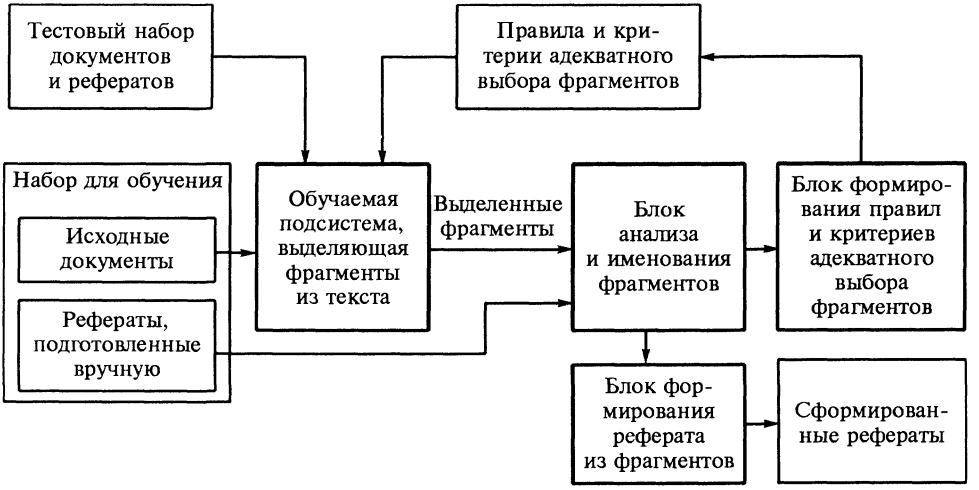


Рис. 3.12. Схема автоматического определения критериев адекватного выбора фрагментов

таксическая информация. Процедуры сжатия манипулируют деревьями с целью сокращения скобок, подчиненных предложений и т. д. При этом дерево разбора упрощается до «структурной выжимки».

При втором подходе в результате разбора строится не дерево, а семантическая сеть текста. Другими словами, в ходе разбора выделяются концептуальные репрезентативные структуры исходного текста. Из них удаляется избыточная информация: поверхностные суждения, концептуальные подграфы [66]. Далее выполняется агрегирование и обобщение информации: слияние некоторых концептуальных графов на базе правил. В результате получается «концептуальная выжимка».

Обобщенная схема для этих двух методов представлена на рис. 3.13.

Стадии синтеза реферата в обоих подходах почти совпадают (используется генератор текста).

Для функционирования подобных систем необходимы:

- исчерпывающие словари (тезаурусы) типа WordNet;
- онтологические справочники типа *Cyc* и Penman Upper Model;
- большие объемы тестовых файлов с текстами (например, The Wall Street Journal или Penn Treebank от Linguistic Data Consortium).

Интеллектуальные автоматизированные системы, обрабатывающие тексты на ЕЯ, требуют развитого лингвистического обеспечения (ЛО). В последнее десятилетие было развернуто множество проектов по его созданию. К числу наиболее интересных из них относится *WordNet — открытая справочная лексическая система, представляющая тезаурус английского языка.* Данный проект выполняется с начала 90-х годов в лаборатории когнитологии Принстонского университета (Cognitive Science Laboratory at Princeton University) под руководством проф. Дж. А. Миллера (George A. Miller)*.

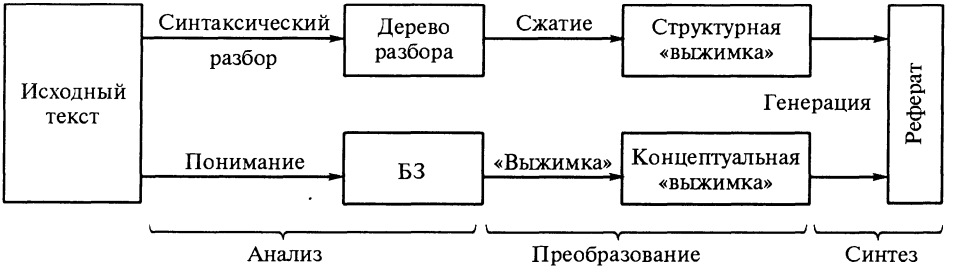


Рис. 3.13. Два основных подхода к формированию реферата в системах с опорой на знания

* http://www.cogsci.princeton.edu/~wn/index.shtml.

Система WordNet основана на психолингвистических теориях организации лексической памяти человека. Существительные, прилагательные, глаголы и наречия группируются в синонимические множества (synonym sets), называемые *синсетами* (synset). Каждый синсет представляет одно базовое лексическое понятие и состоит из множества слов и устойчивых словосочетаний, равнозначных в некотором контексте. Синсеты связаны отношениями различных типов.

Математической моделью тезауруса WordNet служит граф (X, R). Множество вершин в нем разбито на два непересекающихся подмножества: $X = X_1 \cup X_2$. Вершины из X_1 соответствуют словам и словосочетаниям, вершины из X_2 – их значениям (смыслам, толкованиям). Каждое значение соотносится с одной из частей речи: существительным, прилагательным, глаголом или наречием. В графовой интерпретации такая типизация может быть задана раскраской вершин из X_2.

Множество ребер также разбито на два непересекающихся подмножества: $R = R_1 \cup R_2$. Ребра из R_1 связывают слова со значениями, т.е. элементы из X_1 с элементами из X_2. Подобные ребра представляют отношения, входящие в множество $X_1 \times X_2$. Ребра, принадлежащие второму подмножеству, связывают слова со словами и значения со значениями, т. е. представляют отношения, входящие в множества $X_1 \times X_1$ и $X_2 \times X_2$.

Объединение слов и словосочетаний в синсеты (вершины из X_2) выражает отношение синонимии. Прочие тезаурусные отношения задают типы ребер из R_2. В WordNet выделено 14 базовых типов таких отношений (табл. 3.5). Помимо них используются обратные отношения для каждого из перечисленных типов.

Таблица 3.5

Тип отношения	Название отношения	Описание	Примеры прямых отношений
1	Антоним (antonym)	Отношение между словами, имеющими противоположные значения	large — small, большой — малый
2	«Имеет отношение к» (pertains to)	Отношение между прилагательным и другим словом (как правило, существительным, на основе которого оно образовано)	musical — music, музыкальный — музыка
3	Глагол, на основе которого образовано причастие (is a participle of)	Отношение между причастием (прилагательным, деепричастием) и глаголом, на основе которого оно образовано	studied — study, изучаемый — изучать

Продолжение табл. 3.5

Тип отношения	Название отношения	Описание	Примеры прямых отношений
4	Слово, на основе которого образовано наречие (is derived from)	Отношение между наречием и словом, на основе которого оно образовано	quickly — quick, быстро — быстрый
5	Действие, сопровождающее данное действие (entails)	Отношение между действиями (глаголами) *x* и *y*, фиксирующее, что *x* не может быть выполнено до тех пор, пока не выполняется или не совершено *y*	eat — chew, есть — жевать
6	Глагольная группа (verb group)	Отношение между синсетами, объединяющими глаголы и имеющими близкие значения	(agree, accord, consort, fit in, harmonize) — (agree, correspond, jibe, match, tally)
7	Атрибут (attribute)	Отношение между существительным, представляющим некоторый атрибут, и прилагательным, выражающим одно из значений этого атрибута	duration — long, продолжительность — долгая; duration — short, продолжительность — короткая
8	«Смотри также» (see also)	Общий случай ассоциативного отношения	slow — gradual, медленный — последовательный
9	Подобие (is similar to)	Отношение между прилагательным и другим словом, близким к нему по смыслу	auxiliary — subsidiary, вспомогательный — дополнительный
10	Род—вид, вид—род (is a type of, is a kind of)	Родовидовые отношения. Прямое отношение: род—вид; обратное отношение: вид–род	bird — parrot, птица — попугай
11	Целое—часть (has part), часть—целое (is a part of)	Отношения дезагрегации и агрегации. Прямое отношение (целое—часть): *x* включает *y* в качестве составной части (звена); обратное отношение (часть—целое): *y* является составной частью (звеном) *x*	computer — processor, компьютер — процессор

Окончание табл. 3.5

Тип отношения	Название отношения	Описание	Примеры прямых отношений
12	«Сделан из» (is made of, has substance), «служит субстанцией для» (is a substance of)	Субстанциональные отношения. Прямое отношение: *x* состоит из субстанции (компонента) *y*; обратное отношение: *y* входит в *x* в качестве субстанции (компонента)	air — oxygen, воздух — кислород; air — nitrogen, воздух — азот
13	Множество—элемент (has member), элемент—множество (is a member of)	Отношения принадлежности. Прямое отношение (множество—элемент): множество *x* включает элемент *y*; обратное отношение (элемент—множество): элемент *y* является членом множества *x*	regiment — battalion, полк — батальон
14	Цель—способ (is aim for), способ—цель (is one way to)	Отношения между глаголами, выражающими целевое действие и способ его выполнения. Прямое отношение: цель—способ, обратное отношение: способ—цель	separate — cut, отделять — резать

Поскольку отношения типов 1, 6, 8 и 9 являются симметричными, они совпадают со своими обратными отношениями. Для отношений типов 10–14 в табл. 3.5 приведены описания как прямых, так и обратных отношений.

На основе отношений базовых типов определяются прочие типы отношений, представляемых ребрами из R_2.

Web-интерфейс для работы с сетевой версией тезауруса доступен по адресу: http://www.cogsci.princeton.edu/cgi-bin/webwn. Локальную версию WordNet можно загрузить с сайта проекта. Она включает: информационную базу тезауруса; средство для поиска и просмотра тезауруса WordNet Browser; программные библиотеки и исходные тексты программ WordNet Browser; документацию, описывающую структуру и форматы файлов информационной базы, а также программную реализацию WordNet Browser.

WordNet является бесплатным, свободно распространяемым продуктом и может использоваться как в исходном, так и модифицированном виде в коммерческих приложениях. Информационная база WordNet 2.0 содержит 144309 слов и словосочетаний, 115424 значения и 203145 сочетаний слово–значение (ребер графа тезауруса, образующих подмножество R_1).

С проектом WordNet связан ряд проектов, направленных на расширение модели и программных средств WordNet, интеграцией компонентов WordNet в ИАС, созданием интерфейсов для доступа к информационной базе WordNet из приложений, основанных на различных технологиях и программных платформах, построением тезаурусов типа WordNet других ЕЯ*. В частности, разработаны WordNet-интерфейсы для технологий .NET (языка C#), COM, Java/WAP, языков C++, XML, Java, SQL, Lisp, ПРОЛОГ, Haskell, а также множество web-интерфейсов.

Интерактивный графический интерфейс для взаимодействия с тезаурусом WordNet реализован в системе *Visual Thesaurus***, разработанной фирмой Plumb Design***. Система формирует двухмерное или трехмерное представление графа тезауруса. Вершины из X_1 отображаются в виде слов или словосочетаний, а вершины из X_2 — в виде цветных окружностей. В свою очередь, ребра из R_1 обозначаются сплошными, а ребра из R_2 — пунктирными линиями.

Щелчок мыши на вершине-слове перемещает ее в центр окна. Вокруг нее располагаются вершины из X_2, представляющие значения данного слова (рис. 3.14). Аналогично, щелчок мыши на вершине-значении переводит эту вершину в центр окна (рис. 3.15). Вокруг нее отображаются вершины-слова, образующие соответствующий синсет. При подведении мыши к вершине-значению слова и словосочетания, входящие в синсет, выделяются цветом, а на экран выводится краткое определение значения. Указание мышью на ребро из R_2 вызывает вывод на экран типа представляемого им отношения.

Система содержит средства для поиска в тезаурусе и навигации по нему. Фильтр типов отношений позволяет запретить отображение ребер из R_2 определенных типов. При работе в трехмерном режиме можно вращать представляемый на экране фрагмент графа, выбирая наиболее наглядный вид.

Реализация Visual Thesaurus базируется на развиваемой Plumb Design технологии Thinkmap, предназначенной для создания динамических визуальных интерфейсов ИС, содержащих сложно взаимосвязанные данные. Thinkmap позволяет отображать как элементы данных, так и отношения между ними. В ней используется Java-технология и предусмотрены функции для доступа к различным источникам данных. С помощью Thinkmap могут разрабатываться модули визуализации для web-приложений и локальных систем.

---

* http://www.globalwordnet.org.

** http://www.visualthesaurus.com.

*** http://www.plumbdesign.com.

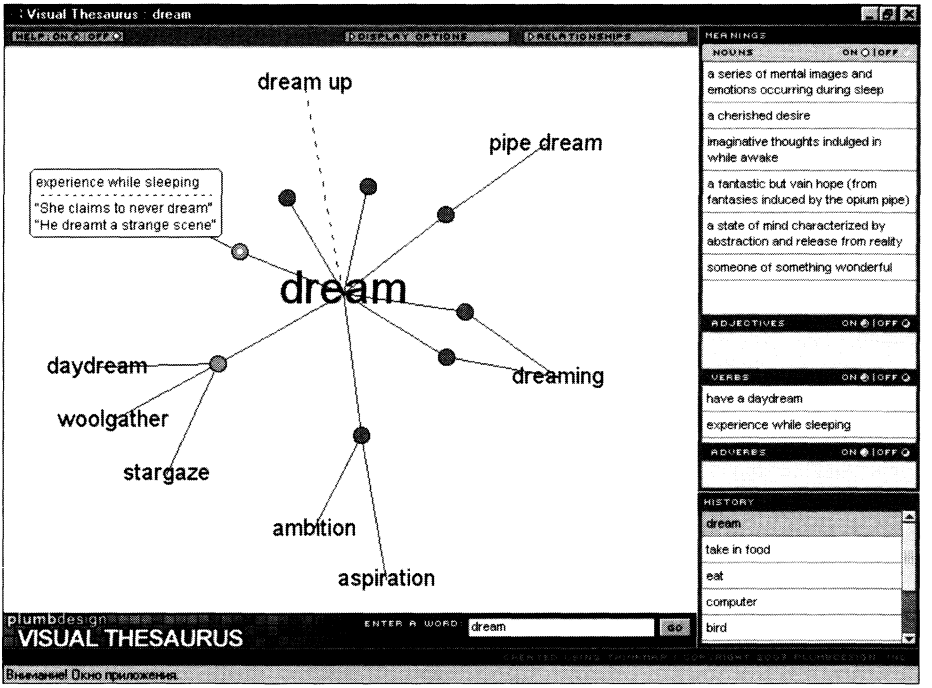


Рис. 3.14. Представление графа тезауруса в системе Visual Thesaurus (в центре окна располагается вершина-слово)

Еще одним продуктом, предоставляющим ЛО и средства для взаимодействия с ним, является пакет «МедиаЛингва Машинная словарная морфология SDK»*. Он служит инструментом для реализации функций морфологической обработки в прикладных ИАС. Пакет включает программные библиотеки, документацию и словари русского, английского, немецкого, итальянского, испанского и французского языков. Предусмотрена возможность подключения словарей других европейских языков.

Программные компоненты пакета поддерживают три главные функции:

- нормализацию (получение базовой грамматической формы слова для заданной словоформы);
- морфологический анализ (определение грамматических характеристик словоформы — род, число, падеж, время и т. д.);
- морфологический синтез (построение словоформы по базовой форме слова и грамматическим характеристикам).

Отметим следующие новые задачи, связанные с компьютерным реферированием.

* http://www.medialingua.ru.

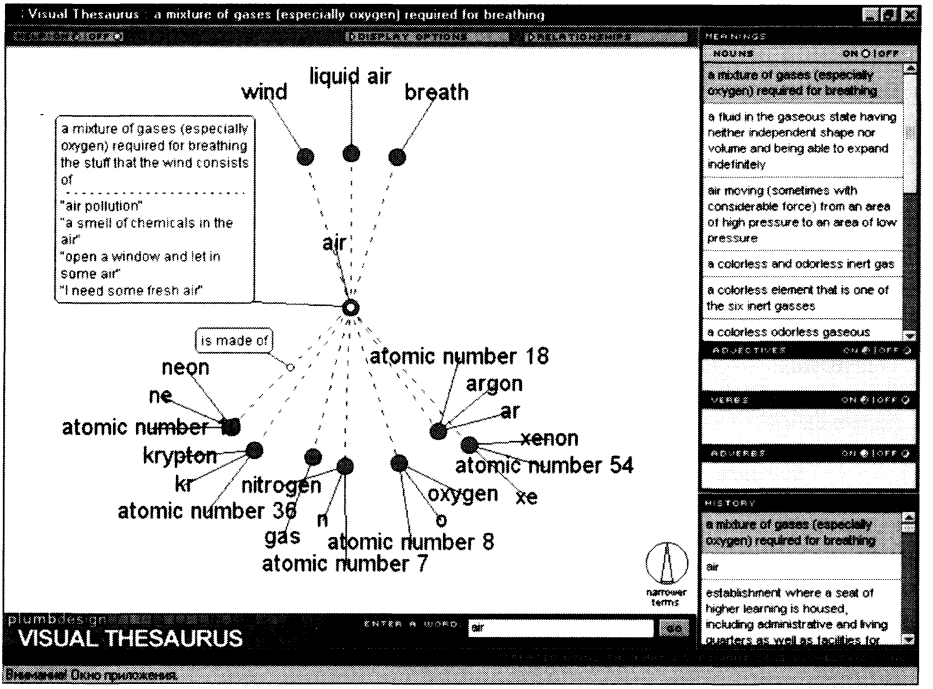


Рис. 3.15. Представление графа тезауруса в системе Visual Thesaurus (в центре окна располагается вершина-значение)

1. Создание одноязычных рефератов из источников на разных языках. На основе таких рефератов можно принимать решения, требуется ли полный перевод исходных документов.

2. Построение рефератов по гибридным источникам, включающим как текстовые, так и числовые данные в разных формах (таблицы, диаграммы, графики и т. д.). Например, документ может содержать статистическую информацию из реляционной БД и комментарии к ней. Методы реферирования для таких документов находятся на стадии теоретической проработки.

3. Создание рефератов на основе массивов документов. Например, построение единого реферата по сборнику тезисов докладов научной конференции. Для решения этой задачи требуются методы, позволяющие анализировать каждый документ из набора и формировать общий реферат путем объединения и обобщения извлеченных сведений. Соответствующие средства должны быть способны выявлять сходство и различие в содержании документов, отбрасывать избыточную информацию и генерировать краткое изложение содержания массива в целом. Одна из областей применения подобных средств — формирование новостных сообщений по газетным источникам.

4. Растущий объем мультимедийной информации обусловливает актуальность разработки средств ее автоматического реферирования. Методы извлечения семантики из мультимедийной информации находятся на начальных стадиях развития.

Средства автоматического аннотирования в целом аналогичны средствам автоматического реферирования. Однако требования к сжатию текста для них, как правило, на порядок более жесткие.

Основные выводы

1. Технологии автоматического реферирования и аннотирования только начинают свою эволюцию. Будущее принадлежит системам, основанным на знаниях. Это требует создания и использования представительных словарей-тезаурусов (таких, как WordNet) и онтологических справочников (таких, как *Cyc* и Penman Upper Model).

2. Для обучения NLP-систем можно использовать большие хранилища текстов и рефератов к ним (например, на основе The Wall Street Journal).

3. В современных системах автоматического реферирования и аннотирования используется комбинированный подход, сочетающий статистические методы и методы, основанные на знаниях.

4. Системы автоматического реферирования и аннотирования должны поддерживать распространенные языки разметки и форматы документов (такие, как HTML, XML, RTF, PDF, DOC), а также основные форматы метаданных для информационных ресурсов.

5. При разработке ЛО ИАС, обрабатывающих тексты на ЕЯ, используются психолингвистические теории организации лексической памяти человека и методы математической лингвистики. Примером служит система WordNet.

Вопросы для самопроверки

1. Чем отличается реферат от аннотации?
2. Почему автоматическое реферирование и аннотирование относят к технологиям ИИ?
3. На чем основываются поверхностные и глубинные методы автоматического реферирования и аннотирования?
4. Какие системы автоматического реферирования и аннотирования Вы знаете?
5. Какие требования предъявляются к реферату?
6. Перечислите виды рефератов.
7. Каковы основные идеи метода составления выдержек?
8. Охарактеризуйте модель линейных весовых коэффициентов. Каковы ее достоинства и недостатки?
9. Какие подходы реализуются в системах автоматического реферирования, основанных на знаниях?

10. Какую роль играют тезаурусы типа WordNet для систем автоматического реферирования и аннотирования?
11. Охарактеризуйте математическую модель тезауруса WordNet.
12. Какие типы тезаурусных отношений представлены в WordNet?
13. Что такое синсет?
14. Какие задачи являются перспективными для систем автоматического реферирования и аннотирования?

3.4. Машинный перевод*

Всякое понимание есть недопонимание,
а всякое разумение есть недоразумение.

Потебня, русский лингвист, XIX век

Машинный перевод (МП) текстов с одних ЕЯ на другие — одна из наиболее ранних задач невычислительных приложений ЭВМ и ИИ. Отметим два аспекта, определяющих актуальность задач МП и не снижающееся внимание к ним со стороны ученых и разработчиков ИАС:

- все возрастающая потребность в переводах в науке, литературе, дипломатии, экономике и других областях деятельности, обусловливаемая повышением открытости границ, интернационализацией науки и экономики, взаимопроникновением культур и т. д.;
- для МП гораздо яснее критерии оценивания результатов, чем в задачах понимания текстов, организации диалога и др.

Создание систем МП требует совместной работы специалистов разного профиля: в первую очередь, лингвистов, математиков и программистов.

Системы МП различают по трем аспектам:

- рабочим языкам;
- типам текста;
- ограничениям по ПрО.

По количеству поддерживаемых рабочих языков различают двуязычные и многоязычные системы МП. Язык исходного текста называется входным, а язык перевода (формируемого текста) — выходным. На рис. 3.16, *а* условно представлены две системы МП, обеспечивающие перевод с языка 1 на язык 2 и с языка 2 на язык 1. На рис. 3.16, *б* условно изображены два класса систем МП. Системы первого класса переводят текст с языка 1 на языки 2.1, 2.2, ..., 2.*k*, а системы второго класса переводят текст с языков 2.1, 2.2, ..., 2.*k* на язык 1.

* Содержание параграфа соответствует направлению исследований в области ИИ 1.4.2.

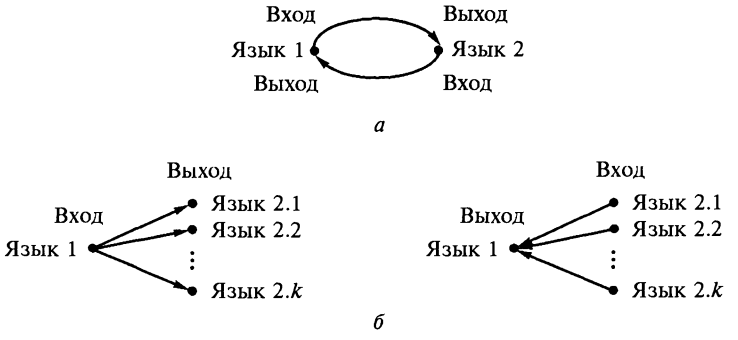


Рис. 3.16. Системы МП:
а — двуязычные; *б* — многоязычные

В современных многоязычных системах МП поддерживаемые языки могут быть и входными, и выходными. Направление перевода определяет роли языков (входной, выходной).

По типу текста выделяются системы для перевода письменного текста и устного диалога. Системы первого типа классифицируются по назначению для перевода:

- деловой прозы (научно-технических статей, заголовков и аннотаций, описаний изобретений, технической документации и др.);
- художественной литературы.

Системы для перевода устного диалога обычно ориентированы на узкую тематику: резервирование мест в гостинице, определение маршрута проезда по городу и т. д. Они интегрируются с системами анализа и синтеза устной речи.

Ограничения систем МП по ПрО обусловлены поддержкой в них лексики, соответствующей той или иной области знаний (медицины, информатики, математики и т. д.).

До последнего времени отсутствовали промышленные *системы распознавания русской речи* (звукового представления текста). К решению этой проблемы подключились компании Intel и Cognitive Technologies (известный российский разработчик OCR-систем). Их совместный продукт получил название RuSpeech. В его основе лежит БД, содержащая цифровое представление звучания непрерывной русской речи с соответствующими текстами и фонетической транскрипцией. БД включает звуковые фрагменты для более 50 тыс. предложений с фонетической разметкой каждого из них. Система «сверяет» с ними естественную речь человека, распознавая не только слова, уже присутствующие в БД, но и отдельные фонемы и их последовательность. Это позволяет минимизировать количество ошибок при распознавании новых слов, отсутствующих в БД.

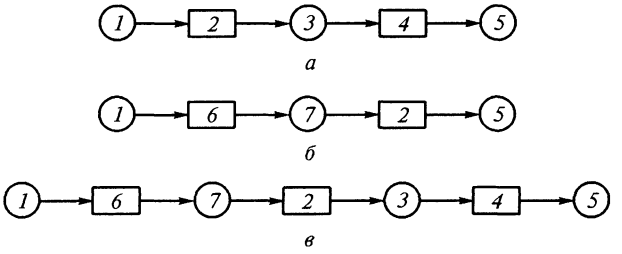


Рис. 3.17. Автоматизированные системы МП:

а — с постредактированием; *б* — с предредактированием; *в* — с пред- и постредактированием; *1* — входной текст; *2* — система МП; *3* — перевод, сформированный системой МП; *4* — человек (редактор), обрабатывающий с помощью текстового редактора перевод, сформированный системой МП; *5* — выходной текст; *6* — человек (редактор), выполняющий предварительную обработку входного текста с помощью текстового редактора; *7* — входной текст после предварительного редактирования человеком

В создании БД RuSpeech приняли участие 220 дикторов. Она содержит около 50 часов непрерывной речи, имеет объем 15 Гб и размещается на 30 CD-ROM.

Практическое применение RuSpeech связано с речевой реализацией пользовательского интерфейса программных систем. Словарный запас RuSpeech достаточен для понимания говорящего в реальном времени. По масштабности RuSpeech может конкурировать с лучшими мировыми аналогами. Фактически это означает новый этап развития речевых технологий в России. По мнению создателей RuSpeech интеграция уникальной звуковой БД с передовыми технологиями анализа и распознавания речи уже в ближайшее время должна привести к созданию речевых интерфейсов, применимых в промышленности, мобильной связи, Internet-порталах, системах управления и иных приложениях.

Системы МП бывают автоматическими и автоматизированными. Во втором классе ряд функций остается за человеком. На рис. 3.17 изображены три схемы автоматизированных систем МП. Их достоинствами являются простота реализации и повышение производительности перевода в 3—5 раз по сравнению с переводом вручную человеком. Недостаток таких систем связан с необходимостью участия в переводе специалиста в ПрО, к которой относится текст, владеющего входным и выходным языками.

Как обычно, перед описанием схемы автоматического решения интеллектуальной задачи полезно рассмотреть процесс ее решения человеком. Выполняя перевод, человек уясняет смысл очередного фрагмента текста (фразы*, абзаца) и выражает его на выходном языке, стараясь обеспечить

* Фраза — законченный оборот речи, предложение.

структурную и смысловую близость к оригиналу (без этого результатом будет не перевод, а пересказ). При переводе человек использует как *лингвистические знания* о входном и выходном языках, так и *экстралингвистические знания* (знания о ПрО, общих закономерностях среды перевода, законах коммуникации). В соответствии с возможностями компьютерной реализации данных функций человека и разрабатывались поколения систем МП. Выделяют три поколения таких систем [78]:

1) П-системы — системы прямого перевода (direct systems);

2) Т-системы (от слова transfer — преобразование);

3) И-системы (от слова interlingua — язык-посредник).

Цикл работы ***П-системы*** состоит из трех этапов. На первом выполняется морфологический анализ входной фразы. С помощью базы правил для входного языка и двух словарей (словаря основ слов и словаря оборотов) она переводится в ее морфологическое представление. При этом каждой основе и каждому обороту ставятся в соответствие свои наборы признаков. Таким образом, морфологическим представлением фразы является множество пар (признак, значение).

На втором этапе выполняется перевод морфологического представления входной фразы в морфологическое представление выходной фразы. Для этого используется база правил соответствия морфологических признаков входного и выходного языков.

На третьем этапе выполняется морфологический синтез: устанавливаются нужный порядок и форма слов согласно правилам грамматики выходного языка. Итоговый результат по качеству получается немного лучше подстрочного перевода.

В ***Т-системах*** помимо процедур морфологической обработки реализуются методы синтаксического анализа и синтеза. Работа Т-системы включает пять этапов. На первом осуществляется морфологический анализ входной фразы (аналогично П-системам). На втором этапе по его результатам выполняется синтаксический анализ, в ходе которого строится представление входной фразы в виде синтаксического дерева (дерева синтаксического разбора). Различают два типа таких деревьев:

- деревья синтаксических составляющих;
- деревья синтаксических зависимостей.

В первом случае грамматика ЕЯ описывается в виде моделей Н. Хомского [75]. Дерево составляющих представляет вложенные группы словоформ. Самая крупная словоформа соответствует фразе, самые мелкие — синтаксически неделимым текстовым единицам (словам, словосочетаниям).

Во втором случае узлы дерева представляют синтаксические единицы текста, а дуги — отношения подчинения между ними. Это позволяет использовать при анализе фильтровый метод.

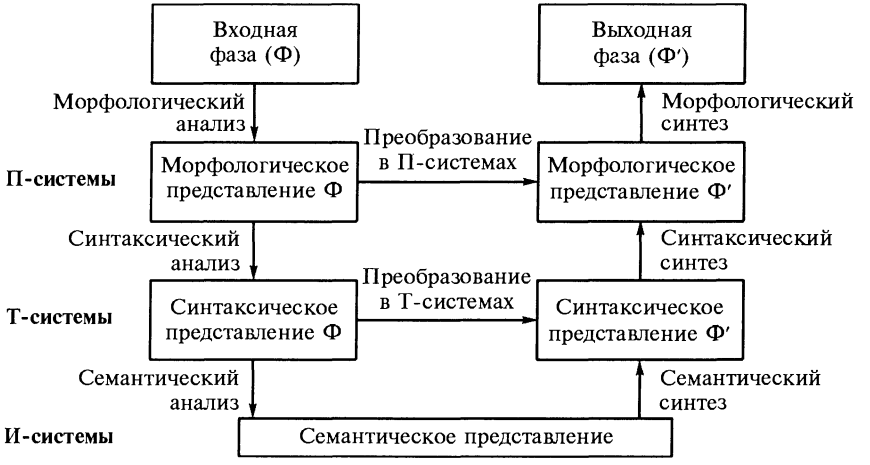


Рис. 3.18. Отношения между этапами функционирования трех поколений систем МП

На третьем этапе выполняется переход от входного к выходному языку. Для этого синтаксическое дерево входной фразы преобразуется в синтаксическое дерево выходной фразы. Выделяются три уровня преобразования:

- поверхностно-синтаксический;
- глубинно-синтаксический;
- синтактико-семантический.

В соответствии с их поддержкой различают и Т-системы.

На четвертом этапе проводится синтаксический синтез. Грамматические правила в Т-системах имеют декларативную (дескриптивную) форму.

На пятом этапе, как и в П-системах, осуществляется морфологический синтез.

В ***И-системах*** наряду с морфологией и синтаксисом используются экстралингвистические знания, т. е. знания о семантике и прагматике ПрО. Поэтому после этапов морфологического и синтаксического анализа входной фразы функционирование И-системы включает этап семантического анализа. Его результатом служат семантические представления входной и выходной фраз, эквивалентные с точностью до лексики.

Отношения между этапами функционирования трех поколений систем МП иллюстрирует рис. 3.18.

Таким образом, системы МП представляют собой сложные программные комплексы с разными видами обеспечений. К лингвистическому обеспечению систем МП относятся:

- словари слов и словосочетаний с соответствующими признаками;
- морфологические таблицы суффиксов и окончаний;

• базы грамматических правил и др.

Математическое обеспечение включает модели для представления лингвистической информации и алгоритмы их преобразования, правила логического вывода для уточнения обрабатываемого текста на основе экстралингвистических знаний. К программному обеспечению относятся программы выполнения перевода, ведения словарей, формирования базы правил и т. д. Информационное обеспечение (ИО) представляет база экстралингвистических знаний о ПрО.

К числу наиболее распространенных в России систем МП и компьютерных словарей относятся:

• Stylus — система МП, включающая множество словарей по разным ПрО;

• Universal Translator — многоязычная система МП;

• Socrat — система, позволяющая сканировать документы, переводить их содержимое и проверять орфографию;

• Polyglossum — многоязычная система МП с широким набором предметных словарей;

• Promt — многоязычная система МП, содержащая множество словарей по разным ПрО;

• WebTranSite — система для перевода web-страниц;

• Lingvo — компьютерный англо-русский и русско-английский словарь.

Основные характеристики компьютерного словаря Lingvo (разработчик — компания ABBYY Software House):

• перевод слова, набранного в панели ввода словаря или перенесенного на пиктограмму работающей системы с помощью операции «drag and drop»;

• перевод слова из буфера промежуточного хранения по горячей клавише;

• одновременная работа с большим количеством предметных словарей;

• гипертекстовое представление словарных статей;

• наличие тезауруса;

• наличие звуковой базы, представляющей произношение основных английских слов;

• полнотекстовый поиск слов и словосочетаний в статьях всех словарей;

• пословный перевод фразы;

• вставка перевода в редактируемый текст с помощью операции «drag and drop»;

• представление транскрипции, грамматических характеристик и парадигмы слова (списка всех его форм);

• предоставление подсказки по правильному написанию слова;

• создание и ведение собственных словарей.

На сегодняшний день лидером в области систем МП является Япония.

Основные выводы

1. МП — активно развиваемая технология ИИ. Она базируется на различных схемах перевода текстов на ЕЯ человеком, использовании знаний о морфологии, синтаксисе, семантике входных и выходных языков, а также экстралингвистических знаний.

2. Современные системы МП значительно (в десятки раз) увеличивают производительность перевода, но по качеству еще не могут сравняться с человеком. Основные трудности связаны с реализацией этапов семантического анализа и синтеза (т. е. с проблемой понимания естественно-языкового текста).

3. Перспективным направлением совершенствования систем МП является использование онтологических словарей и БЗ.

Вопросы для самопроверки

1. Как классифицируются системы МП?
2. Какие схемы обработки текста используются при автоматизированном МП?
3. Чем различаются П-, Т- и И-системы МП?
4. Что такое экстралингвистические знания, и как они используются в системах МП?
5. Почему МП относят к технологиям ИИ?
6. Каковы перспективы систем МП?

3.5. Автоматическая классификация документов*

В действительности же наука начинается с классификации...

Д.А. Поспелов

Потребности в средствах автоматической классификации документов испытывают:

- корпоративные системы документооборота;
- каталоги Internet;
- каналы вещания;
- службы электронной почты;
- электронные библиотеки;
- информационные агентства;
- Internet-порталы и др.

* Содержание параграфа соответствует направлению исследований в области ИИ 1.4.4.

Эффективность поиска в большом информационном массиве существенно повысится, если его разбить на части по некоторому критерию, связанному с целями поиска. Таким образом, классификация документов позволяет сузить область поиска и не только увеличить его скорость, но и значительно повысить точность результатов. Поэтому технологии автоматической классификации документов отводится важное место в системах управления документооборотом.

Суть задачи классификации состоит в автоматическом распределении поступающих в систему документов в зависимости от их типа и содержания по рубрикам (классам).

В теории ИС различают *два типа классификации* [85]. Первый тип предусматривает распределение документов как элементов некоего формального множества по классам по аксиоматически определенным критериям. В рамках второго типа документы классифицируются на основе их эмпирического анализа для достижения заранее заданной цели.

Первый тип классификации подходит для библиотечных ИС, в которых книги, электронные издания и другие информационные ресурсы (ИР) распределяются по достаточно устойчивой системе рубрик. В корпоративных ИС большинство документов первоначально классифицируются приблизительно (неточно), а поисковые запросы «размыты». Поэтому здесь преимущество имеют подходящие для конкретных учреждений эмпирические динамические классификации.

На практике используются следующие *критерии оценивания качества эмпирической классификации:*

- результаты классификации не должны зависеть от порядка обработки документов;
- классификация должна быть устойчивой (малые изменения исходных данных не должны сильно влиять на результаты);
- классификация не должна зависеть от объема выборки (масштабная независимость);
- классификация должна быть кластеризующей (объекты, обладающие большим сходством, не должны попадать в разные классы).

Коротко рассмотрим основные подходы к автоматической классификации документов.

Достаточно эффективен *метод группировки и поиска ближайшего соседа.* Классы формируются путем вычисления «расстояния» между парами документов и объединения ближайших соседей в кластеры. Метод нагляден и прост. Он дает хорошие результаты при удачном определении понятия «расстояние» между документами. В настоящее время он используется в рамках интерактивных кластерных методов. При работе с реализующей их ИС человек, регистрируя входящие документы, видит результаты кластеризации и может при необходимости вмешиваться в этот процесс.

Развитые системы управления документооборотом выполняют классификацию, формируя классы автоматически при поступлении документов в систему независимо от пользователя. При этом документ может быть одновременно отнесен к нескольким классам в соответствии с различными основаниями классификации.

Технология, реализованная в *средствах фильтрации Microsoft Outlook,* включает следующие этапы:

- ручное построение списка рубрик;
- формирование для каждой рубрики ее семантического образа, представляемого составляемым вручную набором ключевых слов (дескрипторов);
- применение программы многоаспектной сортировки, играющей роль порогового разделителя.

Проблемы, возникающие при использовании такого подхода, обусловлены:

- статичностью системы;
- наличием в тексте различных грамматических форм слов и синонимов ключевых слов;
- зависимостью важности слов от контекста;
- большой изменчивостью слов, характерной для ряда языков (в частности, русского и немецкого).

Другой подход к решению задачи автоматической классификации связан с *использованием запросов как основы классификации.* Он предусматривает:

- превращение списков ключевых слов в поисковые запросы;
- передачу запросов поисковым машинам, применяющим их по отношению ко множеству поступивших документов;
- использование при поиске разнообразных лингвистических средств (процедур морфологического анализа, словарей синонимов и т. д.).

Недостатками данного подхода являются фиксированный набор рубрик и ручное построение наборов ключевых слов.

Некоторые новые продукты способны *самостоятельно формировать семантические образы рубрик после самообучения.* Администратор системы указывает рубрики и «образцовые» документы для обучения алгоритмов классификации. Система выделяет в обучающей выборке значимые слова и словосочетания, приводит их к базовым словарным формам, подсчитывает различительную силу терминов и составляет семантические образы из наиболее различительных терминов.

Преимущества такого подхода:

- легкая настройка системы на изменяющийся поток документов;
- бóльшая эффективность по сравнению с системами, предусматривающими ручное формирование наборов ключевых слов.

Данный подход реализован в продуктах Inxight Categorizer* и «МедиаЛингва Классификатор SDK 2.0». Первый продукт обрабатывает более 70 форматов документов на 11 западноевропейских языках. В нем используется метод группировки и поиска ближайшего соседа. Inxight Categorizer может быть интегрирован в Internet-порталы и другие приложения. Он способен взаимодействовать с СУБД, поддерживающими XML-запросы.

Второй продукт представляет собой инструментарий для реализации функций автоматической классификации в ИАС. Его программные компоненты обеспечивают обработку документов на русском и английском языках в форматах TXT, HTML, DOC, RTF и PDF. Алгоритмы классификации учитывают статистические, морфологические и синтаксические характеристики содержимого документов. Сведения о семантических образах и текущем составе рубрик могут быть представлены на XML.

Проблематика автоматической классификации документов будет детализирована в следующем параграфе на примерах конкретных систем.

Основные выводы

1. Автоматическая классификация документов – активно развивающаяся технология ИИ. Она относится к ИИ, так как базируется на механизмах, обеспечивающих понимание естественно-языкового текста.

2. Классификация документов позволяет сузить область поиска, повысить его скорость и точность результатов.

3. Развитие методов автоматической классификации документов связано с использованием онтологического подхода.

Вопросы для самопроверки

1. В каких системах используются средства автоматической классификации документов?
2. Каковы основные подходы к реализации функций автоматической классификации документов?
3. Перечислите критерии качества эмпирической классификации.
4. Какие этапы включает технология автоматической классификации документов, реализованная в средствах фильтрации Microsoft Outlook?
5. Каким образом формируются семантические образы рубрик в методах автоматической классификации документов?

* http://www.inxight.com.

3.6. Комплексные интеллектуальные программные системы для обработки текстов*

Заберите у меня все, чем я обладаю, но оставьте мне мою речь, и скоро я обрету все, что имел.

Д. Уэбстер

Ряд коммерческих программных продуктов реализуют несколько рассмотренных в предыдущих параграфах интеллектуальных технологий обработки текстов на ЕЯ. В данном параграфе описываются три таких продукта:

- комплексный смысловой анализатор текста Text Analyst;
- промышленная ИПС Excalibur RetrievalWare (разработчик — фирма Convera Technologies Corp.; новое название продукта — Convera RetrievalWare);
- пакет NeurOK Semantic Suite (разработчик — компания «НейрОК Интелсофт»).

3.6.1. Комплексный смысловой анализатор текста Text Analyst

Анализатор текста Text Analyst** — отечественное интеллектуальное программное средство для работы с текстовыми документами. Text Analyst относят к категории *программ-экстракторов.* Он предоставляет пользователям следующие основные возможности:

- анализ содержания текста с автоматическим формированием семантической сети — построение «смыслового портрета» документа в терминах основных понятий и их смысловых связей;
- анализ содержания текста с автоматическим формированием тематического дерева — выявление семантической структуры документа в виде иерархии тем и подтем;
- смысловой поиск с учетом скрытых семантических связей слов запроса со словами документа;
- автоматическое реферирование текста — построение его «смыслового портрета» в терминах наиболее информативных фраз;
- кластеризация информации — анализ распределения материала документа по тематическим классам;

---

* Содержание параграфа соответствует направлениям исследований в области ИИ 1.3, 1.4, 2.2.2, 2.3.1, 2.3.3 и 4.2.

** http://www.analyst.ru.

- автоматическая индексация текста с преобразованием в ГТ (автоматическая расстановка гиперссылок);
- ранжирование всех видов информации о семантике текста по степени значимости с возможностью варьирования детальности ее исследования;
- автоматизированное формирование полнотекстовой БД с гипертекстовой структурой и возможностями ассоциативного доступа к информации.

В Text Analyst воплощены процессы, аналогичные некоторым механизмам правополушарного мышления человека. Имеется в виду функциональная аналогия по входу и выходу с процессами, протекающими при так называемом «обучении с погружением».

Процедуры обработки текста включают:

- предварительный анализ текста (выделение в тексте понятий, входящих в базовые словари);
- статистический анализ текста — определение частот встречаемости в тексте слов и словосочетаний (важность понятия оценивается по частоте его использования в тексте);
- по результатам частотного анализа формирование семантической сети для анализируемого текста, отражающей связи между понятиями и объединяющей их в единую смысловую картину (перед построением семантической сети устанавливается порог значимости для понятий и связей между ними);
- на основе семантической сети построение тематической структуры текста в виде дерева или леса понятий (каждой теме соответствует свое дерево понятий);
- автоматическое реферирование текста на основе его тематической структуры;
- формирование гипертекстовой разметки;
- смысловой поиск информации.

Основные принципы, реализуемые Text Analyst:

- принцип ассоциативности;
- построение структуры понятий, представляющей текст, в соответствии с их важностью и взаимосвязями;
- формирование тематической структуры текста в виде многоуровневой иерархии тем и раскрывающих их подтем.

Суть *принципа ассоциативности* заключается в использовании такой модели представления текста, при которой его фрагменты указывают на места их хранения. Эта модель управляет механизмами статистической обработки текста: если фрагменты совпадают, то они указывают на одно и то же место, где записывается частота их встречаемости. В результате частотного анализа формируется семантическая сеть — основная структура, характеризующая смысл текста, в которой понятия (слова и словосочетания)

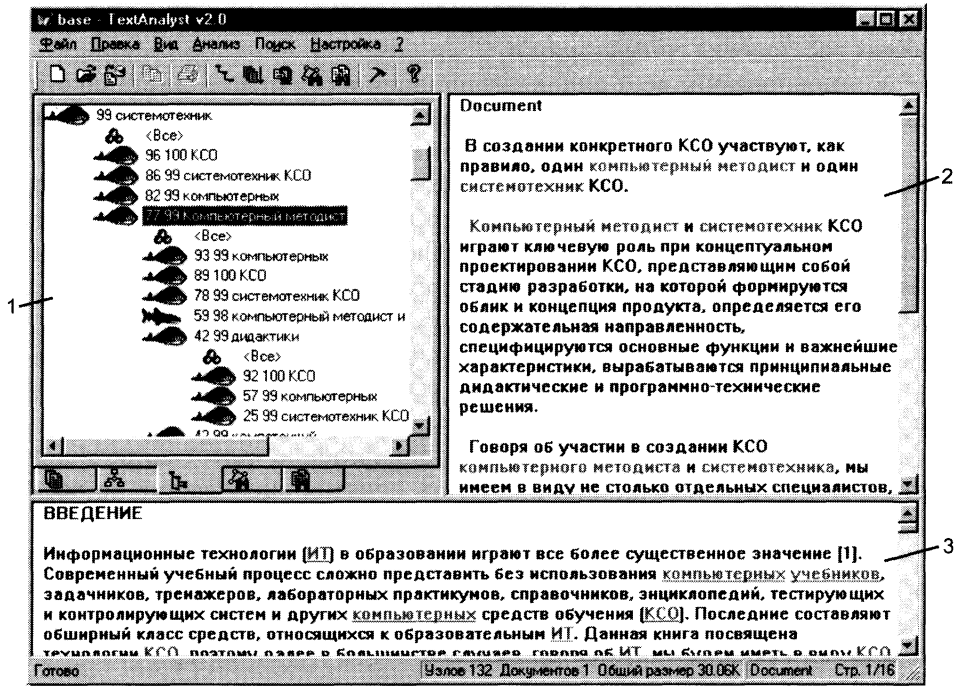


Рис. 3.19. Интерфейс системы Text Analyst:
1—3 — дочерние окна

объединяются ассоциативными связями в соответствии с их совместной встречаемостью. Таким образом, на первом этапе анализа текста все отношения между понятиями условно считаются ассоциациями.

Интерфейс Text Analyst изображен на рис. 3.19. Главное окно приложения содержит три дочерних окна. В окне *1* представляется формируемая (частотно) семантическая сеть или тематическая структура. В окне *2* размещаются выделенные для анализа предложения, в обработку которых можно вмешиваться. В окне *3* отображается исходный текстовый документ.

Помимо применения Text Analyst в качестве самостоятельного программного средства его функции с помощью библиотеки Text Analyst SDK могут встраиваться в прикладные программы. По отношению к модулям Text Analyst взаимодействующее с ними приложение является клиентом (рис. 3.20).

Text Analyst разработан с использованием объектно-ориентированного подхода и COM-технологии. На их основе реализованы два программных объекта: лингвистический процессор (ЛП) и алгоритмическое ядро (АЯ).

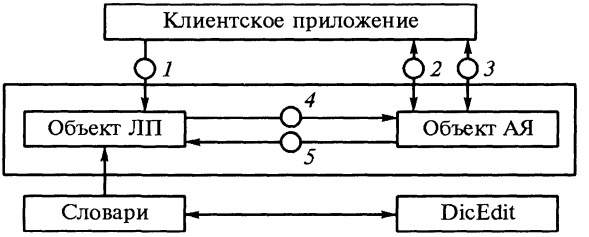


Рис. 3.20. Схема взаимодействия Text Analyst с клиентским приложением:

ЛП — лингвистический процессор (модуль преобразования текста); АЯ — алгоритмическое ядро (модуль анализа текста); DicEdit — редактор словарей; *1* — интерфейс обработки команд пользователя; *2* — интерфейс хранения данных; *3* — интерфейс между клиентским приложением и АЯ; *4* — поток данных от ЛП к АЯ; *5* — поток данных от АЯ к ЛП

Основными функциями Text Analyst являются:

1) создание и редактирование словарей (основных и тематических);

2) построение частотной семантической сети;

3) построение иерархической тематической структуры текста;

4) формирование реферата текста;

5) автоматическое нахождение в тексте мест для установления гиперссылок;

6) смысловой поиск информации;

7) поддержка технологий:

- автоматизации web-дизайна;
- создания гипертекстовых электронных документов;
- построения полнотекстовых БД.

Приложение DicEdit позволяет настраивать словари на ПрО анализируемых текстов или создавать собственные словари. На основе лингвистических правил и словарей в тексте входного документа выделяются последовательности слов, которые вместе с результатами семантического анализа заносятся в БД, обеспечивающую хранение всей информации о содержании текста.

Основные функции модуля ЛП:

- выделение из текста последовательности слов;
- исключение из этой последовательности элементов словаря удаляемых слов (он содержит малозначимые и неинформативные слова);
- маркировка слов атрибутами, определяющими их типы;
- приведение словоформ к базовой грамматической форме.

Таким образом, на вход ЛП поступает строка текста, а на его выходе формируется последовательность слов, маркированных атрибутами, определяющими их типы. Словарь ЛП устанавливает набор слов, удаляемых из текста, и атрибуты слов в выходной последовательности.

Лингвистический процессор создает БД, в которую заносит всю лингвистическую информацию об анализируемом тексте. В дальнейшем работа происходит с этой БД, а не с исходным текстом, что значительно упрощает обработку и увеличивает ее производительность.

Text Analyst включает два базовых словаря (normal_rus.dic и normal_eng.dic) для русского и английского языков. Для них предусмотрены 4 подсловаря:

- словарь удаляемых слов;
- словарь общеупотребимых слов;
- словарь слов-предпочтений пользователя (предметных понятий);
- словарь слов-исключений из правил нормального словоизменения.

На вход модуля АЯ поступает последовательность слов с атрибутами, определенными ЛП. Данный модуль выполняет следующие основные функции:

- статистический анализ входных последовательностей слов и выделение понятий, под которыми в Text Analyst понимаются слова и словосочетания, встречаемость которых в тексте не ниже установленного порога;
- определение смысловых связей между понятиями;
- задание ссылок на предложения, в которые входят выделенные понятия.

На выходе АЯ формируются компоненты БД, представляющие содержание текста. Основой этих компонентов служит *семантическая сеть,* т. е. множество слов и словосочетаний, связанных между собой по пороговому атрибуту (частоте встречаемости). Построенная таким способом семантическая сеть передает смысл текстов, значительно сокращая при этом объем исходной информации (за счет исключения несущественных деталей). Она представляет собой индекс анализируемого текста, который может быть эффективно использован для реализации различных методов доступа к тексту, в том числе ассоциативного (смыслового) поиска.

Вершинами семантической сети являются слова и словосочетания, несущие в тексте основную смысловую нагрузку. Они выделяются по частоте встречаемости в тексте. Пороговое значение этого параметра может задаваться пользователем. В формируемой семантической сети каждое понятие, многократно упомянутое в тексте, представляется единственным элементом, приведенным к базовой грамматической форме. Связи между вершинами отражают совместное использование понятий в тексте. Кроме того, вершина соотносится со списком предложений, в которых употреблено соответствующее ей понятие. Таким образом, в «смысловом портрете» текста интегрируется информация, относящаяся к понятиям.

Каждое понятие, вошедшее в семантическую сеть, представляет некоторую тему текста и характеризуется числовой оценкой — *смысловым ве-*

сом. Эта же оценка приписывается и связям между понятиями. Значение смыслового веса лежит в интервале от 1 до 100 и отражает важность понятия по отношению к смыслу всего текста. Чем оно больше, тем важнее понятие. Понятия с максимальными значениями (равными или близкими к 100) являются ключевыми и представляют важнейшие темы текста.

Высокое значение веса связи первого понятия со вторым указывает на то, что бóльшая часть информации в тексте, относящаяся к первому понятию, относится и ко второму. Однако связь первого понятия со вторым не всегда имеет тот же вес, что и связь второго понятия с первым.

Пользователь может настраивать средства визуализации семантической сети, устанавливая пороговые веса отображаемых понятий и связей, а также способ их сортировки.

Семантическая сеть представляется в окне *1* (см. рис. 3.19). Щелкнув мышью возле выбранного понятия (вершины), можно раскрыть список всех понятий, связанных с ним. Щелчок мыши возле вершины с раскрытым списком закрывает его. Чтобы просмотреть всю информацию по данному понятию, нужно щелкнуть мышью на пункте «Все» в его раскрытом списке. В окне *2* появятся все предложения анализируемых документов, содержащие данное понятие, которое будет выделено цветом.

Для получения информации, касающейся связи пары понятий, необходимо щелкнуть мышью возле второго понятия в раскрытом списке первого понятия. В окне *2* появятся все предложения текстов, в которых встречается данная пара понятий. Оба понятия выделяются цветом. Щелчок по предложению в окне *2* вызывает отображение исходного фрагмента текста в окне *3*.

Тематическая структура описывает содержание анализируемых текстов в виде иерархии тем и подтем. Она задается деревьями, в корнях которых располагаются главные темы, а в промежуточных узлах и листьях – подтемы. Все темы и подтемы выражаются понятиями исходных текстов и соответствуют вершинам семантической сети. Однако связи между понятиями являются односторонними и направлены от подчиняющих понятий к подчиненным. В результате представление тематической структуры оказывается иерархическим.

Тематическая структура отражает смысловое строение текстов. Так, если все их содержание подчинено одной теме, то структура описывается единственным деревом. Если же содержание текстов политематично, то будет сформирован лес независимых деревьев, корни которых представляют главные темы, не связанные друг с другом.

Пороговые значения весов понятий и связей, учитываемых при построении тематической структуры, устанавливаются пользователем. Изменение этих параметров позволяет анализировать структуру текста в разных смысловых плоскостях, выделяя наиболее важные понятия и связи.

Смысловые веса понятий и связей между ними используются при *автоматическом реферировании* текста. Формируемый реферат содержит список наиболее информативных предложений, отражающих основные смысловые связи между главными понятиями семантической сети. Поскольку предложения, включаемые в реферат, выбираются Text Analyst из исходного текста в порядке встречаемости, они не связаны стилистически, что порождает проблему стыков между ними. Другими словами, текст реферата требует ручного «сглаживания». В то же время даже такой «подстрочник» позволяет составить общее представление о тексте и ознакомиться с его основными идеями.

Все предложения реферата снабжены ссылками на соответствующие фрагменты исходных текстов, что дает возможность просмотреть контекст того или иного тезиса. Подробность реферата настраивается путем задания количества входящих в него предложений. При этом каждое предложение характеризуется относительной степенью значимости для всего текста.

В области *автоматизации построения ГТ* Text Analyst позволяет автоматически превратить мегабайтный массив текстовой информации в ГТ, выделив существенные смысловые взаимосвязи между его фрагментами. Основой для формирования ГТ служит семантическая сеть. Ее проекция на исходные тексты трансформирует их в ГТ. В текстах выделяются цветом понятия семантической сети, рекомендуемые в качестве гиперссылок, которые ведут к фрагментам, содержащим либо эти понятия, либо другие понятия, связанные по смыслу с исходными. В результате возникает возможность циклического движения по цепочке: выбранный фрагмент текста — понятия семантической сети — выбранная гиперссылка — фрагмент текста.

Функция *смыслового поиска* Text Analyst позволяет получить ответ на запрос, выраженный в виде фразы ЕЯ, словосочетания или набора ключевых слов. Извлекаемая в ответ информация, связанная по смыслу с запросом, может явно не фигурировать в нем или содержать термины из запроса в других грамматических формах.

Запрос вводится с клавиатуры либо задается участком текста, выделенным мышью. Результаты его выполнения отображаются на экране в виде двух списков. Список в окне *2* включает предложения текстов, содержащие слова, которые связаны по смыслу со словами запроса, представленными в семантической сети. Предложения в списке упорядочены по количеству релевантных понятий, которые выделены цветом. Щелчок мыши на предложении в окне *2* вызывает отображение соответствующего фрагмента текста в окне *3*.

В списке в окне *1* представлены понятия семантической сети, упорядоченные по близости к запросу (степень близости выражает число от 1 до 100). Этот список показывает, что в текстах имеется информация, связанная

по смыслу с содержанием запроса. Дальнейшая работа со списком в окне *1* аналогична работе с семантической сетью.

3.6.2. Промышленная информационно-поисковая система Excalibur RetrievalWare

Информационно-поисковая система Excalibur RetrievalWare* (ERW) представляет собой мощное средство полнотекстового и атрибутивного поиска. Оно позволяет эффективно находить документы, используя в качестве клиентского места обычный web-браузер. ERW работает с естественно-языковыми текстами в различных форматах и кодировках, электронными таблицами, БД (ODBC-совместимыми СУБД, например, MS SQL, Oracle, Sybase, Informix и др.), базами почтовых систем (MS Exchange, Lotus Notes и др.) — всего более 200 форматов. ERW содержит инструментарий, позволяющий настраивать систему на поддержку специфических форматов документов.

Как показывает статистика, доля структурированных данных в современных электронных архивах составляет не более 20 % [90]. Остальные 80 % приходятся на различные текстовые документы. В связи с быстрым развитием мультимедиа изменился характер обрабатываемых электронных документов: кроме текстов на ЕЯ они могут включать графику, видео и звук.

Знания могут извлекаться не только из естественно-языковых текстов, но и таких источников, как фотографии, рисунки, схемы, звукозаписи, телевизионные и компьютерные изображения и т. д. Вначале их преобразуют в цифровую форму, а затем анализируют. Обработка цифровых данных произвольного вида традиционными средствами SQL-СУБД оказывается мало продуктивной. Обычно в таких системах полнотекстовый индекс строится на базе инвертированных списков, в которых словам или словоформам ставятся в соответствие адреса документов. При этом объем индекса для неструктурированных данных достигает до 300 % от объема БД. При работе с графическими и другими мультимедийными данными этот метод не подходит. Для анализа подобной информации предназначены методы, использующие нейронные сети. К их числу относится *технология адаптивного распознавания образов* (Adaptive Pattern Recognition Processing — APRP), созданная Convera Technologies Corp.

В технологии APRP применяется *бинарное индексирование,* при котором размер индекса даже для неструктурированных данных не превышает 30 % от объема исходной информации.

Архитектура ERW представлена на рис. 3.21.

* http://www.convera.com; http://www.vest-meta.ru.

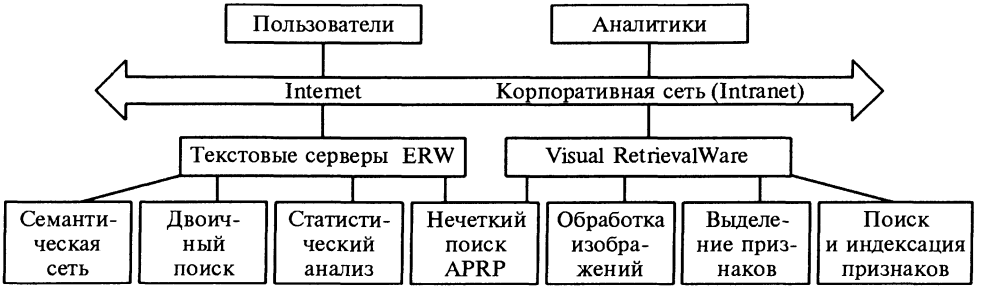


Рис. 3.21. Архитектура Excalibur RetrievalWare

Программные средства ERW позволяют вести ранжированный индексный поиск и поиск по шаблонам, в качестве которых могут выступать фотографии, графические эскизы, фрагменты текста и др.

В технологии APRP для обработки информации используется ИНС. ERW действует как самоорганизующаяся система, которая автоматически выделяет в исходных документах двоичные образы и индексирует их. К преимуществам APRP относятся возможность выполнения нечеткого поиска, высокая точность и полнота поиска, языковая независимость, малые объемы индексных файлов.

Нечеткий поиск, основанный не на выделении совпадений слов документа со словами запроса, а на вычислении их меры близости, позволяет исключить из цикла обработки бумажных документов дорогостоящий этап ручного исправления ошибок, возникших в процессе оптического распознавания символов.

Технология *семантического поиска* ERW ориентирована на работу со знаниями, содержащимися в текстовых документах. В ее основе лежит использование семантических сетей, описывающих смысл слов ЕЯ и связи между обозначаемыми ими понятиями. В ERW семантическая сеть рассматривается как тезаурус, позволяющий не только находить понятия, связанные по смыслу с данным понятием, но и определять количественно «семантическое расстояние» между ними.

К особенностям русского языка относится наличие множества словоформ, образованных от единой основы. Это повышает сложность реализации поиска, учитывающего вхождение данного слова во всех возможных словоформах. Заметим, что многие поисковые системы не учитывают морфологию и ищут либо точное вхождение заданного слова, либо строят словоформы по каноническим правилам.

Семантическая сеть словаря русского языка в ERW содержит около 90 тыс. семантических групп в базовом варианте поставки. Пользователи могут подключать к ERW лингвистические базы сторонних разработчиков.

Использование семантической сети позволяет выполнять запросы, выраженные на ЕЯ. При этом система способна находить документы, контекст которых совпадает с контекстом запроса. Реализуемые в ней модели и методы обеспечивают распознавание слов в любых грамматических формах. Для слов, имеющих несколько значений, пользователь может уточнить, какие именно значения он имеет в виду.

Технология семантического поиска позволяет одновременно работать с несколькими словарями. Например, помимо базового словаря к системе могут быть подключены отраслевой словарь, внутренний словарь организации и личный словарь пользователя.

Семантическая сеть применяется на двух этапах поиска. Во-первых, после ввода запроса входящие в него слова дополняются словами, связанными с ними по смыслу (синонимами, вариантами написания, аббревиатурами и т. п.). Это позволяет находить документы, в которых фигурирующая в запросе идея выражена по-другому (например, слово «Санкт-Петербург» будет расширено словами «Петербург», «Питер» и «северная столица»). Второй этап поиска, на котором используется семантическая сеть, состоит в упорядочении найденных документов по степени соответствия запросу. Применение семантической сети дает возможность учитывать общий контекст документа.

При работе с текстами на разных ЕЯ ERW поддерживает *многоязычный поиск* в двух вариантах:

- использование в одном запросе разных языков и указание языка в явном виде (multi-language search);
- перевод запроса на все языки, документы на которых есть в системе (cross-language search).

Информационно-поисковая система ERW обладает развитым языком построения поисковых запросов, включающим логические и контекстные операторы и метасимволы.

Как видно из рис. 3.21, текстовые серверы ERW обеспечивают три традиционных метода поиска информации:

- методы индексного или двоичного поиска;
- статистические методы;
- методы семантического поиска, использующие семантическую сеть (БЗ).

Помимо перечисленных традиционных методов в ERW реализован оригинальный метод нечеткого поиска на основе APRP.

Индексный или *двоичный поиск* применяется для работы со структурированными данными в БД. В этих методах слова представляются как инвертированные последовательности закодированных символов. Используя формальный синтаксис языка запросов, средства двоичного поиска опреде-

ляют точное соответствие для отдельного слова, цепочки слов либо слов, связанных логическими операторами. Однако в методах двоичного поиска не учитываются различные формы и значения слов, что снижает полноту и точность результатов. Средства двоичного поиска также не позволяют ранжировать документы по степени соответствия запросу.

Статистические методы основаны на использовании частотных характеристик текста: частоты вхождения слова в документ, частоты совместного вхождения нескольких слов, взвешенной частоты вхождения. В этих методах отношения между словами не анализируются с лингвистической точки зрения. Поэтому статистические методы не всегда обеспечивают желаемую точность и полноту результатов поиска, так как важность терминов не напрямую зависит от частоты их употребления в документе.

Методы семантического поиска основываются на четырех механизмах, связанных с применением:

- словарей синонимов;
- иерархии понятий, формируемой пользователем;
- базы лингвистических правил для грамматического анализа текста (эта же база применяется для поиска и ранжирования групп родственных документов; к недостаткам данного механизма относится зависимость от ПрО — для каждой ПрО эта база правил требует обновления);
- специальных семантических сетей, которые могут уточняться пользователями с целью повышения точности поиска.

Если в ИС необходимо представить содержимое документов на бумажных носителях, то они переводятся в электронную форму. Для этого используется технология OCR, рассмотренная в гл. 2. Количество ошибок на выходе OCR-системы при вводе документов хорошего качества может достигать 4 % от числа распознанных символов. Применение полуавтоматических методов коррекции ошибок обходится дорого и связано со значительными временными затратами. Средства нечеткого поиска на основе APRP позволяют отказаться от них. Искусственные нейронные сети, реагирующие не на совпадение слов из поискового запроса и анализируемого текста, а на семантическую меру их близости, установленную в процессе обучения сети, обеспечивают точные результаты поиска даже при наличии значительного числа ошибок, оставшихся после OCR-системы.

Особенностью ERW является совместное использование семантических сетей и APRP. Технология APRP служит основой для реализации нечеткого поиска, устойчивого к ошибкам, содержащимся в документах и терминах запроса.

Семантические сети, применяемые в ERW, отражают синтаксис, морфологию и семантику ЕЯ, предоставляя в распоряжение пользователей БЗ для ведения интеллектуального поиска информации. Например, англоязыч-

ная версия сети охватывает около 400 тыс. смысловых значений слов и свыше 1,6 млн связей между ними.

ERW включает три текстовых сервера:

- сервер семантики и распознавания образов (ERW Semantic and Pattern Server);
- web-сервер (ERW Web Server);
- сервер профилирования (ERW Profiling Server).

Первый сервер обеспечивает поиск по значениям слов и по шаблонам, поиск по запросам на ЕЯ, нечеткий, статистический и двоичный виды поиска. Таким образом, он объединяет технологию APRP и традиционные методы поиска текстовых документов.

Web-сервер поддерживает взаимодействие с широким набором приложений, работающих в Internet и Intranet. При интеграции с реляционными БД он позволяет значительно ускорить обработку потока запросов.

Сервер профилирования предназначен для фильтрации информации, поступающей в систему в реальном времени.

Функциональные возможности ERW расширяют следующие дополнительные модули.

Сервер рубрикации ERW распределяет поступающие документы по тематическим рубрикам в соответствии с ранее введенными запросами. Один документ может быть отнесен к нескольким рубрикам. Рубрикатор может использоваться для ограничения области поиска и определения логической структуры хранилища документов.

Модуль *ERW Internet Spider* обеспечивает извлечение текстовой информации из указанных узлов Internet. Полученная информация автоматически индексируется. При конфигурировании ERW Internet Spider задаются такие параметры, как тип извлекаемых документов, имена доменов и каталогов, глубина, широта и частота сканирования.

Модуль *ERW FileRoom* предназначен для работы с бумажными архивами. В ERW совместно хранятся сканированные образы документов и текстовые файлы, содержащие результаты их оптического распознавания. Документам приписываются учетные карточки. Структура электронного архива отражает структуру бумажного и включает виртуальные шкафы, ящики и папки. Средства нечеткого поиска облегчают работу с информацией, полученной в результате оптического распознавания бумажных документов.

Модуль *Multicosm Refindment* динамически формирует горизонтальные гиперссылки, отражающие смысловые связи между документами, найденными по поисковому запросу. Это облегчает анализ результатов поиска.

Модуль *ERW CDExpress Toolkit* представляет собой комплекс программных средств для создания переносимых баз документов на компакт-дисках. Записываемые с помощью него компакт-диски содержат архив до-

кументов в формате ERW (включая мультимедийную информацию), снабженный поисковым web-интерфейсом.

Оценим основные *интегральные характеристики ERW.*

1. Система масштабируема по объему информационного массива. Экспериментальная оценка показывает логарифмический рост времени поиска при увеличении объема информации. ERW позволяет эффективно работать с архивами, объем которых превышает сотни гигабайт.

2. ERW поддерживает более десятка серверных платформ, может функционировать на базе многопроцессорных и многосерверных конфигураций, обеспечивая возможность эффективно распараллеливать работу.

3. ERW поддерживает более 200 входных форматов документов и позволяет подключать пользовательские конвертеры.

4. ERW имеет развитую систему защиты информации (контроль доступа на уровне отдельных документов, передача данных в зашифрованном виде). Сведения о пользователях и правах доступа могут наследоваться из источников, откуда получены документы.

5. Логический поиск отличается богатым набором команд и возможностей: логические операторы, операторы ограничения расстояния между словами и порядка следования слов, операторы нечеткого и семантического расширения значений слов, операторы поиска по диапазонам чисел и дат, поддержка XML и др.

6. Смысловой поиск учитывает морфологию и семантику ЕЯ. Семантическая сеть представлена в виде орграфа, отражающего взвешенные связи между понятиями. Ее использование позволяет расширять поисковые запросы и ранжировать найденные документы по степени их соответствия запросу.

7. Русскоязычный семантический сервер ERW представляет собой набор программных средств и информационных ресурсов для полнотекстового поиска с учетом специфики русского языка. Библиотека морфологического анализа включает словарь объемом 240 тыс. словарных статей. Семантическая сеть русского языка содержит около 90 тыс. слов и словосочетаний и более 350 тыс. связей между ними. Пользователь может пополнять словари, применять одновременно несколько словарей и семантических сетей.

8. Механизм нечеткого поиска, реализованный на базе технологии APRP, позволяет искать документы, исходя не из точного совпадения слов документа и запроса, а меры их семантической близости. Это исключает необходимость выполнения трудоемких операций проверки орфографии и исправления ошибок после работы OCR-систем. Данный подход лежит в основе технологий ERW для поиска любой цифровой информации – текстов, изображений, звуков и видео.

9. ERW обладает открытой архитектурой, позволяющей разработчикам модифицировать программные компоненты ERW вплоть до ядра поисковой системы.

3.6.3. Пакет NeurOK Semantic Suite

Пакет NeurOK Semantic Suite* представляет собой комплекс программных средств, реализующих интеллектуальные технологии обработки текстов на ЕЯ. Компоненты пакета обеспечивают:

- автоматическую рубрикацию (классификацию) документов;
- автоматическое и диалоговое построение каталогов для рубрикации;
- автоматическое реферирование и аннотирование;
- разнообразные виды поиска в хранилищах документов;
- автоматический мониторинг источников информации (web-сайтов, служб новостей);
- персонализацию информационных потоков и служб;
- структуризацию и визуализацию семантики массива документов.

Интеллектуальные технологии NeurOK Semantic Suite основаны на извлечении знаний из текстов на ЕЯ и оперировании соответствующими семантическими представлениями. Это позволяет перейти от поиска и доступа к документам по словам (терминам) к поиску и доступу к документам по смыслу.

Смысл слова определяется совокупностью его связей с другими словами, задающими контекст, в котором употребляется данное слово. Таким образом, семантика фиксирует ассоциативные отношения между словами ЕЯ, отражающие понятийную структуру ПрО.

В рамках NeurOK Semantic Suite смысл текста, его фрагмента или отдельного термина представляется комбинацией *семантических категорий* (укрупненных понятий), каждая из которых характеризуется определенным набором терминов. Число таких категорий существенно меньше числа слов ЕЯ, поэтому переход от лексических к семантическим описаниям документов обеспечивает значительное сжатие информации.

Построение системы семантических категорий и распределение по ним слов, используемых в массиве документов, выполняется с помощью методов *машинного обучения семантике ЕЯ.* Знания, выявляемые в ходе обучения, отражают статистику совместного употребления слов. Формируемая система семантических категорий представляет собой внутренний тезаурус NeurOK Semantic Suite, применяемый его компонентами для распознавания смысла слов и текстов. В рассматриваемом пакете реализован оригинальный алгоритм обучения семантике ЕЯ, запатентованный компанией «НейрОК Интелсофт». Он основан на циклической схеме построения согласованной системы разложения слов по набору семантических категорий, отражающему статистику их совместного употребления в обучающей выборке.

* http://www.neurok.ru.

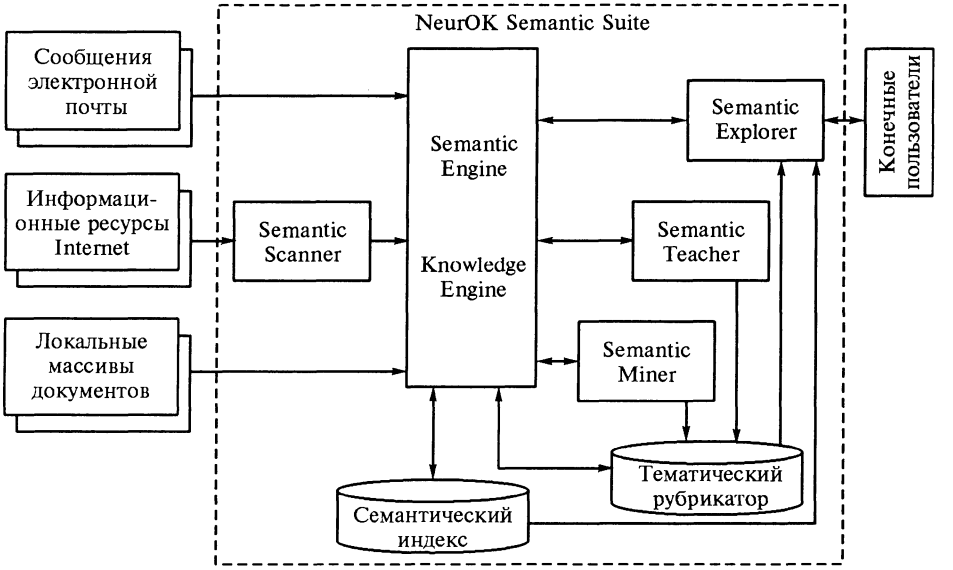


Рис. 3.22. Архитектура пакета NeurOK Semantic Suite

Оперирование семантическими представлениями позволяет учитывать морфологию языка и синонимию. Поскольку различные словоформы данного слова, равно как и слова-синонимы, как правило, употребляются в одном и том же контексте, их семантические образы совпадают (или почти совпадают). Благодаря этому семантический поиск позволяет находить релевантные документы вне зависимости от того, в какой форме присутствует в них слово из запроса, а также документы, содержащие синонимы этого слова.

Архитектура NeurOK Semantic Suite изображена на рис. 3.22. Функциональным ядром пакета служат компоненты *Semantic Engine* и *Knowledge Engine*. Они обеспечивают индексирование, классификацию и аннотирование документов, навигацию и поиск в архивах. Данные компоненты предоставляют сервисы для прочих программных систем, входящих в пакет. Они интегрируются с основными реляционными СУБД (Oracle, SQL Server, InterBase, MySQL и др.), расширяя их возможности функциями обработки текстовой информации.

Основой для поиска документов по смыслу служит *семантический индекс*. Он строится с помощью технологии ассоциативного кластерного индекса *NeurOK CARE* (Content Addressed Retrieval Engine), предназначенной для организации доступа к неструктурированной информации (текстам, изображениям, звуку, видео и т. д.). Такая информация представляется большим числом однородных признаков: слов, пикселей, значений амплитуды и др. Поиск документов, содержащих неструктурированные данные,

осуществляется по образцу, фиксирующему определенную совокупность признаков. При этом используются методы распознавания образов и модели ИНС*.

Технология NeurOK CARE является универсальной платформой для реализации средств обработки неструктурированной информации. Она открыта для любых типов данных, для которых можно описать правила обработки.

Кластерный индекс предусматривает хранение данных по принципу семантической близости. Содержательно подобные документы объединяются в один кластер. Это позволяет проводить поиск в два этапа. На первом отбираются релевантные кластеры, на втором выполняется анализ и ранжирование входящих в них документов. Исключение из рассмотрения нерелевантных кластеров существенно ускоряет поиск.

В Semantic Engine реализованы *семь видов поиска.*

1. Атрибутивный поиск (SQL-запросы направляются на выполнение внешней СУБД, сопряженной с Semantic Engine).

2. Поиск по ключевым словам (запросом служит логическая формула, составленная из ключевых слов).

3. Лексический поиск (релевантность документа пропорциональна количеству содержащихся в нем терминов, указанных в запросе).

4. Ассоциативный поиск, учитывающий не только термины, непосредственно входящие в запрос, но и другие термины, ассоциирующиеся с его семантикой.

5. Поиск ассоциаций, т. е. слов, ассоциирующихся с запросом. Данный вид поиска используется для ассоциативного расширения и уточнения запроса в процессе диалога с пользователем.

6. Поиск документов, семантически подобных данному документу.

7. Комбинированный поиск (сочетания первых шести видов поиска). Позволяет искать документы по их атрибутам и содержанию.

Компонент Knowledge Engine представляет собой вариант Semantic Engine, дополненный средствами для работы с *иерархическим тематическим рубрикатором.* Данный компонент предназначен для автоматической рубрикации документов, а также навигации и поиска в архивах.

Каждому листу рубрикатора ставится в соответствие запрос, определяющий условия отнесения документов к соответствующей рубрике. Например, это может быть набор обязательных ключевых слов и атрибутов документов.

Система *Semantic Miner* обеспечивает автоматическое построение тематического рубрикатора, имеющего заданную глубину и количество рубрик на разных иерархических уровнях, на основе семантического индекса.

* Нейротехнологии рассматриваются в гл. 6.

Для однородного массива текстовых документов формируемая структура играет роль оглавления, отражающего взаимосвязь фигурирующих в них ключевых понятий.

Рубрикатор может быть создан или изменен экспертом вручную с помощью редактора *Semantic Teacher*. В системе предусмотрены средства автоматического определения запросов для листьев рубрикатора в виде перечней обязательных ключевых слов. Данная процедура может быть задействована при наличии обучающего примера — подготовленного ранее распределения массива документов по структуре рубрикатора.

Для мониторинга источников информации в Internet и локальных сетях и доставки контента для индексирования служит компонент *Semantic Scanner*. Он состоит из программного агента (робота) и набора драйверов для чтения документов в разных форматах. Semantic Scanner содержит средства определения расписания обхода источников информации, глубины просмотра ссылок, а также правил фильтрации и предварительной обработки контента. Система имеет модульную архитектуру, обеспечивающую гибкие возможности ее настройки на условия применения.

Компонентом NeurOK Semantic Suite, ориентированным на взаимодействие с конечными пользователями, является ИПС *Semantic Explorer*. В системе реализованы средства представления тематической структуры массива документов и визуализации его семантики, а также навигации и поиска документов.

Вновь поступающие документы автоматически распределяются по тематическим рубрикам. Предусмотрены два режима рубрикации. В первом для каждого листа рубрикатора проводится отбор релевантных документов независимо от их соответствия другим листьям, в результате чего один документ может быть отнесен к нескольким рубрикам. Во втором режиме документ помещается в единственную, наиболее близкую ему рубрику.

Информационно-поисковая система имеет web-интерфейс (рис. 3.23). В левой части окна отображается тематический рубрикатор, в правой — описания отобранных документов. Каждое описание содержит информацию о дате, источнике и степени релевантности документа запросу, а также краткое резюме из фраз, в которые входят термины из запроса. Под описанием располагаются четыре гиперссылки:

- more like this — поиск документов, семантически подобных данному документу;
- full text — вызов полного текста документа;
- auto annotation — автоматическое построение аннотации (с выделением наиболее значимых фраз);
- more from... — отбор других документов из указанной рубрики.

Рис. 3.23. ИПС Semantic Explorer 3.0

В верхней части окна Semantic Explorer размещается область для ввода запроса. В системе реализована процедура итеративного уточнения запроса в ходе диалога с пользователем. Результатом выполнения каждого запроса является не только множество отобранных документов, но и совокупность слов, предлагаемых пользователю для уточнения его контекста. Эти слова выводятся в поле «context hints». Дополнительно в запросе могут быть представлены атрибуты искомых документов (например, время создания) и ограничения, связанные с их распределением по тематическим рубрикам. Формируемые подобным образом запросы более точно и емко отражают интересы пользователей, что способствует существенному повышению точности поиска и уменьшению информационного шума.

В различных версиях Semantic Explorer реализованы разные способы *визуализации семантики массива документов.* В частности, использованы представления на основе самоорганизующихся карт Кохонена [93], а также метафор «звездного неба» и «рыбьего глаза».

В рамках метафоры «звездного неба» отобранным документам соответствуют звезды на панораме ночного неба, причем их яркость пропорциональна степени релевантности запросу. Звезды, обозначающие семантиче-

ски близкие документы, располагаются рядом. Кластеры, релевантные запросу, отображаются в виде скоплений звезд (млечного пути).

Представление на основе метафоры «рыбьего глаза» имеет форму круга. В его центре размещается образ кластера, наиболее близкого к тематике запроса. Вокруг него выводятся образы кластеров, связанных с кластером, попавшим в центр рассмотрения. Прочие кластеры, имеющие слабое отношение к запросу, располагаются на периферии круга.

Визуальные представления семантики в Semantic Explorer являются интерактивными, т. е. позволяют вести навигацию по семантической структуре массива документов.

Отметим, что разработчики NeurOK Semantic Suite позиционируют свой продукт на рынке как платформу для создания систем управления знаниями. Соответствующая проблематика излагается в гл. 7.

Основные выводы

1. Коммерческие интеллектуальные программные продукты для обработки текстов входят в пору зрелости, становятся привычным инструментарием для широкого круга пользователей. Их применение приносит значительный экономический эффект.

2. Опыт практического использования таких продуктов показывает, что они должны базироваться как на традиционных, так и на новых (интеллектуальных) технологиях анализа текстовой информации. Их новые возможности обеспечиваются за счет реализации моделей и методов ИИ: семантических сетей, ИНС, методов нечеткого семантического поиска и др.

3. Интеллектуальные средства обработки текстов должны интегрироваться с современными офисными приложениями и СУБД.

4. Создание подобных продуктов связано со значительными затратами на разработку для них ЛО и ИО, что требует привлечения высококвалифицированных лингвистов, инженеров по знаниям и программистов.

Вопросы для самопроверки

1. Какие основные возможности предоставляет пользователю Text Analyst?
2. Какие принципы и механизмы обработки текста используются в Text Analyst?
3. Как в Text Analyst реализован принцип ассоциативности?
4. Назовите базовые словари и подсловари Text Analyst.
5. Как в Text Analyst используется модель семантической сети?
6. Каким образом определяются смысловые веса понятий в Text Analyst?
7. Что понимается в Text Analyst под тематической структурой текста?
8. Опишите механизм автоматического реферирования, реализованный в Text Analyst.
9. Как с помощью Text Analyst автоматизируется построение ГТ?
10. Охарактеризуйте функцию смыслового поиска в Text Analyst.

11. Для решения каких задач предназначена промышленная ИПС Excalibur RetrievalWare?
12. Назовите основные модули Excalibur RetrievalWare и их назначение.
13. Какие методы поиска информации реализованы в Excalibur RetrievalWare, и на каких моделях они базируются?
14. Каковы принципы нечеткого и семантического поиска в Excalibur RetrievalWare?
15. Для чего и как используются в Excalibur RetrievalWare ИНС?
16. Оцените основные интегральные характеристики Excalibur RetrievalWare.
17. Каково назначение пакета NeurOK Semantic Suite?
18. Как в NeurOK Semantic Suite передается смысл слов и фрагментов текста?
19. Назовите основные компоненты NeurOK Semantic Suite и их назначение.
20. Каким образом в NeurOK Semantic Suite формируется система семантических категорий?
21. Какие виды поиска реализованы в Semantic Engine?
22. Как строится и используется в Knowledge Engine иерархический тематический рубрикатор?
23. Каковы функции ИПС Semantic Explorer?
24. Охарактеризуйте способы визуализации семантики массива документов в Semantic Explorer.

Не существует ответов, есть лишь ссылки.

Библиотечный закон Винера

4. Метаданные для информационных ресурсов

Существенная черта развития Internet — переход от документов, читаемых компьютером (machine readable), к документам, понимаемым компьютером (machine understandable). Решение большинства задач систематизации и понимания компьютером документов связано с использованием метаданных. В главе отражено современное состояние работ в области метаданных для информационных ресурсов. Главное внимание уделено роли метаданных в обеспечении интеллектуализации WWW. Охарактеризованы универсальная система метаданных Дублинское ядро и модель RDF. Рассмотрены концепция семантического web и ее технологическая основа — платформа XML. Содержание главы соответствует направлениям исследований в области ИИ 1.4.1, 2.2.2, 2.3.1, 3.1.4 и 4.2.

4.1. Системы и модели метаданных

Мы должны стремиться не к тому, чтобы нас всякий понимал, а к тому, чтобы нас нельзя было не понять.

Виргилий

Метаданные (metadata) — это информация о документе, понимаемая ЭВМ, т. е. обладающая свойством внутренней интерпретируемости. В общем случае метаданные представляют собой информацию, характеризующую какую-либо другую информацию. Экземпляр метаданных для ИР выступает в качестве описания этого ИР. Подобное описание можно сравнить с записью в библиотечном каталоге. Оно отражает название ИР, его тип, назначение, объем, предметное содержание, технические особенности, сведения об авторах и разработчиках и другую информацию, которая может быть полезна при выборе ресурса. Обеспечение совместимости на уровне метаданных требует унификации их структуры, интерпретации ее компонентов и способа их представления.

Консорциум Meta Data Coalition определяет метаданные как описательную информацию о структуре и смысле данных, а также приложений и процессов, которые манипулируют данными. Метаданные могут характеризовать

сущности, относящиеся не только к виртуальному (информационному) пространству, но и к реальному миру (персоналии, организации, события и др.). Сказанное представляется важным, поскольку описания ИР могут содержать сведения об их создателях, владельцах и распространителях (физических и юридических лицах), а также событиях, в которых они фигурируют (конференции, семинары, симпозиумы, учебные и научные мероприятия и т. п.).

Система метаданных выступает в качестве центрального звена любой ИС. Метаданные могут быть как частью ИР, так и храниться отдельно от него. Например, выходные сведения издания, являющиеся по отношению к нему метаданными, приводятся в самом издании и кроме того включаются в библиотечный каталог.

Как и в технологиях БД, для метаданных определяются два уровня представления:

- инфологический, фиксируемый *схемой метаданных,* которая отражает состав и структуру элементов данных (полей) в экземпляре метаданных, их семантику, типы значений (включая словари и классификаторы) и ограничения целостности;
- даталогический, фиксируемый *форматом метаданных,* который отражает способ представления (кодирования) информации.

К числу *основных требований к системе метаданных* относятся [97]:

- универсальность в рамках установленного понимания ИР как объекта систематизации;
- структурированность и формализованность метаданных, необходимые для их автоматической обработки;
- достаточная выразительность для обеспечения эффективного решения задач, требующих наличия метаданных;
- совместимость с международными стандартами и протоколами в области метаданных и информационного поиска (создание условий для интероперабельности);
- возможность задания ограничений целостности, отражающих взаимосвязи полей описания ИР;
- обеспечение возможности хранения метаданных как совместно с ИР, так и отдельно от него;
- возможность представления в метаданных сведений о создателях, правообладателях и распространителях ИР, а также отношений между ИР.

Метаданные об ИР формируются и используются в различных системах и сервисах: электронных библиотеках; информационных порталах и web-сайтах; системах регистрации ИР; службах публикации ИР; хранилищах (депозитариях) ИР; хранилищах метаданных об ИР; системах разработки контента (авторском инструментарии); системах управления контентом; системах сертификации ИР и пр.

На основе системы метаданных реализуются базовые технологические процессы в электронных библиотеках: навигация по каталогу ИР; поиск ИР; ввод и организация хранения ИР, а также исключение ИР из хранилища; управление правами доступа к ИР, включая защиту авторских прав, организацию платы за пользование ИР и др.

В настоящее время в электронных библиотеках принято выделять две основные информационные составляющие:

1) собственно база (массив) ИР;

2) хранящаяся отдельно либо выделенная функционально база метаданных для этих ИР.

Между названными составляющими существует взаимно однозначное соответствие, на основе которого организуются процессы информационного поиска, что требует обязательного формализованного представления метаданных, т. е. разработки ***модели метаданных.***

Одной из наиболее перспективных моделей метаданных на сегодняшний день является модель *RDF (Resource Description Framework),* разработанная консорциумом W3C. Она определяет основные принципы представления и обработки метаданных и обеспечивает функциональную совместимость web-приложений, обменивающихся такой информацией. В RDF использованы принципы объектно-ориентированного моделирования, элементы языков HTML, SGML и XML. Синтаксис метаданных в RDF описывается на основе языка XML. Несмотря на то, что RDF была разработана в расчете на XML-платформу, она не зависит от XML. Данная модель позволяет представлять семантическую структуру XML-документов и выражать смысл этих и иных ресурсов WWW.

Описание семантики одного или нескольких ИР средствами RDF называется *RDF-спецификацией.* Базовыми категориями такого описания являются ИР (субъект), свойство (предикат) и значение (объект). Упрощенная структура RDF-спецификации изображена на рис. 4.1. В рамках нее для каждого ИР указываются необязательная ссылка на него (URI) и множество пар (свойство, значение). Значение свойства представляется либо текстом, либо ссылкой на другой ИР, либо вложенным описанием другого ИР. Значение свойства, выраженное ссылкой или вложенным описанием другого ИР, задает отношение между ИР.

Именование свойств осуществляется на основе механизма *пространств имен.* Таким образом, имени свойства соответствует некоторый URI. Это позволяет рассматривать свойство как самостоятельный ИР, который, в свою очередь, может быть описан средствами RDF.

Для определения информационных моделей, в соответствии с которыми должны строиться конкретные RDF-спецификации, предназначены метамодель и язык RDF Schema. В их основе лежат принципы объектно-ориентированного моделирования.

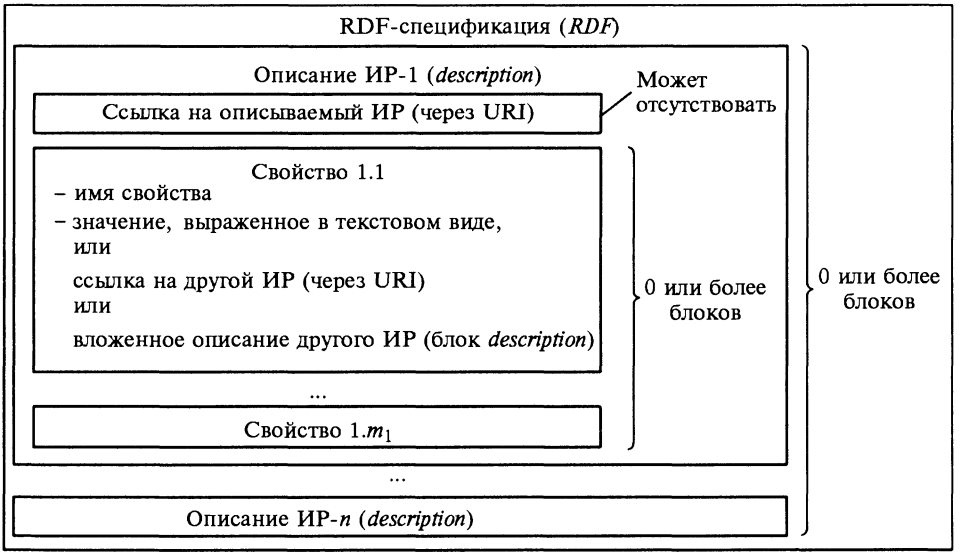


Рис. 4.1. Упрощенная структура RDF-спецификации

По назначению выделяют четыре основных *вида метаданных*:

- описательные (библиографические описания ИР и описания их семантики в виде рефератов и аннотаций);
- структурные (формат, объем и структура ИР);
- административные (правообладатели, права на доступ и коррекцию ИР, сведения о пользователях, платежах и т. п.);
- идентифицирующие, служащие для однозначного представления описываемых объектов для внешнего мира и приложений.

К настоящему времени в мире создано множество систем метаданных, обладающих разным статусом: международные, национальные и отраслевые стандарты, корпоративные спецификации, спецификации международных консорциумов и др. Следует отметить, что существуют разработки, не имеющие статуса официально утвержденных стандартов, но широко применяемые на практике и рассматриваемые в качестве стандартов «де-факто».

Перечислим некоторые системы метаданных:

- «Дублинское ядро» (инвариантный к ПрО набор наиболее общих полей описания ИР, введенный для обеспечения глобальной интероперабельности приложений, работающих с метаданными) [106];
- MARC — предназначена для описания библиотечных ресурсов (как на бумажных, так и на электронных носителях) [105];
- GILS — предназначена для описания любых видов ИР, расширяющая MARC и базирующаяся на протоколе Z39.50;

- ONIX — предназначена для описания товаров в системах электронной коммерции;
- LOM — предназначена для описания образовательных ИР;
- IAFA/WHOIS++ — предназначена для описания сетевых ИР;
- UDDI — предназначена для описания web-сервисов;
- INDECS — ориентирована на системы электронной коммерции и содержащая элементы для управления правами на цифровые объекты;
- EAD — предназначена для описания архивных материалов;
- GEM — расширение «Дублинского ядра» для описания образовательных ИР;
- МЕКОФ — международный коммуникативный формат, выступающий в качестве альтернативы MARC [99–101];
- формат описания БД и машиночитаемых информационных массивов [102].

С точки зрения ориентации на виды ИР и сферы использования различают универсальные и специализированные системы метаданных. К универсальным системам относятся «Дублинское ядро» и GILS.

Наиболее распространенной системой метаданных является *«Дублинское ядро» (Dublin Core Metadata Element Set)*. Основные цели, которые ставились при ее создании, заключались в обеспечении:

- простоты формирования и поддержки метаданных;
- легко понимаемой (как человеком, так и компьютером) семантики;
- возможности представления метаданных на разных ЕЯ;
- расширяемость системы метаданных.

«Дублинское ядро» включает два уровня:

1) простое «Дублинское ядро» (Simple Dublin Core);

2) «Дублинское ядро» с квалификаторами (Qualified Dublin Core)

Первый уровень содержит 15 элементов данных, образующих три группы (табл. 4.1):

- Content (содержание ИР);
- Intellectual Property (интеллектуальная собственность);
- Instantiation (характеристики данного экземпляра ИР).

Состав элементов простого «Дублинского ядра» определен в стандарте ISO 15836:2003*.

На втором уровне к 15 элементам добавлены два дополнительных элемента: Audience (целевая аудитория, категория пользователей) и Rights Holder (правообладатель). Кроме того, для повышения детальности и выразительности описаний на этом уровне вводятся и используются *квалифика-*

* ISO 15836:2003. Information and documentation — The Dublin Core metadata element set.

торы, уточняющие семантику элементов данных и специфицирующие источники и способы представления их значений. Например, с элементом Description связаны два квалификатора: Table Of Contents (оглавление) и Abstract (аннотация). Даты рекомендуется представлять в формате, установленном в стандарте ISO 8601:2000*. Для описания персон и организаций может использоваться система метаданных для электронных бизнесс-карт vCard [97, 107—109], основанная на рекомендациях серии X.500 для служб распределенного каталога [110].

Все элементы «Дублинского ядра» являются необязательными и могут повторяться. Порядок их следования в описании ИР значения не имеет.

Таблица 4.1

Группы элементов данных		
Content	Intellectual Property	Instantiation
Title — Заглавие ИР	Creator — Создатель ИР	Date — Дата
Subject — Предметная область	Publisher — Издатель ИР	Format — Формат ИР
Description — Описание ИР	Contributor — Лицо, внесшее вклад в создание или развитие ИР (соисполнитель)	Identifier — Идентификатор ИР
Type — Тип ИР	Rights — Права на ИР	Language — Язык ИР
Source — Источник ИР		
Relation — Отношение (ссылка на другой ИР)		
Coverage — Охват ИР (пространственный и временной)		

Для определения каждого элемента (поля) системы метаданных служит набор из 10 типовых атрибутов, фиксируемый стандартом ISO/IEC 11179 «Спецификация и стандартизация элементов данных». Перечислим эти атрибуты.

1. Имя — метка, определяющая элемент данных.
2. Идентификатор (уникальный для представляемого элемента данных).
3. Версия (элемента данных).
4. Орган регистрации — организация или лицо, наделенные полномочиями по вводу в действие элемента данных.
5. Язык, на котором дается характеристика элемента данных.

* ISO 8601:2000. Data elements and interchange formats — Information interchange — Representation of dates and times.

6. Определение — содержание элемента данных.

7. Обязательность — признак, отражающий обязательный или факультативный статус элемента данных в рамках экземпляра метаданных.

8. Тип данных, которому соответствуют значения элемента данных.

9. Максимальная распространенность — признак, отражающий допустимость наличия в экземпляре метаданных нескольких экземпляров элемента (т.е. допустимость указания нескольких его значений).

10. Комментарий по применению элемента данных.

Атрибуты «тип данных» и «комментарий» определяют представление данных.

Пример обобщенного абстрактного описания элемента данных:

- имя — «Предметная область»;
- идентификатор — «Subject»;
- язык — «русский»;
- определение — «ПрО, к которой относится содержание ресурса»;
- обязательность — «обязательный»;
- тип данных — «строка, максимальная длина — 200 символов»;
- максимальная распространенность — «до 10 экземпляров»;
- комментарий — «Значение выражается ключевым словом, ключевой фразой или классификационным кодом, характеризующим тему ИР и выбираемым из УДК, ГРНТИ и др.».

Сформулируем ряд рекомендаций по формированию описаний ИР на основе системы метаданных «Дублинское ядро».

1. Желательно значения элементов данных выбирать из распространенных словарей (словников, тезаурусов, классификаторов). Например, источником значений элемента Coverage может служить Тезаурус географических наименований (Thesaurus of Geographic Names*).

2. При описании темы (Subject) необходимо применять тезаурусы для представления характеристик ПрО, с которыми ассоциируется ИР (предметные рубрикаторы и классификаторы). Обычно используются два типа тезаурусов: предметные и функциональные.

Предметный тезаурус систематизирует понятия ПрО и позволяет охарактеризовать содержание ИР, т. е. ответить на вопрос «О чем этот ИР?». Функциональный тезаурус позволяет описать роль ИР в человеческой деятельности, т. е. ответить на вопрос «Для чего нужен этот ИР?».

3. Для выражения типа ИР (Type) может использоваться следующий минимальный набор обобщенных значений:

- text (текст);
- image (изображение);

* http://www.getty.edu/research/conducting_research/vocabularies/tgn.

- sound (звук);
- dataset (набор данных);
- software (программа);
- interactive (интерактивная система);
- event (событие);
- physical object (физический объект).

Каждое из перечисленных значений может служить корнем иерархического классификатора, детализирующего соответствующий обобщенный тип. Например, для типа «event» подчиненными значениями могут быть:

- конференция;
- семинар;
- круглый стол;
- выставка;
- проект.

4. Для элемента «Формат» (Format) рекомендуется выбирать значения из множества типов контента MIME (Multipurpose Internet Mail Extensions) [112, 113]. Например:

- text/xml — документ на XML;
- text/plain — текст без форматирования и разметки;
- image/gif — изображение в формате GIF.

Возможны два способа размещения метаданных. В первом они включаются непосредственно в ИР (например, в HTML-страницу с помощью тегов <META>)*. Во втором они хранятся отдельно от ИР. В этом случае метаданные предпочтительно хранить и передавать в формате, реализованном на базе XML. Обмен метаданными сводится к пересылке XML-файлов или ссылок на эти файлы.

Еще одна универсальная система метаданных — GILS — лежит в основе формата метаданных Государственного регистра баз и банков данных РФ. Предполагается, что этот формат станет ядром навигационной системы всех государственных ИР РФ. Цель GILS — обеспечить организациям и гражданам поиск ИР, созданных на средства налогоплательщиков и представленных на любых носителях и языках. GILS позволяет описывать печатные и электронные издания, БД, персоны, организации, события, собрания (коллекции), артефакты и т. д. Система метаданных GILS поддерживает гиперссылки для доступа к ИР, связанным с описываемым ИР и размещенным Internet. Поиск на основе GILS успешно работает в сочетании с семантикой, представленной в модели «Дублинское ядро».

* Способ предоставления метаданных в пределах тегов <META> описан в спецификации RFC 2731:1999 (Encoding Dublin Core Metadata in HTML; http://www.ietf.org/rfc/rfc 2731.txt).

В силу высокой общности система метаданных «Дублинское ядро» не позволяет отражать специфичные характеристики некоторых видов ИР. Для описания таких ИР применяются специализированные системы метаданных или расширения «Дублинское ядро» на основе квалификаторов. В частности, для описания образовательных ИР предназначена система метаданных *LOM (Learning Object Metadata)*. Наряду с общими атрибутами ИР она содержит группу образовательных характеристик, к которым относятся сложность, контактное время, тип и уровень интерактивности, семантическая емкость, возрастной диапазон пользователей ИР и др. Метаданные, соответствующие модели «Дублинское ядро», отображаются в LOM.

4.2. Семантический web и платформа XML

Internet-технологии — это технологии, позволяющие наступать на грабли, находящиеся на другой стороне земного шара.

Программистский фольклор

Недостатки и ограничения технологий Internet первого поколения (web-1) привели к разработке консорциумом W3C ***концепции «семантической паутины»*** (Semantic Web или web-2). Она направлена на интеллектуализацию WWW и базируется на следующих основных компонентах:

- активном использовании метаданных;
- метаязыке XML;
- онтологическом подходе, позволяющем описывать термины и отношения между ними (см. § 5.4);
- модели RDF, устанавливающей способ представления значений, определенных в онтологии.

В Semantic Web также применяются:

- универсальные идентификаторы ресурсов;
- системы обработки правил логического вывода;
- стандартные протоколы Internet.

Цель реализации Semantic Web состоит в преодолении ограничений технологий web-1 с сохранением их достоинств. К числу основных положительных черт web-1 можно отнести [119]:

- открытый характер Internet — к сети можно подключиться с помощью любого стандартного оборудования и свободно распространяемых программных средств;
- демократическая организация — использование Internet не требует существенных финансовых затрат и каких-либо административных решений;

• эффективная как для пользователей, так и для разработчиков приложений клиент-серверная архитектура WWW;

• простота языка разметки HTML, возможность представления с помощью него не только гипертекстовых, но и гипермедийных данных, наличие множества HTML-редакторов и др.

Однако ряд свойств web-1 сдерживает дальнейшее развитие и эффективное использование WWW. В первую очередь, речь идет о языке HTML, который ориентирован не на логическую, а на форматную разметку контента, отражающую способ его представления на экране. Язык HTML имеет слабые средства поддержки метаданных, описывающих структурные и семантические свойства web-страниц. Еще один его недостаток связан с идентификацией ИР по их адресам в WWW с помощью URL. К HTML-ресурсам возможен только навигационный доступ по гиперссылкам. Доступ по содержанию обеспечивают специальные средства — поисковые машины.

Около 70 % ИР Internet явно не представлены в web-1, т. е. недоступны для автоматической обработки поисковыми машинами. Подобные ресурсы образуют так называемый *скрытый* или *глубинный web* (*deep web*) — это БД, интегрированные в web-сайты, архивы, мультимедийные файлы, а также многочисленные документы в форматах PDF, DOC, RTF, PostScript и др. Отсутствие эффективных методов доступа к таким ИР и описывающим их метаданным затрудняет использование web-1.

Неразвитость HTML в части поддержки метаданных не позволяет верифицировать ИР, ограничивает поисковые возможности web-1 навигационным и контекстным методами поиска. Поэтому поисковые машины web-1 характеризуются высоким уровнем поискового шума.

Основой web-2 служит расширяемый язык разметки XML. Возникла новая технологическая платформа web-2 — ***платформа XML.*** Под ней понимается совокупность взаимосвязанных и согласованных стандартов и спецификаций, имеющих общее функциональное назначение.

За последние годы создано ядро платформы XML. В его основе лежат стандарты XML, понятие XML-документа, способы представления метаданных с помощью XML, более общий по сравнению с URL механизм идентификации ресурсов URI, протоколы передачи XML-данных XMLP и SOAP.

Важно отметить, что платформа XML обеспечивает совместимость web-2 с используемыми в настоящее время технологиями web-1.

Единицей доступа к ИР web-2 является XML-документ. XML обеспечивает *отделение содержательных данных документа (контента) от информации, описывающей его представление на экране.* С помощью XML задается логическая разметка документа в соответствии с некоторым шаблоном, называемым *моделью документа.* Модель определяется с помощью

языков DTD или XML Schema. В первом случае модель часто называют описанием типа документа, во втором — схемой документа.

Модель документа может отсутствовать. XML позволяет представлять как слабоструктурированные данные (документы без модели), так и структурированные данные (документы, ссылающиеся на модели).

Наличие модели позволяет *автоматически верифицировать XML-документ,* т. е. проверять его структуру и содержимое на соответствие ей. Выделяются два уровня верификации:

- проверка соответствия базовому синтаксису XML;
- проверка соответствия модели.

Верификация на первом уровне применима по отношению к любому XML-документу и не использует модель. Успешно прошедший ее XML-документ называется правильным (корректным). Для верификации на втором уровне требуется модель. XML-документ, соответствующий ей, называется допустимым.

Модель документа фактически определяет *приложение XML,* т. е. язык разметки, построенный на основе XML. Таким образом, с помощью XML могут описываться произвольные данные, в том числе в БД. В настоящее время активно развивается новый класс СУБД, использующих XML-платформу. Такие СУБД называются XML-ориентированными.

В табл. 4.2 приведен перечень стандартов и спецификаций, составляющих ядро платформы XML. Их краткая аннотация содержится в [119, 121], а полные описания опубликованы на сайте W3C (http://www.w3.org).

Таблица 4.2

Группа стандартов	Краткое название стандарта (спецификации)	Назначение
Фундаментальные стандарты платформы XML	XML	Спецификация языка XML
	XML Information Set	Спецификация базовых типов данных
	Namespaces in XML	Спецификация механизмов использования пространств имен
Идентификация элементов данных в XML-документах и описание связей между ними	XPointer	XML Pointer Language — язык указателей XML. Определяет правила построения и использования указателей
	XLink	XML Linking Language — язык связывания XML. Определяет правила задания и интерпретации ссылок
	XInclude	XML Inclusions. Определяет механизм включения одного XML-документа в другой

Продолжение таблицы 4.2

Группа стандартов	Краткое название стандарта (спецификации)	Назначение
Описание представления и правил трансформации XML-документов	XSL	Extensible Stylesheet Language — расширяемый язык стилей. Служит для описания представления XML-документов
	XSLT	Extensible Stylesheet Language for Transformation — расширяемый язык стилей для трансформации. Служит для описания правил преобразования XML-документов
	CSS	Cascading Style Sheet — язык описания каскадных таблиц стилей. Служит для описания представления XML-документов
Модели документов и представление метаданных	DTD	Document Type Definition — определение типа документа. Язык описания модели XML-документа
	XML Schema	Язык описания модели XML-документа
	RDF	Модель и язык представления метаданных, являющийся приложением XML
	RDF Schema	Метамодель и язык описания моделей метаданных, базирующихся на RDF
Языки запросов	XQuery (XQL)	XML Query Language — язык запросов для XML-данных
	XPath	XML Path Language — язык описания путей для доступа к фрагментам XML-документов. Используется в XPointer, XSLT и XQuery
Передача XML-данных	XMLP	XML Protocol — протокол передачи XML-данных. Определяет механизмы пакетного обмена XML-данными и способы удаленного вызова процедур в WWW
	SOAP	Simple Object Application Protocol — упрощенный прикладной протокол передачи XML-данных
	XForms	XML Forms. Определяет механизм передачи содержимого web-форм (например, запросов и результатов их выполнения) для XML-данных

Окончание таблицы 4.2

Группа стандартов	Краткое название стандарта (спецификации)	Назначение
Интерфейсы прикладного программирования	DOM	Document Object Model — объектная модель документа. Определяет объектную модель ИР (XML-документа) и API для его обработки
	SAX	Simple API for XML. API для синтаксического анализа XML-документов, основанный на событиях
Обеспечение совместимости с web-1	XHTML	Extensible HyperText Markup Language — расширяемый HTML. Определяет модель XML-документа, соответствующую HTML 4, благодаря чему HTML 4 может интерпретироваться как приложение XML
	XML Base	Определяет способ представления базовых адресов в гиперссылках. Используется в XLink для поддержки соответствующих типов гиперссылок, фигурирующих в HTML-ресурсах
Обеспечение информационной безопасности	XML-Signature	XML-Signature Syntax and Processing. Спецификация электронной цифровой подписи для XML-документов
	XML Encryption	XML Encryption Syntax and Processing. Спецификация шифрования XML-данных
Идентификация ИР	URI	Uniform Resource Identifier — унифицированный идентификатор ресурса. Обобщенный формат идентификатора ИР. Его разновидностями являются URL и URN
	URL	Uniform Resource Locator — унифицированный указатель ресурса. Формат идентификатора ИР на основе его адреса в сети
	URN	Uniform Resource Name — унифицированное имя ресурса. Формат идентификатора, не зависящего от сетевого адреса ИР
Представление научной информации	MathML	Mathematical Markup Language — язык математической разметки. Является приложением XML и позволяет описывать математические выражения
	CML	Chemical Markup Language — язык химической разметки. Является приложением XML и позволяет описывать химические формулы и уравнения

Все стандарты и спецификации платформы XML синтаксически едины: компоненты платформы, расширяющие функциональность XML, используют синтаксис этого языка, т. е. являются приложениями XML. Язык HTML также может быть определен как приложение XML, поэтому переход к платформе XML не грозит потерей ИР, накопленных в WWW.

Основные выводы

1. Актуальность формирования и применения метаданных обусловлена стремительным развитием Internet и электронных библиотек, переходом к использованию ИР, понимаемых компьютером. При этом метаданные призваны обеспечить внутреннюю интерпретируемость электронных документов и тем самым качественно повысить эффективность поиска ИР.

2. Наиболее распространенный универсальный набор элементов метаданных — Дублинское ядро — в настоящее время стандартизован.

3. Формализованное представление метаданных может быть определено на основе модели RDF. Синтаксис метаданных в RDF описывается с помощью XML.

4. Технологии, использующие метаданные, стимулируют разработку программ автоматического извлечения знаний из текстов. Важной задачей является создание средств автоматического формирования метаданных по содержимому ИР. Например, существуют программы, извлекающие метаданные из тегов <META> HTML-страниц.

5. Недостатки и ограничения технологий Internet первого поколения привели к разработке консорциумом W3C концепции семантического web, основанного на расширяемом языке разметки XML. Возникла новая технологическая платформа web-2 — платформа XML.

6. Расширяемость XML, отделение содержательных данных документа (контента) от информации, описывающей его представление, наличие средств моделирования и верификации документов, поддержка метаданных, использование обобщенного механизма идентификации ИР и другие свойства платформы XML обусловливают новые возможности web-2, а также совместимость семантического web с действующими технологиями web-1.

Вопросы для самопроверки

1. Охарактеризуйте понятие «метаданные».
2. Что понимается под системой метаданных?
3. Где и для чего используются метаданные?
4. Каковы основные требования к системе метаданных?
5. Дайте характеристику модели RDF.
6. Назовите основные виды метаданных.
7. Перечислите наиболее известные системы метаданных.

8. Перечислите элементы системы метаданных «Дублинское ядро».
9. Какие типовые атрибуты служат для определения элементов системы метаданных?
10. Где могут храниться метаданные?
11. Какие достоинства, недостатки и ограничения присущи технологиям Internet первого поколения?
12. Что такое скрытый web? Что ограничивает доступ к его ИР?
13. На что направлена концепция Semantic Web? Каковы ее основные компоненты?
14. Что понимается под технологической платформой web-2? Каково ее ядро?
15. Что такое модель документа? Зачем она нужна?
16. Какие языки предназначены для описания моделей XML-документов?
17. Что понимается под приложением XML?
18. Назовите основные группы стандартов XML-платформы.

Историю цивилизации можно выразить в шести словах: чем больше знаешь, тем больше можешь.

Э. Абу

5. МОДЕЛИРОВАНИЕ ЗНАНИЙ О ПРЕДМЕТНЫХ ОБЛАСТЯХ КАК ОСНОВА ИНТЕЛЛЕКТУАЛЬНЫХ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Моделирование знаний о предметных областях — базовое направление ИИ. В главе представлены его методологические и технологические основы.
Изложение начинается с анализа категории знания. Рассмотрены разновидности и концептуальные свойства знаний. Сформулированы требования к представлению знаний. Приведен обзор основных классов моделей знаний. Описаны четыре модели семантических сетей.
Важной разновидностью сетевой модели знаний является онтология. Задачи интеллектуализации Internet стимулируют интенсивное развитие онтологического подхода. Его развернутая характеристика представлена на страницах главы.
В интеллектуальных автоматизированных системах модели знаний реализуются в рамках БЗ. Изложены концептуальные основы технологии БЗ: обобщенная структура БЗ, система операций для работы со знаниями, методы интеллектуальной верификации знаний (в том числе стратегии и алгоритмы разрешения противоречий), механизмы наследования в БЗ.
Содержание главы соответствует направлениям исследований в области ИИ 2 и 3.1.3.

5.1. Категория знания*

Знание — сила.
Ф. Бэкон

В основе исследований в области ИИ лежит подход, связанный со ***знаниями.*** Опора на знания — базовая парадигма ИИ. Как и многие фундаментальные научные категории (например, алгоритм, интеллект, деятельность и т. д.), понятие «знание» относится к интуитивно определяемым.

* Содержание параграфа соответствует направлению исследований в области ИИ 2.1.

В БСЭ дается следующее его толкование: «Знание — проверенный практикой результат познания действительности, верное ее отражение в сознании человека. Знания бывают житейскими, донаучными, художественными, научными (теоретическими и эмпирическими)».

Знания о некоторой ПрО представляют собой совокупность сведений об объектах этой ПрО, их существенных свойствах и связывающих их отношениях, процессах, протекающих в данной ПрО, а также методах анализа возникающих в ней ситуаций и способах разрешения ассоциируемых с ними проблем. Учитывая широту подобного толкования знаний, коротко рассмотрим прочие трактовки этой базовой категории.

В «Словаре русского языка» С.И. Ожегова* знание определяется как «постижение действительности сознанием» и «совокупность сведений, познаний в какой-нибудь области». Интерпретация знаний как «совокупности сведений, образующих целостное описание, соответствующее некоторому уровню осведомленности об описываемом вопросе, предмете, проблеме и т. д.» дана в [10]. Там же приведены толкования следующих разновидностей знаний: декларативных, прагматических, процедурных, эвристических, экспертных, знаний о ПрО.

Указанные трактовки знаний позволяют выделить две их характеристики: объектность и личностность.

Объектные знания связаны с представлением конкретного объекта или класса объектов (вещей, процессов, явлений и т. п.). *Метазнания* напротив описывают систему наиболее общих понятий, принципов и закономерностей природы, общества и мышления.

В личностном аспекте знания ассоциируются с их носителями, т. е. теми, кто ими обладает: конкретным человеком, группой людей или искусственным источником знаний (книгой, учебным курсом, видеофильмом и проч.) Носителем обезличенного или *социального знания* является социум. Поскольку личностные знания субъективны, для их адекватной интерпретации необходимо учитывать точку зрения (мировоззрение, взгляды, привычки) носителя. В случае существования множества носителей знаний важное значение имеют особенности жизненного уклада людей, специфика их профессиональной деятельности и т. п.

Для специалистов в области ИИ при анализе категории знания характерно акцентирование внимания на формально-логических аспектах рассматриваемых вопросов. Сказанное иллюстрируют трактовки знаний как формализованной информации, на которую ссылаются или используют в процессе логического вывода [9], и хранимой (в ЭВМ) информации, формализованной в соответствии с определенными структурными правилами, ко-

* *Ожегов С.И.* Словарь русского языка / Под ред. Н.Ю. Шведовой. — М.: Рус. яз., 1986.

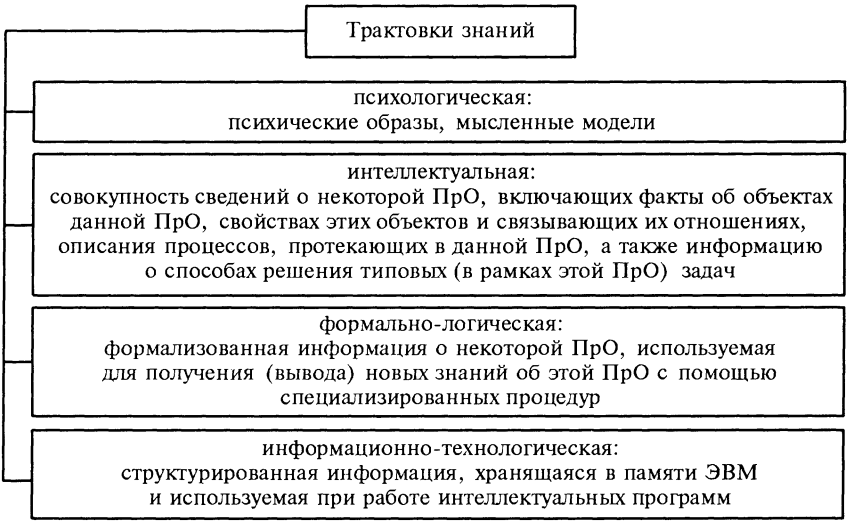


Рис. 5.1. Трактовки знаний

торую ЭВМ может автономно использовать при решении проблем по таким алгоритмам, как логические выводы [138]. Заметим, что оба определения раскрывают знания в плане их практического применения.

Несколько иное понимание знаний присуще психологам. С их точки зрения знания представляют собой психические образы — восприятия, памяти, мышления, в качестве которых выступают образы предметов и явлений внешнего мира, образы различных действий человека и т. д. Часто знания рассматриваются и как динамические мысленные модели предметов, явлений, их свойств и отношений. В психологическом плане знания существуют в формах чувственных моделей, моделей-изображений (зрительных образов) и знаковых моделей.

Упомянутые выше трактовки знаний могут быть объединены в четыре группы (или уровня): психологическую, интеллектуальную, формально-логическую и информационно-технологическую (рис. 5.1). Ясно, что на первом уровне со знаниями оперируют психологи. Второй и третий уровни соответствуют интерпретациям данной категории специалистами в области ИИ и логиками. Наконец, четвертый уровень близок к пониманию знаний прагматиками-программистами и разработчиками информационных систем.

Ряд аспектов классификации знаний иллюстрирует рис. 5.2.

Как видим, в зависимости от источника знаний выделяются *априорные* и *накапливаемые* знания. Первый вид знаний определяется и закладывается в БЗ до начала функционирования включающей эту БЗ интеллектуальной системы. Кроме того, при работе с БЗ достоверность содержащихся в

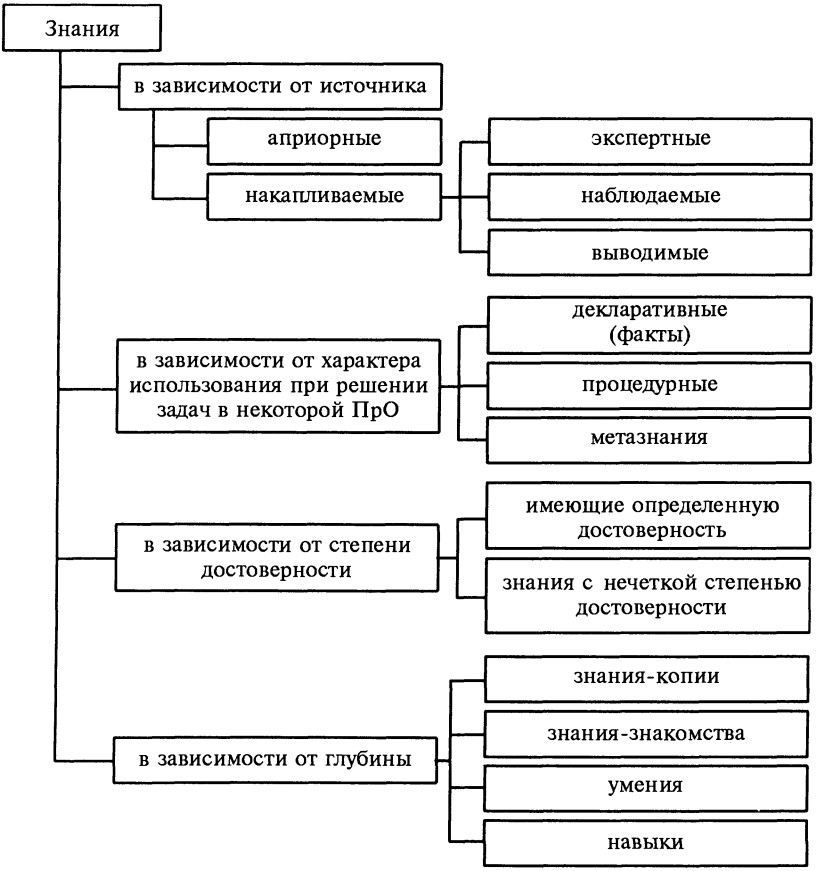


Рис. 5.2. Классификация знаний

ней априорных знаний не переоценивается. В отличие от априорных, множество накапливаемых знаний формируется в процессе использования БЗ. Источниками этих знаний могут быть эксперты, внешние искусственные «устройства-наблюдатели» (различного рода датчики, механизмы распознавания образов и т. п.), а также правила и процедуры вывода и верификации знаний, действующие в рамках интеллектуальной системы.

Широко распространенным аспектом классификации знаний является их деление на *декларативные* (факты, сведения описательного характера), *процедурные* (информация о способах решения типовых задач в некоторой ПрО) и *метазнания.* Метазнания чаще всего определяются как «знания о знаниях» и содержат общие сведения о принципах использования знаний. К уровню метазнаний также относят стратегии управления выбором и применением процедурных знаний.

В основе деления знаний в зависимости от степени их достоверности лежат так называемые *не-факторы,* присущие знаниям [131]: неполнота информации о рассматриваемом фрагменте ПрО, неточность количественных и размытость качественных оценок, неоднозначность ряда правил вывода новых знаний, несогласованность некоторых положений в БЗ и пр. Один из способов учета подобных не-факторов при формализации знаний состоит в использовании аппарата теории нечетких множеств. Примерами декларативных и процедурных знаний, имеющих определенную степень достоверности, могут служить утверждения: «Следующим днем календаря после 31 мая является 1 июня» и «Для кипячения воды при нормальном давлении требуется ее нагрев до 100 °С». Знания с нечеткой степенью достоверности содержатся в суждениях: «Завтра в Москве будет дождь» и «При игре в шахматы не следует располагать коня на краю доски».

К классу процедурных знаний с нечеткой степенью достоверности относятся *эвристики,* описывающие приемы решения задач, базирующиеся на опыте экспертов в данной ПрО. Понятия эвристических правил, приемов и методов являются ключевыми в теории творческой деятельности и креативной педагогике [122, 132, 142]. Типичным примером эвристического правила может служить приведенная выше рекомендация о расположении коня на шахматной доске.

Субъективность эвристик обусловлена отражением в них особенностей представления ПрО и анализа проблемных ситуаций в этой ПрО человеком. Отсюда же следует правдоподобный характер большинства эвристических методов. На практике эвристики широко используются в ИАС и, в частности, ЭС, причем обычно их применение направлено на оптимизацию поисковых процедур и сокращение перебора релевантных вариантов решения задач. Вместе с тем, выявление некоторой эвристической закономерности и включение соответствующих ей правил в БЗ ЭС разделяет значительное «расстояние». Для его преодоления эксперту по ПрО и инженеру по знаниям необходимо выполнить следующие четыре этапа. Во-первых, эксперт должен сформулировать (описать) эвристический прием. Если учесть, что многие эвристики используются специалистами интуитивно, т. е. на подсознательном уровне, названная задача не покажется такой уж тривиальной. Во-вторых, требуется исследовать сферу применения выделенной эвристики и указать логическое условие ее активации в БЗ. Данное условие определяет события, при которых связанное с эвристикой правило начнет выполняться. В-третьих, следует формализовать само эвристическое правило, представив его с помощью выбранного языка описания знаний. Наконец, в-четвертых, необходимо оценить степень правдоподобия закладываемой в БЗ эвристики.

С точки зрения меры возможной формализации различают три группы эвристических методов [142]: полностью формализованные — алгоритмы;

неформализованные на данном уровне развития науки — эвристмы; частично формализованные, частично неформализованные — эвроритмы. Ясно, что основное значение при разработке ИАС имеет третья группа эвристических методов.

Говоря об эвристиках, нельзя обойти стороной психологическое толкование данного понятия. В этом плане эвристики интерпретируются как методы, операционные процедуры, при помощи которых человек получает информацию, необходимую ему для создания гипотез и планов решений, когда эти последние заранее не даны. Трактовка эвристик как механизмов, регулирующих процессы мыслительного поиска, позволяет соотнести их не только с уровнем процедурных знаний, но и с метазнаниями. Так, эвристики могут рассматриваться как комплекс факторов, влияющих на выбор субъектом того или иного правдоподобного рассуждения. Как видим, четыре «среза» понимания знаний вообще (см. рис. 5.1) имеют место и в случае эвристических знаний.

Для специалистов в области информатики характерно интерпретировать категорию знаний, сопоставляя ее с другой распространенной и хорошо известной им категорией — данными. В результате выделяются шесть так называемых ***концептуальных свойств знаний*** [131]:

1) внутренняя интерпретация;
2) наличие внутренней структуры связей;
3) наличие внешней структуры связей;
4) шкалирование;
5) погружение в пространство с семантической метрикой;
6) наличие активности.

Первое из этих свойств позволяет соотнести данные, хранящиеся в памяти ЭВМ, с их смысловым содержанием. Например, пусть в оперативном запоминающем устройстве ЭВМ записано число «4». Очевидно, что этот факт сам по себе мало что говорит, так как непонятно, что конкретно обозначает число «4». По иному обстоят дела, если информация представлена выражением: «Оценка студента Иванова на экзамене 4». Поскольку оценка на экзамене — целое число, не большее 5 и не меньшее 2 (для простоты мы не учитываем искусственные оценки 0 и 1, соответствующие недопуску и неявке на экзамен), такое представление накладывает ограничения на данные, заносимые в поле оценки. Кроме того, наличие *внутренней интерпретации* обеспечивает возможность построения процедур, отвечающих «от имени ЭВМ» на вопросы человека о содержимом ее памяти. Избыточность, возникающая в результате повторения названий полей, минимизируется за счет использования таблиц, объединяющих однотипные записи. Названия полей, общие для всех записей базы, указываются в заголовке таблицы. Примерный вид такой таблицы для рассмотренного примера приведен

ниже (табл. 5.1). В настоящее время принцип внутренней интерпретации реализован в рамках реляционной модели данных и реляционных БД.

Таблица 5.1

Ф.И.О. студента	Оценки, полученные на экзаменах в ... сессию .../... учебного года			
	Математика	Физика	Информатика	...
Иванов И.И.	4	4	5	...
Петров П.П.	4	5	4	...
Сидоров С.С.	5	4	5	...
...	...	...	...	...

Следующие два концептуальных свойства знаний — *наличие внутренней и внешней структур связей* — основываются на структурном подходе к представлению ПрО, согласно которому в объекте ПрО могут быть выделены его части (элементы). Отношения между объектом-целым и его составляющими называются отношениями типа целое-часть (включение) и часть-целое (вхождение). Подчеркнем, что объекты-части следует трактовать шире, чем физические «детали» целого. В общем случае это — качества, признаки, описывающие целое, его атрибуты и т. д.

Распространение принципа деления объектов на уже выделенные компоненты целого позволяет строить многоуровневые иерархические представления. Объекты-части могут интерпретироваться независимо друг от друга, т. е. как элементы множества. Если взаимозависимость частей является существенной, ее необходимо отразить в БЗ. Поскольку отношения включения и вхождения для данной цели не подходят, на множестве объектов ПрО (как целых, так и частей) вводятся различные *семантические отношения* (родовидовые, пространственные, временные и т. д.), описывающие структуру фрагмента ПрО.

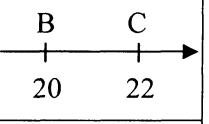
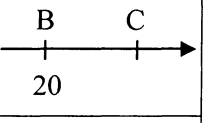
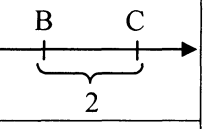
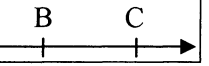
Итак, внутренняя и внешняя структуры связей отражают две стороны структурного подхода. Важность структурных представлений ПрО обусловливается их высокой психологичностью. Другими словами, принципы декомпозиции объектов ПрО и выделения системы отношений между ними базируются на подобных механизмах человеческого мышления.

Еще одно концептуальное свойство знаний — *шкалирование* — позволяет сопоставлять и упорядочивать качественно одинаковые, но различающиеся в количественном плане свойства и отношения объектов ПрО. Мера этого различия называется *интенсивностью* свойства или отношения.

Как известно, шкала или координатная ось задается тремя параметрами: точкой начала отсчета, единичным отрезком и направлением. Специфика *семантических шкал,* в частности, состоит в том, что для их определения обязательно указать только один из названных параметров: направление.

Комбинации наличия/отсутствия в определении шкалы двух других задающих ее параметров порождают четыре вида шкал (табл. 5.2). В последнем столбце табл. 5.2 изображены схемы шкал, на которые проецируются интенсивности или соотношения интенсивностей характеристики «температура (выраженная в градусах Цельсия) днем в Москве» вчерашнего и сегодняшнего дня.

Таблица 5.2

Вид шкалы	Параметры, задающие шкалу			Пример	Схема
	направление	точка начала отсчета	единичный отрезок		
Метрическая абсолютная	+	+	+	Вчера днем в Москве температура составила 20 °С. Сегодня днем в Москве температура 22 °С	
Порядковая с фиксированной точкой отсчета	+	+		Вчера днем в Москве температура составила 20 °С. Сегодня днем в Москве теплее	
Метрическая относительная	+		+	Сегодня днем в Москве температура на 2 °С выше, чем вчера	
Порядковая	+			Сегодня днем в Москве теплее, чем вчера	

Примечание. В — вчера; С — сегодня; + — наличие данного параметра в определении шкалы.

Неопределенность единичного отрезка на порядковых шкалах не следует трактовать абсолютно, поскольку в большинстве случаев на них задаются нечеткие метрики, позволяющие различать интенсивности отношений, связывающих несовпадающие точки шкал. Иллюстрацией последнего положения может служить дифференциация таких отношений, выраженных в суждениях: «Сегодня теплее, чем вчера», «Сегодня значительно теплее, чем вчера», «Сегодня жарче, чем вчера» и т. д. Понятно, что если все указанные отношения рассматривать как обычное отношение порядка, т. е. без учета интенсивности, приведенные суждения в логическом плане будут эквивалентны.

Для выражения интенсивностей свойств и отношений в ЕЯ используются специальные слова и словосочетания, называемые *модификаторами* или *размытыми квантификаторами* [7, 8, 129]. В таком качестве обычно выступают слова: близко, рядом, далеко, редко, часто, всегда, никогда, как правило, сильнее, слабее, выше, ниже, очень, быстро, медленно, мало, много и пр. Упорядочение модификаторов, принадлежащих одному классу, осуществляется с помощью порядковых лингвистических шкал [129, 131].

В психологическом плане особую роль играют *оппозиционные шкалы,* представляющие собой разновидность порядковых шкал, концы которых соответствуют крайним, несовместимым интенсивностям свойств и отношений, обозначаемым парами слов-антонимов. Среднее, промежуточное положение на оппозиционных шкалах является нейтральным. Примерами антонимов, располагающихся на концах оппозиционных шкал, могут служить пары: быстрый—медленный, острый—тупой, сильный—слабый, добрый—злой и т. д. Применение оппозиционных шкал позволяет не только сопоставлять интенсивности свойств и отношений, но и прослеживать направления их потенциальных изменений, а также очерчивать границы этих изменений. Отметим, что существующие границы оппозиционных шкал не следует интерпретировать абсолютно, так как с помощью модификаторов «очень», «сверх» и т. п. их можно «раздвинуть».

Предположение об отображении в сознании человека приобретаемых им знаний на систему оппозиционных шкал лежит в основе пятого выделенного концептуального свойства знаний. Интеграция базовых оппозиционных шкал образует *многомерное семантическое пространство,* точки которого соответствуют различным понятиям, а расстояния между точками — семантической дистанции между этими понятиями. Исследования, проводимые в данном направлении психологами, свидетельствуют о том, что точки в семантическом пространстве располагаются неравномерно, группируясь в кластеры в зависимости от двух ключевых факторов: частоты использования того или иного понятия в ЕЯ и ситуативной общности понятий. Само понятие многомерной шкалы выходит за рамки традиционной математической интерпретации шкалы как чисто линейного (т. е. одномерного) объекта типа координатной оси. Вместе с тем, поскольку многие свойства и отношения могут изменяться в нескольких аспектах (интенсивностях), существование подобных шкал объективно. Так, например, звук принято характеризовать амплитудой, частотой и фазой (трехмерное представление), а положение точки на плоскости — значениями пары координат по осям X и Y (два измерения).

Акцентирование внимания на свойстве *активности знаний* обусловлено естественным с прагматической точки зрения и искусственным в психологическом плане делением знаний на декларативные и процедурные. При таком подходе активным, т. е. порождающим новые знания, компонентом БЗ

является процедурная составляющая, а «пассивная» декларативная часть используется в качестве хранилища фактических сведений. Основной недостаток подобной организации БЗ заключается в ее несоответствии особенностям восприятия, обработки и интерпретации знаний человеческим мышлением, для которого характерна активация декларативными знаниями процедурных. На практике в процессе функционирования интеллектуальной системы указанная активация должна иметь место в следующих типовых ситуациях.

1. При поступлении в систему новых знаний должна осуществляться их верификация с целью согласования новых знаний и уже содержащейся в БЗ информации. Результатом этой верификации может быть модификация новых знаний и (или) сведений из БЗ, генерация запроса на уточнение «сомнительных» положений или же полная фальсификация поступившей информации.

2. При поиске ответа на внешний запрос в случае отсутствия в БЗ релевантных «готовых» знаний используются механизмы вывода знаний, в которых обращение к тем или иным компонентам БЗ может активировать ассоциируемые с ними процедуры.

3. В ходе автономного функционирования ИАС:

- при выявлении в БЗ устаревших сведений выполняется их корректировка (обновление), включающая анализ связанных с ними компонентов БЗ;
- при выявлении в БЗ неполноты знаний, относящихся к некоторой ситуации, осуществляется генерация запроса на пополнение БЗ;
- при выявлении в БЗ «сомнительных» семантических структур проводится их преобразование с целью разрешения содержащихся в них скрытых противоречий;
- осуществляется формирование «оптимального» представления знаний в БЗ с точки зрения частоты обращения к ним, характера поступающих внешних запросов и т. п.

В настоящее время практическая реализация свойства активности в БЗ в основном связывается с использованием структур, объединяющих декларативные и процедурные компоненты БЗ. К подобным структурам относятся фреймы, слоты которых могут ассоциироваться с так называемыми присоединенными процедурами. В этом же плане перспективной является объектно-ориентированная парадигма.

Еще один аспект активности связан с переходом от знаний, интерпретируемых как фактические сведения (знания-копии, знания-знакомства) к уровням умений и навыков (см. рис. 5.2, деление в зависимости от глубины знаний). Постепенное усвоение и закрепление знаний в ходе обучения преследует цель превращения этих знаний в своего рода «руководства к действиям»: совокупность правил, непосредственно используемых человеком в процессе принятия решений. Таким образом, практическую ценность обу-

чения правомерно связать со степенью активности приобретенных знаний или мерой готовности их применения для получения новых знаний.

Одна из ключевых проблем, возникающих при построении ИАС, состоит в необходимости выбора и реализации ***способа представления знаний.*** Важность данной задачи обусловливается тем, что именно представление знаний в конечном итоге определяет характеристики системы. Используемый метод представления знаний оказывает влияние на решение таких вопросов инженерии знаний, как извлечение знаний из различных источников, приобретение знаний от экспертов, интеграция и согласование знаний и пр. Последующий материал настоящей главы посвящен анализу принципов и требований к организации представления знаний в БЗ.

Поскольку на практике ИАС создаются для решения не какой-то одной, частной задачи, а множества задач, лежащая в их основе формализация знаний также должна быть достаточно универсальной, т.е. пригодной для адекватного описания совокупности *типовых проблемных ситуаций.* Соотнесение способа формализации с классом типовых проблем позволяет говорить о целесообразности выделения знаний о предметной (или проблемной) области, отражающих как общевыполнимые законы и правила, так и специфику объектов этой ПрО.

Итак, первое требование к представлению знаний – *общность* или относительная (в рамках выбранной ПрО) универсальность. Следующее подобное требование заключается в применении *психологичного,* интуитивно понятного людям (экспертам и пользователям системы) представления знаний. Заметим, что большинство рассмотренных выше концептуальных свойств знаний (структурный подход, шкалирование, активность) были определены на основании анализа психологических аспектов естественного мышления. Однако, 100%-ное следование данному требованию невозможно, так как ЭВМ, обрабатывающие знания, оперируют с ними во внутреннем, жестко структурированном представлении. Таким образом, необходимо обеспечить преобразование знаний из внутреннего во внешнее (пользовательское) представление и наоборот. В современных ИАС эти функции выполняются интерфейсным модулем, реализующим методы естественно-языкового диалога и визуализации информации, включая применение мнемонических схем, иконических образов (пиктограмм), когнитивной графики и пр. Развитие интеллектуальных интерфейсов также связано с использованием мультимедиатехнологий и систем «виртуальной реальности».

Следующее важное требование к представлению знаний — *однородность.* Однородное представление приводит к упрощению механизма управления логическим выводом и упрощению управления знаниями. Практически однородность связывается с унификацией типов (единиц) знаний (понятий, свойств, отношений), введением системы их иерархических классификаций, упорядочением внешних запросов и т. д.

В историческом плане возможность практического применения знаний возникла вместе с появлением механизма логических выводов, а механизм выводов, в свою очередь, был определен из представления знаний [138]. Таким образом, для того чтобы система обработки знаний обеспечивала реализацию заданных функциональных потребностей, должно быть создано соответствующее представление знаний. Сказанное еще раз иллюстрирует теснейшую взаимосвязь вопросов представления и обработки (использования) знаний. Фактически же к представлению знаний как направлению ИИ традиционно относят задачи проверки содержимого БЗ на непротиворечивость и полноту (верификация знаний), пополнения БЗ за счет логического вывода на основе имеющихся в базе знаний, обобщения и классификации знаний (систематизация знаний) [10].

Дифференциация ментальных (мысленных) представлений человека и формально-логического представления знаний (см. различные уровни трактовки знаний на рис. 5.1) служит фундаментом для выделения понятия ***модели знаний,*** определяющей способ формального описания знаний в БЗ. Модель – объект, реальный, знаковый или воображаемый, отличный от исходного, но способный заменить его при решении задачи. Она определяет форму представления знаний в БЗ.

Основные выводы

1. Опора на знания — базовая парадигма ИИ.
2. Существует развитая классификация знаний. В интеллектуальных информационных технологиях используются все разновидности знаний.
3. Категория знания относится к неопределяемым и специфицируется шестью концептуальными свойствами, отличающими знания от данных.
4. Основным средством представления знаний в ИАС служит БЗ. Способ представления знаний в БЗ определяется моделью знаний.

Вопросы для самопроверки

1. Охарактеризуйте понятие «знания».
2. Какие виды знаний принято выделять?
3. В чем заключается различие между декларативными и процедурными знаниями?
4. Что понимается под не-факторами?
5. Что такое эвристики? Для чего они используются?
6. Что относят к концептуальным свойствам знаний? Поясните каждое свойство.
7. Что такое размытые квантификаторы?
8. Что представляют оппозиционные шкалы?
9. В каких типовых ситуациях должно проявляться свойство активности знаний?
10. Что понимается под моделью знаний? Какова ее роль?

5.2. Модели знаний*

Представление — это совокупность соглашений относительно того, как описывать вещь.

П. Уинстон

В настоящее время применяются семь классов моделей знаний (рис. 5.3): логические, продукционные, фреймовые, сетевые, объектно-ориентированные, специальные и комплексные. Коротко остановимся на каждом из перечисленных классов.

В ***логических моделях*** знания представляются в виде совокупности правильно построенных формул какой-либо *формальной системы* (ФС), которая задается четверкой

$$(T, P, A, R), \tag{5.1}$$

где T — множество базовых (терминальных) элементов, из которых формируются все выражения ФС; P — множество синтаксических правил, определяющих синтаксически правильные выражения из терминальных элементов ФС; A — множество аксиом ФС, соответствующих синтаксически правильным выражениям, которые в рамках данной ФС априорно считаются

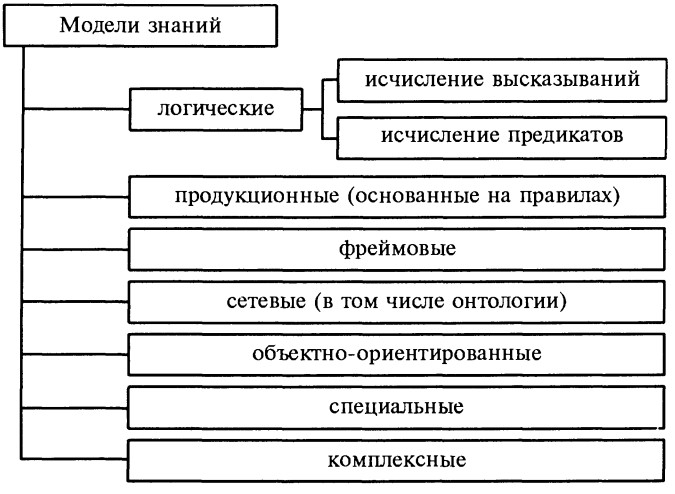


Рис. 5.3. Классы моделей знаний

* Содержание параграфа соответствует направлению исследований в области ИИ 2.1.1.

истинными; R — конечное множество правил вывода, позволяющих получать из одних синтаксически правильных выражений другие.

Простейшей логической моделью является *исчисление высказываний.* Аксиоматический базис исчисления высказываний (множество A из (5.1)) составляет совокупность правильно построенных формул, являющихся тождественно истинными. К системе аксиом предъявляют требования непротиворечивости, независимости и полноты. Названным требованиям удовлетворяет система из четырех аксиом, предложенная Д. Гильбертом [149].

В качестве правил вывода исчисления высказываний (множество R из (5.1)) обычно используют два: правило отделения (если X и $(X \rightarrow Y)$ — истинные формулы, то Y также истинна) и правило подстановки, разрешающее в правильно построенных формулах заменять все вхождения одного высказывания на другое.

Исчисление высказываний представляет собой один из начальных разделов математической логики, служащий основой для построения более сложных формализмов. В практическом плане исчисление высказываний применяется в ряде ПрО (в частности, при проектировании цифровых электронных схем). Однако как самостоятельная модель знаний логика высказываний бедна выразительными средствами: ее возможностей достаточно лишь для описания связей между высказываниями, рассматриваемыми как неразделенное целое. С помощью подобного логического исчисления нельзя формализовать умозаключения, в которых существенное значение имеет внутренняя логическая структура образующих их суждений.

Развитие логики высказываний нашло отражение в *исчислении предикатов первого порядка.* В основе исчисления предикатов лежит отождествление свойств и отношений с особыми функциями, называемыми логическими или пропозиционными. Аргументами пропозиционных функций (или предикатов) являются термы, соответствующие объектам-носителям свойств и отношений, а значениями этих функций могут быть истина или ложь. В случае выражения свойства истина интерпретируется как факт принадлежности данного свойства терму-аргументу, ложь обозначает отрицание этого факта.

Алфавит исчисления предикатов образуется за счет добавления к алфавиту исчисления высказываний символов предикатов, предметных переменных и констант, а также кванторов общности и существования. Последние два символа соответствуют операторам, позволяющим уточнить объем класса объектов, на которые распространяется действие того или иного предиката. Квантор, располагающийся перед одноместным предикатом, содержащим переменную в качестве аргумента, превращает этот предикат в высказывание, имеющее определенное истинностное значение.

Множество аксиом исчисления предикатов основывается на совокупности аксиом логики высказываний, к которым добавляются аксиомы, ха-

рактеризующие кванторы. Аналогично, множество правил вывода исчисления предикатов базируется на правилах вывода исчисления высказываний.

Представление знаний в рамках логики предикатов служит основой направления ИИ, называемого *логическим программированием* [152, 153]. Методы логического программирования в настоящее время широко используются на практике при создании ИС в ряде ПрО.

Положительными чертами логических моделей знаний в целом являются:

- высокий уровень формализации, обеспечивающий возможность реализации системы формально точных определений и выводов;
- согласованность знаний как единого целого, облегчающая решение проблем верификации БЗ, оценки независимости и полноты системы аксиом и т. д.;
- единые средства описания как знаний о ПрО, так и способов решения задач в этой ПрО, что позволяет любую задачу свести к поиску логического вывода некоторой формулы в той или иной ФС.

Отметим следующие недостатки логических моделей. Во-первых, представление знаний в таких моделях ненаглядно: логические формулы трудно читаются и воспринимаются. Во-вторых, ограничения исчисления предикатов первого порядка не допускают квантификации предикатов и использовании их в качестве переменных. В-третьих, обоснованность обозначения свойств и отношений однотипными пропозиционными функциями вызывает сомнения. Анализ данного вопроса изложен в [155, 158]. Наконец, в-четвертых, описание знаний в виде логических формул не позволяет проявиться преимуществам, которые имеются при автоматизированной обработке структур данных.

Пути повышения эффективности логических моделей знаний связаны с использованием многоуровневых и специальных логик [3, 153].

Следует заметить, что (5.1) определяет *закрытую ФС,* соответствующую аксиоматической системе, все аксиомы которой тождественно истинны вне зависимости от рассматриваемой ПрО. В базирующихся на таких ФС логических моделях используются процедуры монотонного вывода в закрытых БЗ. Свойство *монотонности* означает, что истинность полученных в процессе вывода утверждений (формул) сохраняется при расширении множества посылок. Иными словами, поступающие в систему новые факты не могут изменить истинностные значения выведенных ранее утверждений. Модели знаний с открытыми БЗ и немонотонными механизмами выводов основываются на понятии *расширенной ФС (семиотической системы)* [148], задаваемой кортежем:

$$(T, P, A, R, cT, cP, cA, cR), \tag{5.2}$$

где T, P, A, R — составляющие замкнутой ФС (5.1); cT — правила изменения базовых элементов ФС; cP — правила изменения синтаксиса ФС; cA — правила изменения аксиом ФС; cR — правила изменения правил вывода ФС.

Принадлежащие (5.2) правила модификации составляющих замкнутой ФС определяют переходы от одной ФС к другой при изменении рассматриваемой ПрО. Таким образом, единая семиотическая система может содержать противоречивые и несогласованные сведения, которые не вступают в конфликты, так как соотносятся с разными ПрО.

Центральным звеном ***продукционной модели*** является множество продукций или правил вывода. Каждая такая продукция в общем виде может быть представлена выражением

$$(W_i, U_i, P_i, A_i \rightarrow B_i, C_i), \tag{5.3}$$

где W_i — сфера применения i-й продукции, определяющая класс ситуаций в некоторой ПрО (или фрагменте рабочей ПрО), в которых применение данной продукции правомерно; U_i — предусловие i-й продукции, содержащее информацию об истинности данной продукции, ее значимости относительно прочих продукций и т. п.; P_i — условие i-й продукции, определяемое факторами, непосредственно не входящими в A_i, истинностное значение которого разрешает применять данную продукцию; $A_i \rightarrow B_i$ — ядро i-й продукции, соответствующее правилу «если..., то...»; C_i — постусловие i-й продукции, определяющее изменения, которые необходимо внести в систему продукций после выполнения данной продукции.

Трактовка ядра продукции может быть разной, например: «Если A_i истинно, то B_i также истинно»; «Если A_i содержится в БЗ, то B_i следует включить в БЗ»; «Если A_i истинно, то следует выполнить B_i» (в этом случае подразумевается, что B_i — имя процедуры, а A_i — условие ее активации или набор исходных данных для B_i); «Если A_i присутствует в описании, к которому применяется i-я продукция, то A_i следует заменить на B_i» и т. д.

Системы, основанные на продукционной модели, состоят из трех типовых компонентов: базы правил (продукций), базы фактов, содержащей декларативные знания о ПрО, используемые в качестве аргументов в условиях применимости продукций, и интерпретатора продукций, реализующего функции анализа условий применимости продукций, выполнения продукций и управления выбором продукций (управления выводом в продукционной системе).

Существуют два типа механизмов вывода в продукционных системах: прямой и обратный. В первом случае отправной точкой рассуждений служат исходные факты, к которым далее применяются продукции. В процессе вывода новые факты и правила пополняют БЗ. Часто подобные механизмы называют восходящими выводами или выводами, управляемыми данными.

Отправной точкой обратных выводов является цель рассуждения, т. е. знания, истинность которых необходимо доказать (подтвердить) или опровергнуть. Если цель согласуется с заключением какого-либо правила (с B_i), то посылка этого правила (A_i) принимается за подцель, и процесс повторяется до тех пор, пока очередная подцель не будет согласовываться с исходными фактами. Преимущество обратных (нисходящих или ориентированных на цель) выводов состоит в том, что при их использовании анализируются части дерева решений, имеющие отношение к цели, тогда как для прямых выводов характерно наличие большого числа оценок, не имеющих непосредственного к ней отношения.

В процессе вывода в продукционной системе может возникнуть ситуация, когда на каком-либо шаге удовлетворяются условия применимости нескольких продукций. Для выбора выполняемого правила привлекается информация о приоритетности, достоверности, значимости, взаимосвязи и прочих свойствах продукций. Эти знания содержатся в предусловиях правил (U_i) и могут быть эвристическими.

В системах, использующих правдоподобный вывод, каждой продукции приписывается коэффициент достоверности (фактор уверенности), отражающий оценку вероятности получения верного заключения с помощью данного правила. Таким образом, в случае конкурирования между собой нескольких продукций, ИАС может выдавать пользователю ряд альтернативных ответов, снабженных оценками их правдоподобия.

Реализация логических и продукционных моделей знаний базируется на языках типа ПРОЛОГа.

Итак, положительные стороны продукционной модели знаний состоят в ясности и наглядности интерпретации отдельных правил, а также простоте механизмов вывода (выполнения продукций) и модификации БЗ. Недостатками продукционной модели являются:

- сложность управления выводом, неоднозначность выбора конкурирующих правил;
- низкая эффективность вывода в целом, негибкость механизмов вывода;
- неоднозначность учета взаимосвязи отдельных продукций;
- несоответствие психологическим аспектам представления и обработки знаний человеком;
- сложность оценки целостного представления ПрО.

Фундаментом ***фреймовой модели знаний*** служит понятие *фрейма* — структуры данных, представляющей некоторый концептуальный объект или типовую ситуацию. Фрейм идентифицируется уникальным именем и включает в себя множество слотов. В свою очередь, каждому слоту соответствует определенная структура данных. В слотах описывается информация о

фрейме: его свойства, характеристики, относящиеся к нему факты и т. д. Кроме того, слоты могут содержать ссылки на другие фреймы или указания на ассоциируемые с ними присоединенные процедуры. Представление ПрО в виде иерархической системы фреймов хорошо отражает внутреннюю и внешнюю структуры объектов этой ПрО.

В историческом плане развитие фреймовой модели связано с теорией фреймов М. Минского, определяющей способ формализации знаний, используемый при решении задач распознавания образов (сцен) и понимания речи. «Отправным моментом для данной теории служит тот факт, что человек, пытаясь познать новую для себя ситуацию или по-новому взглянуть на уже привычные вещи, выбирает из своей памяти некоторую структуру данных (образ), называемую нами фреймом, с таким расчетом, чтобы путем изменения в ней отдельных деталей сделать ее пригодной для понимания более широкого класса явлений или процессов»*. Другими словами, фрейм — это форма описания знаний, очерчивающая рамки рассматриваемого (в текущей ситуации при решении данной задачи) фрагмента ПрО. В процессе адаптации обобщенного фрейма к конкретным условиям выполняется уточнение значений слотов, изначально определенных по умолчанию. В случае поиска в памяти фрейма, релевантного некоторому образу, на первом этапе проверяется совпадение всех существенных слотов фрейма-кандидата с соответствующими составляющими образа. На втором этапе значения слотов, заданные по умолчанию, согласуются с прочими аспектами образа. Механизм подобного согласования предусматривает выявление качественной сопоставимости компонентов фрейма и образа (например, принадлежность атрибутов образа интервалам, определенным в слотах по умолчанию), после чего значения по умолчанию конкретизируются.

Организация вывода во фреймовой системе базируется на обмене сообщениями между фреймами, активации и выполнении присоединенных процедур. Отражение в иерархии фреймов родовидовых отношений обеспечивает возможность реализации в рамках фреймовой модели операции наследования, позволяющей приписывать фреймам нижних уровней иерархии свойства, присущие фреймам вышележащих уровней. Аналогичные механизмы наследования используются в настоящее время в объектно-ориентированном проектировании.

Реализация фреймовой модели знаний базируется на языках линии LISP, FRL, KRL.

Положительными чертами фреймовой модели в целом являются ее наглядность, гибкость, однородность, высокая степень структуризации зна-

* *Минский М.* Фреймы для представления знаний: Пер. с англ. — М.: Энергия, 1979. — С. 7.

ний, соответствие принципам представления знаний человеком в долговременной памяти, а также интеграция декларативных и процедурных знаний. Вместе с тем, для фреймовой модели характерны сложность управления выводом и низкая эффективность его процедур.

Наиболее общий способ представления знаний, при котором ПрО рассматривается как совокупность объектов и связывающих их отношений, реализован в ***сетевой модели знаний***. В качестве носителя знаний в этой модели выступает *семантическая сеть,* вершины которой соответствуют объектам (понятиям), а дуги — отношениям между понятиями. Кроме того, и вершинам, и дугам присваиваются имена (идентификаторы) и описания, характеризующие семантику объектов и отношений ПрО.

Типизация семантических сетей обусловливается смысловым содержанием образующих их отношений. Например, если дуги сети выражают родовидовые отношения, то такая сеть определяет классификацию объектов ПрО. Аналогично, наличие в сети причинно-следственных (каузальных) отношений позволяет интерпретировать ее как сценарий. Построение сети на ассоциативных отношениях формирует ассоциативную структуру понятий рассматриваемого фрагмента ПрО. Вообще при практическом использовании семантических сетей для представления знаний решающее значение имеют унификация типов объектов и выделение базовых видов отношений между ними. Учет особенностей объектов и отношений, заданных их типами, обеспечивает возможность применения тех или иных классов выводов на сети.

Поскольку фактически сетевая модель объединяет множество методов представления ПрО с помощью сетей, сопоставление данной модели с прочими способами представления знаний затруднительно. Очевидные достоинства сетевой модели заключаются в ее высокой общности, наглядности отображения системы знаний о ПрО, а также легкости понимания подобного представления. В то же время в семантической сети имеет место смешение групп знаний, относящихся к совершенно различным ситуациям при назначении дуг между вершинами, что усложняет интерпретацию знаний. Другая проблема, присущая сетевой модели, состоит в трудности унификации процедур вывода и механизмов управления выводами на сети.

Перспективным направлением повышения эффективности сетевого представления, развиваемым в настоящее время, является использование онтологий и параллельных выводов на семантических сетях.

Разновидности сетевой модели знаний и технологии их реализации рассматриваются в § 5.3. Особое внимание к сетевой модели объясняется ее общим характером и значительными выразительными возможностями, а также тем, что она мало описана в доступной литературе.

Разновидностью сетевой модели является *онтология.* Быстрое развитие онтологического подхода в последние годы обусловлено распростране-

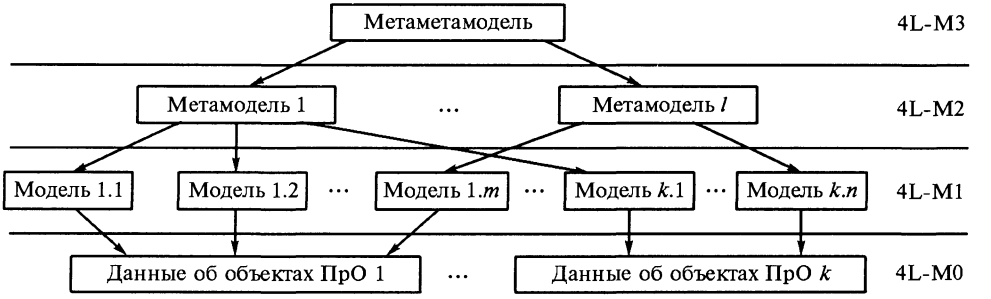


Рис. 5.4. Четырехуровневая модель MDA

нием Internet-технологий, использующих онтологические модели. Характеристика этого класса сетевых моделей приведена в § 5.4.

Объектно-ориентированная модель знаний получила широкое применение в современных технологиях проектирования разнообразных программных и информационных систем. В настоящее время существуют два основных подхода к моделированию знаний, базирующихся на объектной парадигме. Это модель MDA (Model Driven Architecture) [164, 166—168] консорциума Object Management Group (OMG) и модель ODP (Model of Open Distributed Processing), зафиксированная в стандарте ISO/ITU [160—163].

Модель MDA включает четыре концептуальные уровня: 4L-M0, 4L-M1, 4L-M2 и 4L-M3 (рис. 5.4). На уровне 4L-M0 представляются данные об объектах моделируемой ПрО, образующие универсум. На уровне 4L-M1 располагаются модели, которым следуют описания объектов ПрО на предыдущем уровне. Каждая модель на уровне 4L-M1 соответствует определенной точке зрения на ПрО (области интересов). Уровень 4L-M2 занимают метамодели, определяющие способы описания моделей уровня 4L-M1. Наконец, на уровне 4L-M3 находится единственная метаметамодель, не зависящая от точки зрения (области интересов) и определяющая способы построения метамоделей уровня 4L-M2.

Модель ODP включает три уровня: 3L-M0, 3L-M1 и 3L-M2 (рис. 5.5). Уровень 3L-M0 содержит данные об объектах моделируемой ПрО. На следующем уровне располагаются модели, определяющие способ описания объектов ПрО на уровне 3L-M0 (по одной модели для каждой ПрО). Все модели уровня 3L-M1 построены с помощью одних и тех же средств моделирования, представленных метамоделью уровня 3L-M2. Таким образом, структура всех моделей уровня 3L-M1 унифицирована.

Помимо моделей уровень 3L-M1 содержит *представления* (viewpoints), соответствующие различным точкам зрения на ПрО. Их структура также унифицирована (как и у моделей). Стандарт ODP определяет пять возможных видов представлений: корпоративное (enterprise), информационное (in-

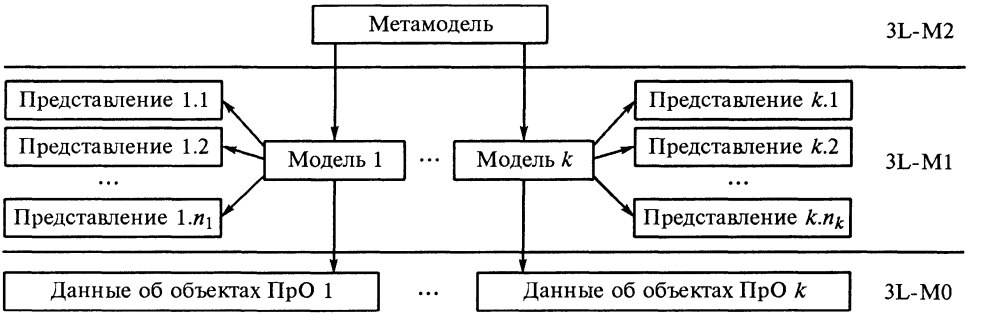


Рис. 5.5. Трехуровневая модель ODP

formation), вычислительное (computational), разработчиков (engineering) и технологическое (technology).

На уровне 3L-M2 располагается единственная метамодель, определяющая способ построения моделей уровня 3L-M1. Она отражает отношения между концептуальными категориями в них, а также семантику средств моделирования.

Сравнение рассмотренных подходов показывает, что они позволяют моделировать одну и ту же ПрО с разных точек зрения: в MDA — моделями, располагающимися на уровне 4L-M1, в ODP — представлениями уровня 3L-M1. Но если в MDA 4L-M1-модели описывают ПрО непосредственно, то в ODP 3L-M1-представления описывают ПрО через 3L-M1-модели. Это имеет как плюсы (возможность установления формального взаимного соответствия 3L-M1-представлений, построенных для одной и той же ПрО), так и минусы (возможна потеря информации при переходе от ПрО через 3L-M1-модель к 3L-M1-представлению). Кроме того, набор видов 3L-M1-представлений предопределен 3L-M2-метамоделью [165].

Класс ***специальных моделей знаний*** объединяет модели, отражающие особенности представления знаний и решения задач в отдельных, относительно узких ПрО. Характеристики ряда специальных моделей даны в [150, 152, 154]. В качестве примера подобного способа формализации знаний можно привести модель представления ПрО «объект-признак», используемую в автоматизированных системах поиска аналогов и построения классификаций [146, 155, 158].

Применение на практике того или иного способа формализации обусловливается спецификой задачи, для решения которой планируется использовать БЗ. По мнению специалистов [2—4] наиболее перспективны смешанные или ***комплексные модели,*** интегрирующие преимущества рассмотренных выше базовых моделей представления ПрО. Примером комплексной модели может служить представление, использующее для описания знаний ленемы [3, 148].

Подводя итоги анализа проблемы представления знаний, выделим девять ключевых *требований к моделям знаний:*

1) общность (универсальность);

2) «психологичность», наглядность представления знаний;

3) однородность;

4) реализация в модели свойства активности знаний;

5) открытость БЗ;

6) возможность отражения в БЗ структурных отношений объектов ПрО;

7) наличие механизма «проецирования» знаний на систему семантических шкал;

8) возможность оперирования нечеткими знаниями;

9) использование многоуровневых представлений (данные, модели, метамодели, метаметамодели и т. д.).

Ни одна из рассмотренных моделей знаний не удовлетворяет всем девяти требованиям. Сказанное подтверждает актуальность разработки методики построения комплексных многоуровневых моделей представления ПрО, обеспечивающих возможность реализации свойств, соответствующих указанным требованиям.

Основные выводы

1. В ИАС используются семь классов моделей знаний. Каждый из них имеет свои области применения, достоинства и недостатки. Из традиционных моделей наибольшее распространение получили продукционная и фреймовая модели.

2. В ИИ сложился устоявшийся набор требований к моделям знаний. Однако ни одна из базовых моделей полностью этому набору не удовлетворяет.

3. В настоящее время интенсивно развиваются объектно-ориентированные и сетевые модели знаний, в том числе онтологии.

4. В промышленных ИАС часто используют комплексные и специальные модели знаний.

Вопросы для самопроверки

1. Какие существуют классы моделей знаний?
2. Дайте характеристику логических моделей знаний.
3. Сформулируйте определение семиотической системы.
4. Что служит центральным звеном продукционной модели знаний?
5. Чем различаются прямой и обратный выводы в продукционной модели знаний?
6. Охарактеризуйте фреймовую модель знаний.
7. Охарактеризуйте сетевые модели знаний.
8. Сравните объектно-ориентированные модели знаний MDA и ODP.
9. Сформулируйте основные требования к моделям знаний.

5.3. Сетевые модели знаний*

Всегда, везде, во всем ищу причинно-следственные связи.

С.А. Стебаков

Семантические сети — наиболее мощный класс математических моделей для представления знаний о ПрО, одно из важнейших направлений ИИ. В общем случае под семантической сетью понимается структура

$$S = (O, R), \tag{5.4}$$

где $O = \{O_i, i = \overline{1, n}\}$ — множество объектов ПрО; $R = \{R_j, j = \overline{1, k}\}$ — множество отношений между объектами.

В литературе выделено около 200 типов семантических отношений. Все они представлены в ЕЯ и обладают разными свойствами как с математической, так и с лингвистической точек зрения. Разработка универсального представления семантической сети при таком количестве типов отношений является сложной и трудоемкой задачей. К тому же оперировать подобной универсальной моделью было бы затруднительно.

При решении конкретных задач значимыми являются не все типы отношений, а их подмножество. Поэтому модели семантических сетей обычно разрабатываются как проблемно-ориентированные. В таких моделях используют ограниченное число типов отношений (обычно не более 20—30).

Среди моделей семантических сетей, разработанных в России в последние десятилетия, наибольшую известность получили следующие:

*М*1 — расширенные семантические сети И.П. Кузнецова [173];

*М*2 — неоднородные семантические сети Г.С. Осипова [177];

*М*3 — нечеткие семантические сети И.А. Перминова [178, 179];

*М*4 — обобщенная модель представления знаний о ПрО А.И. Башмакова [169, 170, 175].

Кратко рассмотрим выделенные модели.

5.3.1. Модель *М*1 — расширенные семантические сети

Модель расширенной семантической сети (РСС) служит единой основой для комплексной обработки различных видов знаний, выражаемых в форме ЕЯ. Расширенные семантические сети выступают в качестве однородного средства, ориентированного на представление и обработку структу-

* Содержание параграфа соответствует направлению исследований в области ИИ 2.1.1.4.

рированной информации. Это объясняется тем, что ЕЯ с современной точки зрения является структурированным объектом.

Открытый характер многих ПрО, постоянное расширение круга решаемых в них задач требуют обеспечения развитых функциональных возможностей модели. Эту цель следует достигать не за счет трудоемкого программирования, а вводом в модель новых декларативных сведений в удобной для человека форме. Они должны автоматически преобразовываться во внутреннее представление.

Необходимо, чтобы язык представления знаний был ориентирован на автономную внутрисистемную обработку. Поэтому средства представления знаний должны быть простыми, однородными и достаточно выразительными для описания структуры ПрО. В наибольшей степени этим критериям соответствуют семантические сети.

Для сетевой модели знаний требуется унифицированная методика комплексного решения задач ввода и обработки знаний, использующая средства, отличные от логики предикатов, которая является недостаточной для поддержки лингвистической компоненты, организации интерфейса и т. д.

Для устранения неоднородности обычных семантических сетей, обусловленной наличием кванторов, совокупностей объектов, связанных отношениями, отношений, связанных между собой отношениями, и другими факторами, в модели РСС введены вершины, соответствующие именам отношений, и вершины связи, выполняющие роль «развязывающих» элементов.

Вершина связи разрывает дугу сети и подсоединяется одним ребром к вершине-отношению, а другим ребром к вершинам-объектам.

Множество вершин обозначим через D. Тогда *элементарный фрагмент (ЭФ)* РСС представляет собой k-местное отношение

$$D_0(D_1, D_2, ..., D_k / D_{k+1}), \tag{5.5}$$

где D_0 — имя отношения; $D_1, ..., D_k$ — объекты, участвующие в отношении; D_{k+1} — вершина связи, обозначающая всю совокупность объектов, участвующих в отношении; эта вершина также называется *с-вершиной ЭФ*; $D_0, D_1, D_2, ..., D_{k+1} \in D$, $k > 0$.

Расширенная семантическая сеть рассматривается как конечное множество ЭФ:

$$\text{РСС} = \{\text{ЭФ}_i\},\ i = \overline{1, n}. \tag{5.6}$$

Она может представлять наборы отношений, различные ситуации, сценарии и т. д. Разные ЭФ обязаны иметь различные c-вершины, выполняющие роль обозначений своих ЭФ.

Формально описание РСС имеет вид:

1) если $\{D_0, D_1, ..., D_k, D_{k+1}\} \subseteq D, \quad k > 0,$ то $D_0(D_1, ..., D_k / D_{k+1}) = T_0$;

2) каждый T_k есть РСС;

3) если T_1 и T_2 – РСС, то композиции T_1T_2 и T_2T_1 – тоже РСС, при этом $T_1T_2 = T_2T_1$.

Имена отношений играют роль объектов и могут, в свою очередь, вступать в отношения. Этим определяется высокая однородность модели. Вершина связи ЭФ T_k может входить в другие ЭФ, но не в роли *c*-вершины.

Множество вершин D разбито на три непересекающихся подмножества:

$$D = G \cup X \cup E, \tag{5.7}$$

где G — распознанные вершины (определенные компоненты); X — нераспознанные вершины (переменные компоненты, их роли определяются в ходе дальнейшей обработки модели); E — специальные вершины (используются при описании продукций).

Множество G состоит из трех подмножеств:

$$G = R \cup A \cup \{t, f\}. \tag{5.8}$$

Здесь R — множество имен отношений (соответствуют D_0); A — множество объектов и их классов (понятий); t — истина (true); f — ложь (false).

Через ЭФ *можно выражать и операции* над ними. Для этого проводится расширение множества R. Отсюда и произошло название модели — РСС.

Для представления операций в R вводятся вершины:

- для теоретико-множественных отношений — $\{\cap, \cup, \in, \setminus\} \subset R$;
- для арифметических выражений — $\{+, -, *, :\} \subset R$;
- для логических конструкций — $\{\wedge, \vee, \neg\} \subset R$;
- для языка логики предикатов — $\{\forall, \exists\} \subset R$;
- для запросов — вершина $? \in R$.

Модель РСС позволяет представлять нечеткие категории, различного рода неопределенности, а также динамику изменения объектов и стратегии их поведения.

Важным аспектом модели $M1$ является возможность отражения в ней *продукций.* При этом левой и правой частям продукций ставятся в соответствие ЭФ $Tn1$ и $Tn2$, а продукция записывается в виде

$$Tn1 \rightarrow Tn2. \tag{5.9}$$

Для представления продукций в множество E включены две специальные вершины: P и S; $P \in E$ — специальная вершина модели РСС, соответствующая отношению причина—следствие, $S \in E$ — специальная вершина, соответствующая отношению часть—целое. Пусть $V_1, V_2, ..., V_m$ —

c-вершины, представляющие ЭФ из сети *Tn*1, $W_1, W_2, ..., W_k$ — *c*-вершины, представляющие ЭФ из сети *Tn*2; D_1 — вершина, соответствующая левой части продукции в целом; D_2 — вершина, соответствующая правой части продукции в целом. Для отделения левой части продукции от правой вводятся ЭФ, в записях которых на первых местах стоят символы *P* и *S*. Запись $S(D_1, V_i)$ означает, что вершина V_i — часть D_1, а запись $P(D_1, D_2)$ фиксирует, что РСС с именем D_1 является причиной активации РСС с именем D_2.

Тогда подробная запись продукции (5.9) принимает вид:

$$Tn1\ \ Tn2\ \ S(D_1, V_1)\ ...\ S(D_1, V_m)\ \ P(D_1, D_2)\ \ S(D_2, W_1)\ ...\ S(D_2, W_k) \quad (5.10)$$

В свою очередь, продукции могут входить как в левые, так и в правые части других продукций. Таким образом, на однородной основе создаются метауровни РСС.

Для обработки знаний, представляемых РСС, используется принцип сопоставления по образцу в виде метода наложения двух сетей. В его основе лежат правила идентификации, позволяющие связывать вершины и сопоставлять сети в соответствии с законами логики.

На базе модели РСС построен язык продукционного программирования Декл. В отличии от языка ПРОЛОГ в Декл условная часть записывается слева, а следствие справа. В обеих частях может быть множество ЭФ.

Понятие ЭФ шире, чем предикат за счет *c*-вершин. Метауровень Декл также организован иначе: продукции на нижнем уровне записываются как наборы ЭФ, которые могут обрабатываться продукциями. Процесс вывода носит более гибкий характер за счет операторов активации и использования принципов магазинной памяти.

5.3.2. Модель *M*2 — неоднородные семантические сети

Неоднородная семантическая сеть (НСС) представляет собой ориентированный граф с помеченными вершинами и ребрами. Вершины сети соответствуют событиям, образующим их фактам и комбинациям событий. Ребра представляют отношения совместности событий и отображение событий в заключения в соответствии с правилами вывода.

Формально НСС описывается структурой

$$(D, \tau, \Delta, F, R), \quad (5.11)$$

где *D* — семейство произвольных множеств D_1 ..., D_n; $\tau = \{\bar{k}_1, \bar{k}_2, ..., \bar{k}_e\}$ — множество типов; Δ — множество событий; *F* — семейство функций, соотносимых с типами и отражающих связи между событиями и одним из множеств семейства *D*; *R* — семейство отношений совместности событий.

Тип $\overline{k_i}$ представляет собой упорядоченное подмножество индексов из множества $\{1, 2, \ldots, n\}$ ($n = |D|$). Для каждого типа $\overline{k_i} = (n_1, \ldots, n_{k_i})$ строится декартово произведение множеств из D:

$$D_{\overline{k_i}} = D_{n_1} \times \ldots \times D_{n_{k_i}} \quad (n_{k_i} \leqslant n). \tag{5.12}$$

Тип $\overline{k_i}$ интерпретируется как тип декартова произведения (5.12).

Для каждого $D_{\overline{k_i}}$ определяется совокупность его подмножеств $\Delta_{\overline{k_i}}$. Всякое $d \in \Delta_{\overline{k_i}}$ называется *событием* типа $\overline{k_i}$. Объединение всех $\Delta_{\overline{k_i}}$ образует множество Δ: $\Delta = \bigcup_{\overline{k_i} \in \tau} \Delta_{\overline{k_i}}$.

Функции из F, соответствующие типу $\overline{k_i}$, отображают $D_{\overline{k_i}}$ в одно из множеств семейства D.

На множестве событий задано семейство *отношений совместности* событий:

$$R = \{R_1, R_2, R_3, R_4, R_5, R_6, R_1^0, R_2^0, R_3^0, R_4^0, R_5^0, R_6^0\}, \tag{5.13}$$

причем R_i^0 обратно для R_i: $R_i^0 = (R_i)^{-1}$, $i = \overline{1, 6}$.

При создании модели НСС набор отношений совместности был сформирован, исходя из потребностей обработки данных в медицинских ЭС. Он включает следующие отношения:

R_1 — отношение нестрогого порядка (транзитивно, рефлексивно, антисимметрично);

R_2 — нетранзитивно, рефлексивно, симметрично;

R_3 — нетранзитивно, антирефлексивно в классе отношений совместности, симметрично;

R_4 — транзитивно, антирефлексивно в классе отношений совместности, антисимметрично;

R_5 — нетранзитивно, рефлексивно, антисимметрично в классе отношений совместности;

R_6 — нетранзитивно, антирефлексивно, асимметрично в классе отношений совместности.

В модели $M2$ установлена связь между отношениями из R и типами формируемых ЭС сообщений. Эта связь выражена в виде правил перевода и подстановки. Для моделирования рассуждений на НСС разработаны алгоритмы, основанные на использовании свойств отношений совместности и следования событий. Доказаны теоремы, устанавливающие связь между отношениями совместности и выполнимостью событий в различных состояниях НСС.

Модель *М*2 ориентирована на использование в ЭС и положена в основу системы прямого приобретения экспертных знаний SIMMER. Эта система позволяет в процессе взаимодействия с экспертом определять выполнимость событий в различных состояниях НСС.

5.3.3. Модель *М*3 — нечеткие семантические сети

Один из основных недостатков существующих языков ИИ связан с тем, что они обеспечивают плохие возможности структуризации как для представления данных, так и особенно для организации процесса вычислений, что затрудняет эффективное создание с помощью них сложных интеллектуальных систем. Кроме того, при разработке прикладных ИАС помимо задач ИИ необходимо решать и общепрограммистские задачи, для чего языки ИИ плохо приспособлены. По этим причинам была поставлена задача создания хорошо структурированного модульного языка ИИ, который мог бы служить и языком программирования общего назначения.

Предусмотренный в *М*1 механизм обработки РСС имеет два существенных недостатка: слабая структурированность формируемых программ и сложность организации процесса вычисления. Для их преодоления на базе *М*1 разработана модель *объектно-ориентированной семантической сети (ОСС)*. В ее основе лежит объединение РСС с нечетким ПРОЛОГом с использованием объектно-ориентированного подхода. В сеть добавлены средства обработки нечеткой информации, вершины-переменные, вершины-классы, вершины-объекты и вершины-экземпляры. Аргументы предикатов интерпретируются как вершины семантической сети, а факты представляются ее фрагментами. Правила привязаны к классам.

Синтаксис ПРОЛОГа расширен для обработки сети. При этом предикат ПРОЛОГа ставится в соответствие ЭФ РСС. Так, предикату L:*relation*(X_1, …, X_k) соответствует ЭФ $D_0(D_1, \ldots, D_k / D_{k+1})$, где вершина D_0 соответствует имени отношения *relation*, вершины D_i — предметным переменным X_i ($i = \overline{1, k}$), а L — вершине связи D_{k+1}. При таком подходе правила ПРОЛОГа являются частным случаем сетевых продукций (СП) в РСС.

В объектно-ориентированную семантическую сеть введены средства обработки нечеткой информации. Для этого каждому $ЭФ_j$ сети ставится в соответствие показатель $h_j \in [0; 1]$, выражающий степень истинности при $h_j \in]0{,}5; 1]$ или ложности при $h_j \in [0; 0{,}5[$ факта, представленного $ЭФ_j$. Такой нечеткий ЭФ служит элементарной структурной единицей *нечеткой ОСС* (*НОСС*). При $h_j = 0{,}5$ (неизвестно, истинен или ложен факт) $ЭФ_j$ отсутствует в ОСС.

В нечеткой ОСС различаются вершины следующих типов.

1. Простые вершины-сущности ПрО без детализации объектно-ориентированными средствами.

2. Вершины-переменные, содержащие присваиваемую ссылку на другую вершину сети.

3. Вершины-описатели отношений (ассоциируются с вершинами D_0 ЭФ). С каждой такой вершиной связано описание свойств отношения.

4. Вершины-связи (ассоциируются с вершинами D_{k+1} ЭФ), используемые для ссылок на ЭФ как на единое целое.

5. Вершины-описатели классов, представляющие используемые в сети классы объектов. С каждой из них связано описание свойств и правил класса, а также значения свойств.

6. Вершины-объекты, представляющие используемые в сети объекты. С каждой такой вершиной ассоциированы значения свойств, принадлежащих объекту данного класса.

7. Вершины-экземпляры. С каждой из них ассоциированы значения свойств, принадлежащих данному экземпляру.

Следующие типы вершин представляют встроенные в НОСС типы данных:

- вершины-нечеткие множества;
- вершины-строки;
- вершины-целые числа;
- вершины-вещественные числа.

Для эффективной организации больших объемов данных и кода система, реализующая модель НОСС, имеет средства для разбиения на модули.

Вершина-описатель класса НОСС содержит выполнимые правила и описания вычислимых и невычислимых свойств. Свойства и методы имеют атрибут доступности из других классов и модулей. Как и в нечетком ПРОЛОГе, выполнение программы базируется на методе нечеткой резолюции. Результат вывода характеризуется степенью истинности заключения, степенью доверия к резольвенте и интегральной оценкой, основанной на первых двух, — смешанной истинностью заключения. Из-за недостатков известной стратегии нечеткого вывода в НОСС использована линейная стратегия вывода с бэктрекингом, позволяющая рассматривать программу с декларативной точки зрения как выполнение нечеткого вывода, а с процедурной точки зрения как вызов рекурсивных процедур или функций.

Каждое правило программы, функционирующей на базе НОСС, принадлежит какому-либо классу в сети и рассматривается как метод в этом классе. Различают правила, встроенные в систему и определенные пользователем.

Для описания данных и программного кода, реализующих сеть, разработан язык FSNL (Fuzzy Semantic Network Language) — объектно-ориентированный язык без строгой типизации, предназначенный для оперирования НОСС и удовлетворяющий выдвинутым выше требованиям. Язык FSNL

может быть использован для решения как задач ИИ, так и общепрограммистских задач.

5.3.4. Модель *М*4 — обобщенная модель представления знаний о предметной области

При разработке модели *М*4 учитывались следующие ключевые требования:

- обеспечение возможности настройки на различные ПрО;
- наглядность представления (наличие геометрической интерпретации основных компонентов модели, обеспечение возможности их визуального формирования);
- высокая однородность модели, упрощающая представление знаний и манипулирование ими;
- открытость, понимаемая как возможность расширения модели без переопределения ее ядра;
- наличие условий для реализации свойства активности знаний;
- высокая структурированность, основанная на наличии в модели механизмов композиции и декомпозиции;
- обеспечение возможностей оперирования с нечеткими представлениями.

Отправной точкой при создании любой модели знаний о ПрО является *выбор ее категориального аппарата.* Анализ категориального аппарата моделирования знаний позволил выделить в качестве его центрального звена *триаду вещь—свойство—отношение.* Необходимо подчеркнуть, что в аппарате формальной логики категории вещи, свойства и отношения занимают одну из ключевых позиций [180, 181]. Вместе с тем, данные категории, как правило, трактуются как самые общие и элементарные понятия, в результате чего в центр моделей знаний всегда ставились более сложные с точки зрения внутренней структуры производные категории.

Среди основных проявлений взаимосвязи составляющих триады необходимо отметить:

- *взаимообоснование* свойств и отношений в вещах (отношения, связывающие вещи, устанавливаются по свойствам этих вещей, а свойства вещи рассматриваются в определенных отношениях);
- *взаимопереход* вещей, свойств и отношений (каждая из базисных категорий может интерпретироваться как вырожденный случай других базисных категорий и, в пределе, переходит в них).

Принципы взаимообоснования и взаимоперехода обусловливают *высокую степень однородности* модели представления ПрО, основанной на триаде вещь—свойство—отношение. Действительно, между вещью, свойст-

вом и отношением нет непреодолимых границ: любая из этих категорий является особым случаем других, интерпретируется через них и, в пределе, в них переходит. Таким образом, вместо отдельных множеств вещей, свойств и отношений правомерно рассматривать единое множество объектов, каждый из которых в конкретных описаниях выступает в одной из трех указанных форм.

В типовой архитектуре интеллектуальной системы выделяются три уровня: совокупность информационных структур (БЗ), множество операций над ними и стратегия управления операциями. Интерпретируемая в широком смысле модель представления ПрО, реализуемая в подобной системе, также охватывает эти уровни. Этот принцип воплощен в *М4*, которая включает *три базовых уровня:*

- информационных структур;
- операций;
- стратегии управления операциями.

Вывод об однородности был сформулирован по отношению к уровню информационных структур. Его формальное описание базируется на следующих основополагающих принципах.

1. Центральным элементом модели представления ПрО является множество объектов. Каждый объект существует в трех различных формах: как вещь, как свойство и как отношение, причем только вещь может выделяться и рассматриваться самостоятельно.

2. Для выражения обоснований можно использовать три способа:

- непосредственное указание обоснующей категории (для свойства — отношения и его коррелятов, для отношения – свойств);
- конкретизация или уточнение обоснуемой категории;
- указание ссылки на обоснующую категорию.

3. Среди свойств вещи выделяются качества, или существенные свойства, без которых данная вещь существовать не может. Факт наличия отношений, установленных по качествам вещей, вытекает из факта существования их вещей-коррелятов.

4. Свойства, характеризующие вещи, могут быть базовыми и составными. Составным называется свойство, выражение которого содержит явное указание на отношение данной вещи к какой-либо другой вещи. В отличие от составных, выражения базовых свойств неструктурированны.

5. Выделяются следующие основные элементы структуры определения вещи с помощью свойств и отношений:

- в составе вещи могут находиться другие вещи — части данной вещи, причем факты наличия у вещи частей отражаются посредством указания соответствующих составных свойств;

• вещь может характеризоваться совокупностью свойств и отношений, принадлежащих другим вещам, причем подобные факты также выражаются с помощью составных свойств;

• между отдельными вещами в целом, а также частями вещи могут существовать внешние и внутренние отношения соответственно;

• каждые свойство и отношение обоснуются отношением и свойствами соответственно с помощью одного из трех (или комбинации) указанных выше способов.

6. Представления вещей являются размытыми. Выделяются следующие аспекты устойчивости принадлежности (нечеткости) формирующих их свойств и отношений:

• интенсиональная распределенность в классе вещей, являющихся элементами (частями) рассматриваемой вещи;

• экстенсиональная распределенность в классе вещей, охватываемых объемом понятия, соответствующего рассматриваемой вещи;

• временная распределенность;

• пространственная распределенность;

• интегральные оценки устойчивости.

Для выражения и дифференциации интенсивностей названных аспектов устойчивости используются специальные признаки и показатели с соответствующими шкалами, значения которых приписываются свойствам и отношениям, включенным в описание вещи.

Интенсиональная и экстенсиональная распределенности ассоциируются с типами предикации свойств и отношений вещам. Типы предикации отражают потенциальную принадлежность (распределенность) свойства или отношения вещам, входящим в состав данной вещи (интенсиональный аспект) или охватываемых объемом соответствующего ей понятия (экстенсиональный аспект). Четыре типа предикации (P^{I} — P^{IV} для свойств, R^{I} — R^{IV} для отношений) обеспечивают выделение на качественном уровне различных степеней распределенности свойств и отношений в классе вещей: среди всех элементов класса (P^{I} и R^{I}), среди большинства элементов (P^{II} и R^{II}), среди некоторых элементов (P^{III} и R^{III}) и отсутствие распределенности (P^{IV} и R^{IV}).

Экстенсиональная распределенность в ЕЯ выражается словами «все» (P^{I} и R^{I}), «большинство» (P^{II} и R^{II}), «некоторые» (P^{III} и R^{III}), «данный», «этот» (P^{IV} и R^{IV}) и др. Различия между интенсиональными и экстенсиональными типами предикации на примере свойств иллюстрирует табл. 5.3.

Итак, на уровне информационных структур модели выделяется *множество объектов* $\{O_i\}$. Объект задается тройкой:

$$O_i = (A_i, P_i, R_i), \tag{5.14}$$

где A_i — вещь, соответствующая объекту O_i; P_i — свойство, соответствующее объекту O_i; R_i — отношение, соответствующее объекту O_i.

Таблица 5.3

Тип предикации	Пример суждения и комментарий	
	для интенсиональной предикации	для экстенсиональной предикации
P^{I}	*Обед превосходен.* Оценка относится ко всем компонентам обеда: блюдам, напиткам, сервировке, обслуживанию и т. д.	*Все металлы являются проводниками.* Все виды металлов (т. е. представители данного класса) обладают хорошей электропроводностью
P^{II}	*Отряд вооружен.* Подразумевается, что большинство членов отряда вооружены	*Большинство дней в декабре пасмурные.* Указанным свойством обладают большинство членов данного класса
P^{III}	*Сложное устройство ненадежно.* Сложное устройство является ненадежным, если содержит ненадежные элементы. Таким образом, имеется в виду, что указанное свойство относится к некоторым элементам сложного устройства	*Некоторые сложные устройства ненадежны.* Указанное свойство присуще некоторым членам данного класса
P^{IV}	*Долгий путь.* Свойство относится к пути в целом и не распределяется по его составляющим	*Сегодняшний обед превосходен.* Оценка относится к индивидуальной вещи (сегодняшнему обеду) и не распределяется в классе вещей

Данное определение отражает единство категорий вещи, свойства и отношения, а также положение об их взаимопереходе. Присущая вещи самостоятельность отражается не только в зафиксированной первой позиции кортежа (5.14), но и дальнейшей конкретизации ее определенности (5.15). В рассматриваемом плане можно провести аналогию между структурой, формируемой на основе (5.14), с одной стороны, и входами и выходами некоторого материального объекта, с другой: A_i будет выходом, посредством которого объект раскрывает свою сущность, а P_i и R_i — входами, описывающими прочие объекты, примыкающие к ним (рис. 5.6).

Структура определенности вещи A_i задается шестеркой

$$A_i = (N_i, P_i^{AI}, P_i^{AH}, P_i^{P}, R_i^{H}, R_i^{A}), \qquad (5.15)$$

где N_i — уникальное имя; P_i^{AI} — нечеткое множество составных свойств типа P^{AI}; P_i^{AH} — нечеткое множество составных свойств типа P^{AH}; P_i^{P} — не-

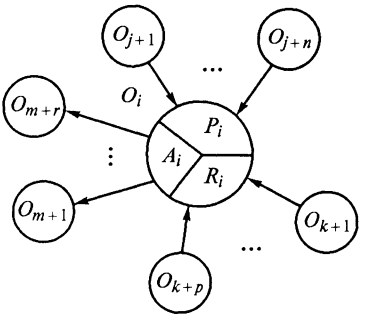


Рис. 5.6. Геометрическая интерпретация объекта в модели *M*4

четкое множество базовых свойств; R_i^H — нечеткое множество внутренних отношений; R_i^A — нечеткое множество внешних отношений.

Имя N_i соответствует вербальному носителю понятия – слову (термину) или семантически связанной последовательности терминов.

Типизация нечетких свойств и нечетких отношений зависит от проблемной ориентации БЗ, в которой реализуется *M*4, и может быть различной. Определение (5.15) отражает частный случай интерпретации составных свойств, среди которых выделены два типа: указание на вещь, обладающую свойствами и существующую в отношениях, присущих рассматриваемой вещи (родовидовые отношения), — P^{AI} и указание на вещь, являющуюся частью рассматриваемой вещи (отношения целое—часть), — P^{AH}.

В самом общем виде *нечеткое свойство* или *отношение* X можно задать парой

$$(I(X), B(X)). \tag{5.16}$$

Здесь $I(X)$ — ссылка на объект, соответствующий свойству или отношению X; $B(X)$ — обоснование свойства или отношения X.

Приведенное выражение служит наиболее общим и универсальным представлением принципа взаимообоснования. Определим следующую *интерпретацию обоснования* $B(X)$:

$$B(X) = (B'(X), S(X), T(X), \mu_M(X), \mu_I(X)), \tag{5.17}$$

где $B'(X)$ — выражение обоснования свойства или отношения X с помощью одного из трех вышеуказанных способов; $S(X)$ — распределенность свойства или отношения X; $T(X)$ — временная метка свойства или отношения X; $\mu_M(X)$, $\mu_I(X)$ — интегральные оценки устойчивости свойства или отношения X.

Распределенность свойства $S(X)$ задается кортежем

$$S(X) = (q, x_I, x_E). \tag{5.18}$$

Здесь q — качество распределенности (положительное или отрицательное), отражающее принадлежность или непринадлежность данного свойства вещи; x_I — интенсиональный индекс распределенности; x_E — экстенсиональный индекс распределенности.

Индексы x_I и x_E могут принимать четыре значения, ассоциируемые с типами предикации свойств P^{I} — P^{IV}. Интенсиональный индекс x_I отражает распространение рассматриваемого свойства «вглубь» вещи, т. е. среди ее элементов. Экстенсиональный индекс x_E соответствует распределению данного свойства «вширь», т. е. среди вещей, охватываемых объемом понятия об исходной вещи.

Распределенность бинарного отношения $S(X)$ включает четыре индекса (по два различных индекса на каждый коррелят):

$$S(X) = (q, x_{I1}, x_{E1}, x_{I2}, x_{E2}), \tag{5.19}$$

где q — качество распределенности; x_{I1}, x_{I2} — интенсиональные индексы распределенности отношения по первому и второму коррелятам соответственно; x_{E1}, x_{E2} — экстенсиональные индексы распределенности отношения по первому и второму коррелятам соответственно.

Значения индексов распределенности отношения соответствуют типам предикации R^{I} — R^{IV}.

Временная метка свойства или отношения $T(X)$ фиксирует период времени, в течение которого описываемая вещь обладает данным свойством или выступает в качестве коррелята данного отношения. Совокупность типов временных меток, статически характеризующих периоды принадлежности элементов описания вещам, предложена в [175]. Эта совокупность позволяет выделить три глобальных интервала времени (настоящее, прошедшее и будущее), а также их комбинации.

В качестве интегральных оценок устойчивости свойств и отношений предлагается использовать показатели *значимости* $\mu_M(X)$ и *информативности* $\mu_I(X)$. Первый показатель позволяет выделить наиболее важные, существенные свойства и отношения, исключение из представления вещи каждого из которых приводит к невозможности идентификации этой вещи с помощью оставшихся элементов определенности. Информативность свойства или отношения обусловливается двумя основными аспектами. Первый из них относится к внешнему проявлению восприятия рассматриваемого элемента определенности. Речь идет о яркости, запоминаемости той или иной части образа. Второй аспект касается типичности свойства или отношения и соответствует устойчивости ассоциативных связей между ними и описываемой вещью.

Интегральный характер значимости и информативности обусловлен тем, что в отличие от других элементов кортежа (5.17), отражающих различные аспекты устойчивости принадлежности свойств и отношений, которые рассматриваются по отдельности друг от друга, данные показатели оценивают место и роль элементов определенности в совокупности всех свойств и отношений, входящих в описание вещи.

В отличие от классических определений нечетких множеств, в которых каждому их члену ставится в соответствие только одно значение одной функции принадлежности, в (5.17) выделена пара показателей нечеткости (значимость и информативность). Кроме того, прочие компоненты обоснования по смыслу также являются характеристиками, отражающими различные стороны нечеткости обоснуемого элемента определенности. Формально эти характеристики могут быть описаны как нечеткие переменные. Таким образом, модель *M*4 способна отражать *многоаспектную нечеткость* представления ПрО.

Следует отметить, что предложенная в (5.17)—(5.19) детализация обоснования является лишь одной из возможных интерпретаций $B(X)$. При построении моделей представления ПрО для конкретных БЗ могут выделяться другие аспекты обоснования и использоваться альтернативные способы их формализации. Тем не менее важно подчеркнуть, что нечеткий элемент определенности должен представляться парой (5.16), первый член которой служит для семантической идентификации данного элемента, а второй отражает различные стороны его обоснования и устойчивости принадлежности описываемой вещи. При этом чем более выразительны средства представления обоснований, тем глубже интеллектуальные возможности, обеспечиваемые моделью. Сделанные замечания иллюстрирует табл. 5.4, в которой различным способам интерпретации составляющих (5.16) поставлены в соответствие известные классы моделей знаний (данных) и ИС.

Структура первого уровня *M*4 наглядно иллюстрируется ориентированным мультиграфом, вершины которого соответствуют объектам, а ребра — характеризующим вещи элементам определенности. Любая вершина (объект) в совокупности с множеством исходящих из нее ребер трактуется как вещь. Ребро, выходящее из вершины, отражает приписывание данной вещи свойства или отношения, задаваемого объектом, к которому это ребро направлено. Кроме того, каждое ребро помечено значимым и информативным весом (μ_M и μ_I), а также различными составляющими обоснования (B', S, T) указываемого им свойства или отношения. Поскольку фактически веса эквивалентны значениям функций принадлежности нечетких свойств и отношений, сформированную структуру можно интерпретировать как нечеткий взвешенный ориентированный мультиграф или семантическую сеть специального вида.

Обобщенный характер модели *M*4 состоит в том, что она *определяет методологию построения моделей представления знаний о ПрО,* реализуемых в прикладных интеллектуальных системах. Остальные уровни *M*4 будут рассмотрены в § 5.5 как пример подхода к формированию системы операций для работы со знаниями.

Таблица 5.4

Способы интерпретации элементов, представляющих нечеткое свойство или отношение X		Классы моделей знаний (данных) или информационных систем
$I(X)$	$B(X)$	
Индекс (порядковый номер)	—	Информационно-поисковая система, обеспечивающая поиск по ключевым словам
Ссылка на объект (запись)	—	Традиционные БД
Ссылка на объект (запись)	Обоснование $B'(X)$, выражаемое через родовую ссылку	Объектно-ориентированные БД
Ссылка на объект (константу, переменную, предикат, функцию и др.)	$S(X) = (q, x_E, \dots)$; логические операции	Логические модели знаний
Ссылка на объект (константу, переменную, предикат, функцию и др.)	$S(X) = (q, x_E, \dots)$, $\mu(X)$; нечеткие логические операции	Нечеткие логические модели знаний
Ссылка на объект (константу, переменную, предикат, функцию и др.)	$S(X) = (q, x_E, \dots)$, $T(X)$; логические операции, специальные операции	Временные логики (возможно, нечеткие)

Основные выводы

1. Семантические сети — наиболее мощный класс математических моделей для представления знаний о ПрО, одно из важнейших направлений ИИ.

2. В модели РСС ($M1$) за счет введения вершин, соответствующих именам отношений, и вершин связи удалось в однородном виде представить отношения, кванторы, продукции, а также теоретико-множественные, арифметические и логические операции. Эта модель реализована в языке и системе программирования Декл. Недостатками РСС являются слабая структурированность формируемых программ и сложность организации процесса вычисления.

3. Модель НСС ($M2$) ориентирована на использование в ЭС, поэтому в ней предусмотрено ограниченное множество отношений совместности событий. Это повышает эффективность реализации $M2$ в системе прямого приобретения экспертных знаний SIMMER и в то же время сужает ее применимость.

4. Модель $M3$ обеспечивает хорошую структурированность формируемых программ, благодаря чему уменьшается сложность организации вы-

вода. В ней РСС объединена с нечетким ПРОЛОГом с использованием объектно-ориетированного подхода: в сеть добавлены средства обработки нечеткой информации, вершины-переменные, вершины-классы, вершины-объекты и вершины-экземпляры. Модель *М3* реализована в языке FSNL, который может использоваться для решения как задач ИИ, так и общепрограммистских задач.

5. Обобщенная модель представления знаний о ПрО (*М4*) учитывает все основные требования к моделям знаний. Она охватывает три уровня ИАС (информационные структуры, операции, стратегии управления операциями). Ядром ее категориального аппарата служит триада вещь—свойство—отношение. Взаимосвязь компонентов триады отражают принципы взаимообоснования свойств и отношений в вещах и взаимоперехода вещей, свойств и отношений, что обусловливает высокую однородность уровня информационных структур *М4*. Обобщенный характер данной модели состоит в том, что она определяет методологию представления знаний о ПрО для прикладных ИАС.

Вопросы для самопроверки

1. Дайте определение семантической сети.
2. Какие принципы лежат в основе модели РСС?
3. Что такое вершина связи РСС и каково ее назначение?
4. Дайте определение элементарного фрагмента РСС.
5. Приведите формальное определение РСС.
6. Как в РСС представляются отношения?
7. Охарактеризуйте систему операций над ЭФ в *М1*.
8. Как в РСС описываются продукции?
9. На каких принципах базируется модель НСС?
10. Какие типы отношений совместности событий представлены в НСС?
11. Какие принципы лежат в основе модели НОСС?
12. Какие типы вершин представлены в НОСС?
13. Каковы основные требования к обобщенной модели представления знаний о ПрО (*М4*)?
14. Назовите базовые уровни и опишите основополагающие принципы модели *М4*.
15. Что обусловливает выбор категориального аппарата модели *М4*?
16. В чем состоят принципы взаимообоснования и взаимоперехода? Каково их влияние на характеристики модели *М4*?
17. Что представляет собой объект уровня информационных структур модели *М4*?
18. Какова структура определенности вещи в модели *М4*?
19. Как в модели *М4* выражаются обоснования свойств и отношений?
20. Что отражают показатели значимости и информативности в модели *М4*?

5.4. Онтологический подход и его использование*

Определите значения слов, и вы избавите человечество от половины его заблуждений.

Р. Декарт

Прежде чем спорить, давайте договоримся о терминах.

Вольтер

5.4.1. Понятие онтологии

Разработка и использование онтологий является одной из НИТ. Быстрый рост объемов ИР Internet стимулировал переход от теоретических исследований в области онтологий к практическому применению их результатов.

Основной идеей НИТ является автоматизация процедур извлечения необходимой информации по запросу пользователя либо построения программы, интересующей пользователя, на основе введенного им в систему описания постановки задачи на привычном для него профессиональном языке. Таким образом, НИТ обеспечивает возможность общения с компьютерной системой пользователя, который не является профессиональным программистом. Для воплощения этой идеи необходимо, чтобы система обладала интеллектуальным интерфейсом, БЗ и решателем, т. е. была интеллектуальной системой. Онтологии имеют непосредственное отношение к построению БЗ и частично к реализации интеллектуального интерфейса.

Другим назначением НИТ является поддержка процессов совместного решения задач, когда коллеги общаются друг с другом через сеть, используя общую БЗ. Онтологии помогают обеспечить одинаковое понимание всеми пользователями смысла применяемых при решении терминов, их атрибутов и отношений между ними.

Термин «онтология» в ИИ употребляется в контексте с такими понятиями, как концептуализация, знания, модели знаний, системы, основанные на знаниях.

Напомним, что под ***концептуализацией*** понимается процесс перехода от представления ПрО на ОЕЯ (или ЕЯ) к точной спецификации этого описания на некотором формальном языке, ориентированном на компьютерное представление. Концептуализация также трактуется как результат

* Содержание параграфа соответствует направлениям исследований в области ИИ 2.1.1.5 и 3.1.3.

подобного процесса, т. е. описание множества понятий (концептов) ПрО, знаний о них и связях (отношениях) между ними. Онтология — это формально представленные на базе концептуализации знания о ПрО.

Самым распространенным на данный момент является определение, предложенное в [197], согласно которому *онтология есть точная (выраженная формальными средствами) спецификация концептуализации.* С этой точки зрения каждая БЗ, система, основанная на знаниях, или программный агент явно или неявно базируются на некоторой концептуализации. Множества понятий и отношений между ними отражаются в словаре. Таким образом, считается, что основу онтологии составляют множества представленных в ней терминов.

Формально онтология состоит из терминов, организованных в таксономию, их определений и атрибутов, а также связанных с ними аксиом и правил вывода.

Часто набор утверждений, входящих в онтологию, представляют в форме *логической теории первого порядка,* причем термины словаря служат именами унарных и бинарных предикатов, называемых соответственно концептами и отношениями. В простейшем случае онтология описывает только иерархию концептов, связанных отношениями категоризации. В более сложных случаях в нее также включаются аксиомы для выражения других отношений между концептами и организации их интерпретации. Из сказанного следует, что онтология может быть реализована как БЗ, описывающая факты, которые предполагаются всегда истинными в рамках определенной ПрО, на основе общепринятого смысла, фиксируемого используемым словарем.

Онтология является не абсолютной (единственной) спецификацией концептуализации ПрО, а зависит от целей ее создания, т. е. задач, при решении которых планируется ее применять. Независимо от вида онтологии она должна включать словарь терминов и некоторые спецификации их значений, что позволяет ограничивать возможные интерпретации терминов и отражать взаимосвязь понятий данной ПрО. При таком подходе онтология похожа на известное понятие *тезауруса.*

Ряд исследователей располагают онтологию в центре системы представления знаний, так как в каждой ПрО могут существовать различные понимания одних и тех же терминов. Онтология используется для структурирования информации, являясь посредником между человеко- и машинно-ориентированным уровнями ее представления. При таком подходе онтология интерпретируется как система соглашений о некоторой области интересов для достижения заданных целей. Это предполагает комплексное представление соответствующей ПрО, включающее формальные и декларативные (описательные) компоненты, содержащее словарь терминов, накладываемые

на них ограничения целостности, а также логические утверждения, ограничивающие интерпретацию терминов и их отношения друг с другом.

Выделяют следующие шесть интерпретаций понятия «онтология» [198]:

1) неформальная концептуальная система (представление концептуализации);

2) формальный взгляд на семантику;

3) спецификация концептуализации;

4) представление концептуальной системы через логическую теорию;

5) словарь, используемый логической теорией;

6) метауровневая спецификация логической теории.

Согласно первой интерпретации онтология является концептуальной системой, которую можно использовать в качестве фундамента некоторой БЗ. Согласно второй интерпретации онтология, на основе которой построена БЗ, служит для формального представления семантики входящих в нее терминов. Согласно третьей интерпретации онтология – это БЗ особого типа, отчуждаемая от ее разработчиков и позволяющая однозначно понимать представленные в ней знания. Интерпретации 4—6 трактуют онтологию как специальную грамматическую систему.

В неформальной трактовке онтология представляет собой описание некоторой ПрО. Это описание состоит из терминов и правил их использования, ограничивающих их значения в этой ПрО. На формальном уровне онтология — это система, состоящая из набора понятий и набора утверждений об этих понятиях, на основе которых можно строить классы, объекты, отношения, функции и теории.

На метауровне онтология является *разновидностью сетевой модели знаний о ПрО.* Эта модель может быть статической или динамической. Во втором случае говорят об онтологии как о модели мира, которая может представлять состояния моделируемой ПрО во времени.

Web-онтологией называют онтологию, которая либо доступна на одном из web-узлов Internet, либо используется в рамках корпоративного портала.

5.4.2. Основные задачи, решаемые с помощью онтологий

Сфера применения онтологий быстро расширяется. Коротко охарактеризуем основные задачи, в которых опыт применения онтологий насчитывает от 5 до 10 лет.

1. Создание и использование БЗ. Хотя в ИИ использование БЗ декларируется уже 30—40 лет, их практическое формирование, на наш взгляд, началось с развитием онтологического подхода. Одним из ключевых аспектов технологии БЗ является создание алгоритмических и программных средств построения онтологий и работы с ними.

Онтологии позволяют формировать модели ПрО, интегрируя декларативные описания и определения (в том числе тексты на предметном языке экспертов). Выделяют следующие основные требования экспертов в прикладных областях к средствам построения онтологий [183]:

- близость языка, которым оперирует система, основанная на знаниях, к профессиональному языку специалиста ПрО;
- возможность использования введенных знаний для решения большинства (в пределе всех) предметных задач, а не формирование БЗ заново каждый раз для постановки и решения новой задачи;
- открытость языка, т. е. возможность включения в него новых языковых конструкций, которые появляются в данной ПрО.

Со стороны программистов, реализующих НИТ, также предъявляется ряд требований к языкам представления знаний в рамках онтологий:

- универсальность языка, т. е. возможность представления знаний разного типа независимо от ПрО;
- унифицированность языковых конструкций, обеспечивающая возможность разным ИАС обмениваться знаниями;
- наличие эффективных алгоритмов разбора языковых выражений.

Язык — ключевой компонент в каждой области знаний. Он формируется в процессе деятельности специалистов и отражает развитие понятийного аппарата ПрО. Существует несколько *языков представления знаний,* удовлетворяющих названным требованиям. К их числу относится получивший достаточное распространение язык Knowledge Interchange Format (KIF), предназначенный для межмашинного обмена знаниями и взаимодействия программ.

2. Организация эффективного поиска в БД, информационных каталогах, БЗ. Совершенствование механизмов поиска по ключевым словам и формальных языков запросов не избавляет от высокого уровня информационного шума и неполноты получаемых результатов. Идеальным решением было бы формулирование запросов на ЕЯ. Однако приемлемых методов, реализующих такой подход, в настоящее время нет. Использование онтологий позволяет *точнее интерпретировать смысл терминов,* фигурирующих в запросах, а также *дополнять или расширять запрос* понятиями, которые связаны с терминами запроса отношениями род—вид, синоним, часть–целое, ассоциация и др. (рис. 5.7). Подобное расширение запроса преследует цель уменьшения неполноты ответа на него.

С другой стороны, в современных ИПС и поисковых машинах Internet онтологии используются для *уточнения смысла запросов путем «фильтрации» их содержания,* что способствует уменьшению информационного шума. При этом применяются процедуры формирования так называемых *профилей информационных интересов пользователей* и процедуры семантического пересечения запроса или информации, приготовленной к выдаче, с

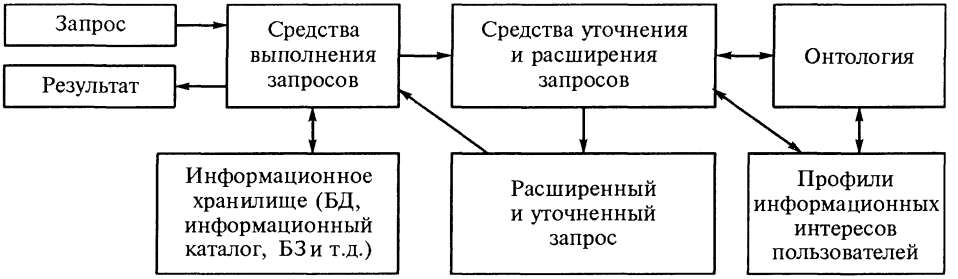


Рис. 5.7. Использование онтологий для организации эффективного поиска в БД, информационных каталогах, БЗ

этими профилями. В первом случае семантическое представление запроса, расширенное с помощью онтологии, сопоставляется с профилем информационных интересов, «фильтруется» и передается поисковой машине. Во втором случае «фильтрации» подвергается не запрос, а результат его выполнения, т. е. найденная по нему информация.

3. Создание систем, реализующих механизмы рассуждений (ЭС, системы управления, интеллектуальные роботы и др.). Подобные системы активно разрабатываются в последние годы, и потребность в них быстро растет. Прежде всего к ним относятся ЭС для технической и особенно медицинской диагностики. Обязательным компонентом таких систем является блок объяснения решения. Объяснения должны раскрывать суть возникающих ситуаций и их причины, а также обосновывать предлагаемые сценарии действий. Наличие объяснений позволяет человеку действовать осмысленно, а не слепо доверять ИАС. И при принятии решения, и при объяснении должна учитываться семантика как отдельных терминов, так и составленных из них высказываний и их композиций. Достижению данной цели способствует использование онтологий. Реализация средств формирования объяснений на базе онтологического подхода позволяет существенно снизить уровень ошибок, обусловленных человеческим фактором. И чем сложнее система, тем актуальнее такой путь.

4. Организация поиска по смыслу в текстовой информации. Текстовая информация до сих пор является основой документооборота. Ее объем очень велик, а задачи поиска и систематизации ответственны и сложны. Механизм индексирования текстовых документов весьма трудоемок. К тому же он не решает проблем неполноты и поискового шума. Это особенно проявляется при использовании запросов типа «где» и «как», а также фактографических запросов.

Для организации поиска по смыслу в текстовой информации необходимы методы извлечения семантики из текстовых документов и запросов и сопоставления получаемых семантических представлений. Подобные методы

также повысят эффективность автоматического реферирования, аннотирования и классификации документов, позволят автоматизировать построение ГТ. Новыми задачами, связанными с извлечением знаний из текста, являются:

- формирование сообщений на заданную тему;
- извлечение новых фактов по интересующей теме;
- реализация виртуального собеседника.

Роль онтологий в рамках данной проблематики была охарактеризована выше.

5. Семантический поиск в Internet. Одной из центральных проблем Internet является организация эффективного поиска информации. Онтологии позволяют формировать *информационные профили узлов сети* и на этапе предварительного отбора подходящих для поиска узлов отсеивать нерелевантные узлы (рис. 5.8).

Существуют идеи выделения *семантических областей Internet* с описанием на онтологическом уровне их информационных профилей. Подобная организация, базирующаяся не на географическом, а информационно-профильном принципе, позволяет на порядок снизить как время поиска, так и нагрузку на сеть.

Общей целью практически всех проектов в данной области является разработка новых подходов к построению *пространств знаний* Internet и средств работы с ними, которые бы обеспечивали:

- использование семантики при управлении процедурами выполнения запросов;
- возможность формирования ИР, содержащих компоненты, формально представляющие семантику и обладающие простым синтаксисом, которые могут интерпретироваться программными агентами и другими программными системами;
- гомогенный доступ к информации, которая физически распределена и гетерогенно представлена в Internet;
- возможность получения информации, которая явно не присутствует среди фактов, извлеченных из сети, но может быть выведена из этих фактов и базовых знаний, зафиксированных в онтологии.

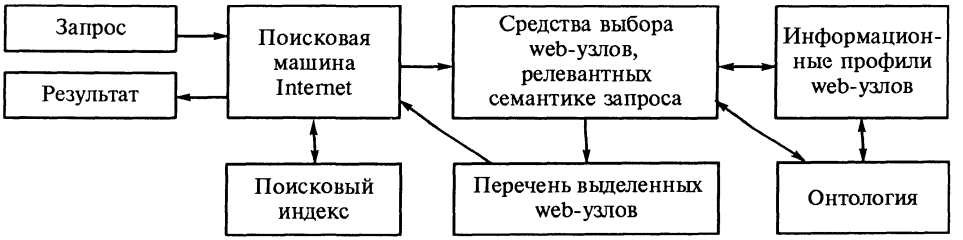


Рис. 5.8. Использование онтологий для организации семантического поиска в Internet

6. Представление смысла в метаданных об ИР. Современные языки представления метаданных, как правило, строятся на базе языка XML и модели RDF. В рамках данной задачи онтологии применяются при формировании пространств имен, словарей и квалификаторов для обеспечения их единообразных интерпретаций. Заметим, что без RDF (или другой модели такого рода) конструкциям, основанным на XML, недостает семантической выразительности.

В модели RDF используется объектно-ориентированная система классов. Базовая модель включает три типа объектов:

- ресурс, идентифицируемый URI;
- свойство, описывающее ресурс (в таком качестве также может выступать ссылка на другой ресурс или описание другого ресурса);
- утверждение, состоящее из ресурса-субъекта, свойства (ресурса-объекта) и предиката, связывающего их и представляющего значение свойства.

Методология *управления знаниями* (Knowledge Management) при использовании онтологического подхода позволяет решать задачи каталогизации и классификации ИР (в том числе неструктурированной информации) путем создания аналитических метаданных. Для этого применяются стандартизованные открытые интерфейсы с общими структурами и определениями метаданных.

7. Построение и использование баз общих знаний (common knowledge) для различных интеллектуальных систем. Человек в процессе рассуждений использует не только знания, ассоциируемые с данной ПрО, но и знания более высокой степени общности. К таким знаниям относятся описания свойств пространства, времени, личности и т. п. *Знания верхнего уровня* (common knowledge) позволяют доопределять модели конкретных предметных ситуаций с учетом взглядов и роли человека. Эти знания представляются в онтологиях верхнего уровня (общих онтологиях, онтологиях общих знаний).

Степень общности отражаемых знаний служит основанием для выделения *трех уровней онтологий*:

- общих онтологий;
- предметных онтологий;
- онтологий задач.

Создание *онтологий общих знаний* под силу только крупным консорциумам. В одиночку их не построить. Подобные онтологии размещаются в Internet с поддержкой открытого доступа к ним. Приложения, основанные на частных предметных онтологиях и онтологиях задач, могут обращаться к онтологиям общих знаний для получения информации, выходящей за их рамки. Формирование онтологий общих знаний и обеспечение доступа к ним через Internet стало возможным только в последние годы.

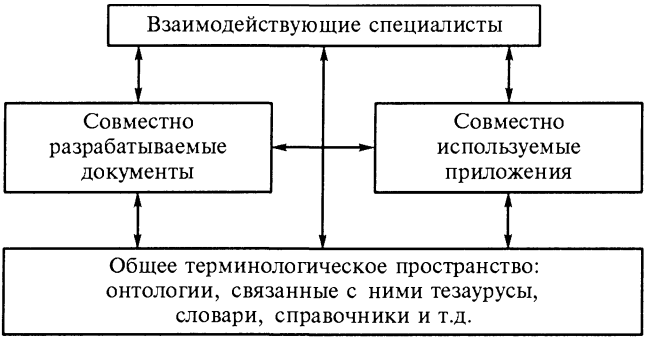


Рис. 5.9. Использование онтологий для представления общего понятийного пространства множества специалистов

8. Обеспечение общей терминологии для множества специалистов и совместно используемых приложений. Большинство практических задач относятся не к одной, а к нескольким ПрО. Такие задачи, как правило, решаются в рамках совместной деятельности группы специалистов, имеющих разную предметную подготовку. Члены группы могут взаимодействовать друг с другом с помощью телекоммуникационных технологий. Все это требует формирования общих понятийных пространств, обеспечивающих адекватное понимание информации, которой обмениваются специалисты (рис. 5.9). Онтологический подход существенно упрощает решение данной проблемы. Очевидно, что в данном случае применяются онтологии, относящиеся ко всем трем перечисленным выше уровням.

9. Многократное применение БЗ и информационных массивов, представляющих сведения о технических системах на различных стадиях их жизненного цикла. Данная задача приобрела актуальность в связи с развитием *CALS-технологий,* в рамках которых жизненный цикл технической системы (технического объекта) рассматривается с единых позиций, начиная с момента выявления потребности и до момента прекращения эксплуатации объекта и его утилизации. Каждый этап жизненного цикла должен быть обеспечен соответствующей информационной моделью. Важно, чтобы эта модель не создавалась каждый раз заново, а передавалась с этапа на этап, доопределяясь и развиваясь на каждом этапе.

Состояния единой информационной модели для всех этапов жизненного цикла технической системы сохраняются в репозитории и используются при решении задач анализа, формирования статистики и прогнозирования. Методология CALS обеспечивает значительные экономические выгоды, позволяет избежать ошибок, связанных с согласованием различных информационных моделей, способствует повышению качества принимаемых решений. Роль онтологий в рамках CALS-технологий аналогична той,

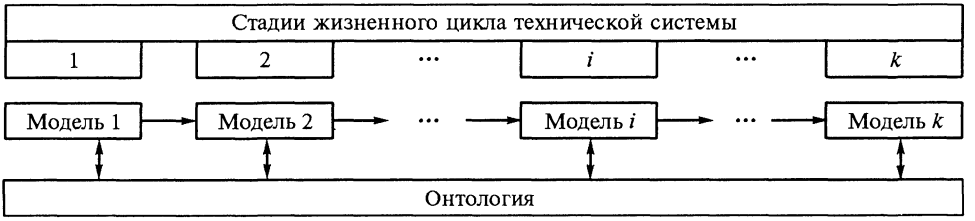


Рис. 5.10. Использование онтологий для согласования информационных моделей технической системы на всех стадиях ее жизненного цикла

что была охарактеризована применительно к предыдущей задаче. В данном случае онтология описывает техническую систему на разных этапах ее жизненного цикла и выступает в качестве базиса для построения и трансформации информационных моделей, БД и документов (рис. 5.10). Наличие онтологии обеспечивает их согласование и адекватное понимание.

5.4.3. Модель онтологии

Определение онтологии как формального представления ПрО, построенного на базе концептуализации, предполагает выделение ее трех взаимосвязанных компонентов: таксономии терминов, описаний смысла терминов, а также правил их использования и обработки. Таким образом, ***модель онтологии*** O задает тройка [1]:

$$O = (X, R, \Phi), \tag{5.20}$$

где X — конечное множество концептов (понятий, терминов) ПрО, которую представляет онтология; R — конечное множество отношений между концептами; Φ — конечное множество функций интерпретации, заданных на концептах и (или) отношениях.

Как видим, модель (5.20) является разновидностью сетевой модели знаний.

На ранних этапах развития онтологического подхода для представления онтологий использовались языки исчисления предикатов первого порядка и другие языки математической логики. В настоящее время все чаще применяются языки нелогического типа, специально предназначенные для описания онтологий [185, 199]. Причина данной тенденции состоит в том, что языки математической логики не позволяют формально выразить все необходимые свойства онтологии ПрО. Следствием недостаточной семантической выразительности средств математической логики является громоздкость языковых конструкций, отражающих смысл онтологических соглашений. Языки исчисления предикатов первого порядка не позволяют

вводить в онтологии термины, имеющие высокий уровень общности и непосредственно не участвующие в описании ситуаций. В результате представление онтологии оказывается трудно обозримым даже в случае, когда входящие в нее термины могут быть декомпозированы на классы терминов, имеющих одинаковые свойства.

Специальные языки нелогического типа, предназначенные для описания онтологий, по своей семантике эквивалентны языкам исчисления предикатов. Однако они существенно легче в освоении и применении специалистами, не обладающими навыками формализации и программирования.

Граничные варианты (5.20) рассмотрены в [1]. Понятно, что X не должно быть пусто. Если $R = \varnothing$ и $\Phi = \varnothing$, то онтология вырождается в *модель простого словаря* V:

$$O = (X, \varnothing, \varnothing) = V. \qquad (5.21)$$

В (5.21) смысл терминов явно не выражается.

Другой крайний случай имеет место, когда $R = \varnothing$, а $\Phi \neq \varnothing$. При этом множество терминов X разбивается на два класса: $X = X_1 \cup X_2$, $X_1 \cap X_2 = \varnothing$, где X_1 – множество интерпретируемых терминов, X_2 – множество интерпретирующих терминов. Тогда $\forall x \in X_1 \ \exists \ y_1, y_2, \ldots, y_k \in X_2$, такие, что $x = f(y_1, y_2, \ldots, y_k)$, где $f \in \Phi$. Другими словами, смысл термина x «вычисляется» (интерпретируется) каждый раз при обращении к нему. Это позволяет уточнять семантику терминов динамически в зависимости от текущей ситуации.

Заметим, что:

- пустота пересечения X_1 и X_2 исключает циклическую интерпретацию;
- функции f от нескольких аргументов обеспечивают более полную интерпретацию по сравнению с функциями от одного аргумента;
- вид отображения $f \in \Phi$ влияет на выразительную мощность и практическую полезность онтологии;
- часть интерпретирующих элементов из X_2 (подмножество X_2'') также может задаваться процедурно, т. е. с динамически изменяемым смыслом ($X_2 = X_2' \cup X_2''$, $X_2' \cap X_2'' = \varnothing$).

Если $X_2'' = \varnothing$, то онтология включает *пассивный словарь* ПрО. В противном случае в ней используется *активных словарь* ПрО. В обоих вариантах в отличие от (5.21) смысл терминов задается явно.

Активные словари ПрО находят применение при поиске информации в Internet, для представления пространств имен XML и в других задачах.

Третий крайний вариант (5.20) возникает, если $R \neq \varnothing$, а $\Phi = \varnothing$. Пусть R содержит лишь один тип отношения *is-a* (вид–род). Тогда $O = (X, \{is\text{-}a\}, \varnothing) = T$ — *простая таксономия* или иерархическая система понятий. В этом

случае предполагается, что семантика понятий заранее фиксирована, а структура понятий онтологии задается деревом.

Выделим следующие направления обобщения частных случаев моделей онтологии:

- представление множества концептов в виде сетевой структуры;
- представление в R отношений, отражающих специфику конкретной ПрО, а также средств расширения R;
- использование декларативных и процедурных представлений элементов Φ и R, включая возможность определения новых интерпретаций.

Для спецификации пространств знаний, охватывающих несколько взаимосвязанных ПрО, предложена ***модель расширенной онтологии*** [1]:

$$O_{\text{расш}} = (O_m, \{(O_p, O_z)\}, \text{МВ}), \qquad (5.22)$$

где O_m — онтология верхнего уровня (метаонтология); O_p — предметная онтология; O_z — онтология задач ПрО; МВ — модель машины вывода.

Метаонтология O_m оперирует общими (т. е. не зависящими от конкретной ПрО) концептами и отношениями. Например, это такие общие понятия, как «объект», «свойство», «значение» и т. д. Метаонтология изменяется весьма незначительно и может считаться статической. Поэтому вывод на ее уровне достаточно эффективен. На уровне O_m представляются интенсиональные описания свойств предметных онтологий O_p и онтологий задач O_z.

В общем случае расширенная онтология содержит множество пар (O_p, O_z). Онтологии O_p и O_z задают экстенсиональную часть описания ПрО, а в МВ специфицируются критерии релевантности получаемой при поиске и выводе информации. Онтологии O_p и O_z должны быть открыты для пополнения.

Предметная онтология O_p содержит понятия, описывающие конкретную ПрО, семантически значимые для нее отношения, а также декларативные и процедурные интерпретации этих понятий и отношений. Если множество понятий в каждой O_p имеет специфику, то отношения более универсальны. Это позволяет выделить некоторое ядро отношений, достаточное для начального описания предметных онтологий. Данное ядро является открытым для пополнения. Обычно в него включают базовые типы семантических отношений.

В *онтологии задач* O_z в качестве понятий выступают типы решаемых задач, а отношения, как правило, специфицируют декомпозицию задач на подзадачи. Отметим, что для поисковой машины, решающей единственную задачу поиска релевантной запросу информации, онтологию задач могут представлять словарные модели, рассмотренные выше.

Машина вывода начинает работу при активации понятий или отношений, описывающих исходную ситуацию (задачу). Вывод на семантических

сетях, представляющих в общем случае онтологию, организуется как волновой процесс, использующий свойства отношений, выходящих из узлов, задающих исходную ситуацию. Критерием останова процесса является либо достижение целевой ситуации, либо превышение длительности времени, отведенного для решения задачи.

Итак, онтология — важная разновидность сетевой модели представления знаний и воплощающая ее технология. Укажем *основные аспекты специфики онтологического подхода.* Во-первых, это представление содержит как формальные, так и описательные (выражаемые на ЕЯ) компоненты. Если первое важно для логического вывода (новых знаний, решений, рекомендаций, оценок и т. п.), то второе — для представления человеку смысла, стоящего за каждым действием интеллектуальной системы. Во-вторых, для отражения семантики определяются все используемые термины, а это требует наличия спецификации общих терминов в рамках онтологии верхнего уровня. Фактически имеет место иерархия онтологий: онтологии общих знаний на вершине, затем предметные онтологии и онтологии задач. В-третьих, онтологический подход, как правило, предполагает общение ИАС с пользователями на языках, близких к естественным (формальные языки применяются программистами, реализующими оболочки для работы с онтологиями).

Онтологический подход предъявляет жесткие требования к технологичности его воплощения. Основные усилия ученых и специалистов, работающих в данной области инженерии знаний, сосредоточены именно в этом направлении.

5.4.4. Методики построения онтологий и требования к средствам их спецификации

В настоящее время известен лишь один стандарт, регламентирующий процесс разработки онтологий и связанных с этим исследований: IDEF5, хотя различными организациями онтологии создаются весьма активно. Существует множество предложений по методикам разработки онтологий. В рамках таких методик обычно выделяются следующие основные задачи.

1. Анализ целей создания и области применения создаваемой онтологии.
2. Построение онтологии.

2.1. Сбор и фиксация знаний о ПрО, включающие:

- определение основных понятий и их взаимоотношений в выбранной ПрО;
- создание точных непротиворечивых определений для каждого основного понятия и отношения;

• определение терминов, которые связаны с основными понятиями и отношениями;

• согласование перечисленных компонентов онтологии.

2.2. Кодирование, включающее:

• разбиение совокупности основных терминов, используемых в онтологии, на классы;

• выбор или разработку специального языка для представления знаний;

• формирование концептуализации в рамках выбранного языка представления знаний.

Наиболее проработанными методиками являются [186, 191, 194, 201, 211]. Так, в [186] описан *конструктор онтологий мультиагентных систем,* используемых при создании информационных моделей деятельности предприятий. В таких онтологиях представляются корпоративные знания по всем аспектам деятельности предприятия (маркетингу, менеджменту, финансам, конструированию, проектированию, производству и др.). Онтологии позволяют решить проблему интеграции знаний для обеспечения согласованной работы специалистов в разных ПрО. Поскольку корпоративные знания, как правило, не полны, разнородны, несогласованны и распределены по организации, онтологии рассматриваются как открытые мультиагентные системы, способные к самоорганизации и эволюции. Используемая схема создания онтологий включает четыре уровня: микромира, модели мира, базовых миров, производных миров.

Уровень микромира представляет знания о первичных понятиях и отношениях, описываемых сетевыми структурами данных произвольного вида (на базе двухсвязных списков).

Модель мира строится на основе понятий и отношений, представленных в модели микромира, и предоставляет пользователю средства формирования модели ПрО с помощью категорий «объект», «свойство», «сценарий действия», «отношение» и «атрибут». При этом объекты определяются как наборы свойств и функций, а каждое свойство (функция) есть ссылка на сценарии действий и набор атрибутов. Сценарий задается совокупностью свойств и телом, состоящим из набора действий. Отношения трактуются как определенные виды связей между всеми другими концептами, а атрибуты задаются элементарными компонентами типа «ссылка», «символьная строка», «целое число» и т. д. Кроме того, на этом уровне строятся такие понятия, как «агент», «цель», «роль», «задача», «знание», «инструмент», «результат» и др. Заметим, что каждое такое понятие — набор объектов и отношений микромира.

Данная модель получила условное название «модель Аристотеля», который первым определил, что «объекты — суть свойства». Она предполагает вполне определенную конструкцию мира (от объекта — к свойствам и

отношениям, от свойств — к сценариям действий или атрибутам, от действий — к отношениям и объектам) и механизм ее функционирования (свойства и отношения запускают процессы, процессы изменяют объекты и т. д.). В результате уровень модели мира определяет интерпретацию всех последующих базовых миров и, в свою очередь, интерпретируется микромиром.

Базовые миры создаются на основе моделей двух предыдущих уровней. Они рассматриваются как типовые схемы организаций или предприятий со своими наборами задач и сценариев их решения.

Производные миры связаны с оригинальными, а не типовыми решениями, которые требуется разработать и исследовать.

Стандарт онтологического исследования IDEF5 [201] подготовлен фирмой Knowledge Base Systems, Inc. в качестве проекта национального стандарта США. Процесс построения онтологии в рамках IDEF5 состоит из пяти основных этапов.

1. Изучение и систематизация начальных условий. Этот этап устанавливает основные цели и контекст разработки онтологии, а также распределяет роли членов проекта.

2. Сбор и накопление данных для построения онтологии.

3. Анализ и группировка собранных данных для облегчения согласования терминологии.

4. Начальное развитие онтологии. На этом этапе формируется предварительная онтология на основе систематизированных данных.

5. Уточнение и утверждение онтологии (заключительный этап).

Для поддержки процесса построения онтологий в IDEF5 определены специальные онтологические языки:

- схематический язык (Schematic Language — SL);
- язык доработок и уточнений (Elaboration Language — EL).

Язык SL является наглядным графическим языком, специально предназначенным для представления специалистами в ПрО совокупности основных сведений о ней в форме онтологической информации. Этот несложный язык обеспечивает формирование начального представления онтологии, а также дополнение существующих онтологий новыми данными. Он включает разнообразные типы диаграмм и схем, служащих для визуального представления основной онтологической информации.

Язык EL является структурированным текстовым языком, позволяющим детализировать элементы онтологии. С помощью него решаются задачи анализа и обеспечения полноты представления сведений, полученных в результате онтологического исследования.

В стандарте IDEF5 предусмотрены четыре основных вида схем, предназначенных для представления онтологической информации в наглядной графической форме.

Диаграммы классификации служат средством логической систематизации знаний, накопленных при изучении системы. Существует два типа таких диаграмм: диаграмма строгой классификации (Description Subsumption — DS) и диаграмма естественной или видовой классификации (Natural Kind Classification — NKC). Основное отличие диаграммы DS заключается в том, что в ней определяющие свойства класса являются необходимым и достаточным признаком принадлежности объекта этому классу.

С помощью диаграмм DS, как правило, классифицируются логические объекты. Диаграммы NKC, наоборот, не предполагают того, что свойства класса являются необходимым и достаточным признаком принадлежности ему тех или иных объектов. В диаграммах этого типа интерпретация свойств класса является более общей.

Композиционные схемы (Composition Schematics) служат для графического представления состава классов онтологии. В частности, с помощью них можно наглядно отобразить состав объектов, относящихся к тому или иному классу.

Схемы взаимосвязей (Relation Schematics) позволяют разработчикам визуализировать и изучать связи между различными классами объектов системы. В некоторых случаях эти схемы используются для представления зависимостей между взаимосвязями классов.

Диаграмма состояния объекта (Object State Schematic) позволяет описать процесс изменения состояния объекта. С объектом могут произойти два типа изменений: он может поменять либо свое состояние, либо класс. Между этими типами не существует принципиальной разницы: объекты, относящиеся к определенному классу в начальном состоянии, могут быть переведены в его дочерний или родственный класс. Например, полученный в процессе сжижения азот относится не к классу «Азот», а к его дочернему классу «Жидкий азот». Однако на уровне формального описания для исключения путаницы следует дифференцировать данные типы изменений. Для этого используются обозначения вида <класс>:<состояние>. Например, жидкий азот будет описываться как «азот:жидкий», газообразный азот – «азот:газообразный» и т. д. Таким образом, диаграммы состояния в IDEF5 наглядно представляют изменения состояния или класса объекта в рамках его жизненного цикла.

Стандарт IDEF5 отражает методологию, с помощью которой можно наглядно и эффективно разрабатывать онтологии. К сожалению, на сегодняшний день существуют единичные программные средства, поддерживающие IDEF5. Кроме того, данный стандарт охватывает не все этапы создания онтологий.

Еще одна методология построения онтологий описана в [1, 191, 194]. Для ее поддержки предназначена специальная *инструментальная среда проектирования онтологий* (Ontology Design Environment). Она включает подсистемы управления проектом и поддержки разработки. Первая подсис-

тема обеспечивает решение задач планирования, контроля за ходом выполнения проекта и управления качеством. Вторая ориентирована на задачи приобретения знаний, их оценки, интеграции, документирования и управления конфигурациями. Процесс разработки онтологии включает четыре стадии:

1) спецификация;
2) концептуализация;
3) формализация;
4) реализация.

Спецификации подлежат цели создания онтологии, ее предполагаемое применение и потенциальные пользователи. Концептуализация обеспечивает структурирование предметных знаний в рамках эксплицитной модели. На этапе реализации создается вычислительная модель, соответствующая онтологии и выражаемая на одном из языков представления знаний.

Наиболее сложной задачей является концептуализация. От успешности ее выполнения зависит эффективность всей разработки. Поэтому методике концептуализации уделяется особое внимание.

Концептуализация включает два этапа. В рамках первого решаются следующие задачи:

- построение глоссария терминов;
- построение классификационных деревьев концептов.

Вторая задача начинает решаться тогда, когда объем глоссария по мнению экспертов достигает существенного объема. Затем для каждого классификационного дерева формируются словарь концептов и совокупность таблиц, описывающих бинарные отношения между концептами, экземпляры, атрибуты экземпляров и классов, логические аксиомы, константы и формулы.

Программные решения для создания ИАС на основе онтологического подхода разрабатывает фирма Ontoprise GmbH (Карлсруе, Германия)*. Пакет *Smarter Knowledge Suite* включает следующие основные компоненты:

- редактор онтологий OntoEdit;
- машину вывода для онтологий OntoBroker;
- интеллектуальную ИПС SemanticMiner;
- систему OntoOffice, служащую интеллектуальной поисковой надстройкой над пакетом Microsoft Office.

OntoEdit представляет собой визуальный инструментарий для создания и редактирования предметных онтологий. Функциональность системы расширяется путем подключения к ней внешних программных компонентов (plug-ins). Сформированная онтология может быть выражена на языках RDF, OXML (формат OntoEdit), DAML+OIL, F-Logic (Frame Logic).

* http://www.ontoprise.de.

Коротко рассмотрим структуру онтологии, с которой оперирует OntoEdit. Ядром онтологии служит иерархия концептов (абстрактных понятий ПрО). Иерархические отношения соответствуют типу род–вид и используются в механизмах наследования. Для описания прочих, неиерархических типов связей между концептами предназначены бинарные отношения. Концептам приписываются атрибуты. Атрибут рассматривается как отношение определенного типа между концептом и значением. Реализации концептов, называемые экземплярами, представляют конкретные сущности.

Онтология также содержит аксиомы, под которыми понимаются правила, справедливые в моделируемой ПрО. Подобные правила выражаются на основе отношений. Правило может включать одно или несколько отношений. Например: «ЕСЛИ *X* <является отцом> *Y* И *Y* <имеет пол> <мужской>, ТО *Y* <является сыном> *X*». В этом примере *X* и *Y* — концепты, <является отцом> и <является сыном> — отношения между ними, а <имеет пол> — отношение, задающее для *Y* атрибут «пол».

Различные типы аксиом используются для описания ограничений целостности, накладываемых на онтологию, математических и функциональных связей, логических зависимостей и отношений между фактами.

Иллюстрации работы с редактором онтологий OntoEdit приведены на рис. 5.11 и 5.12.

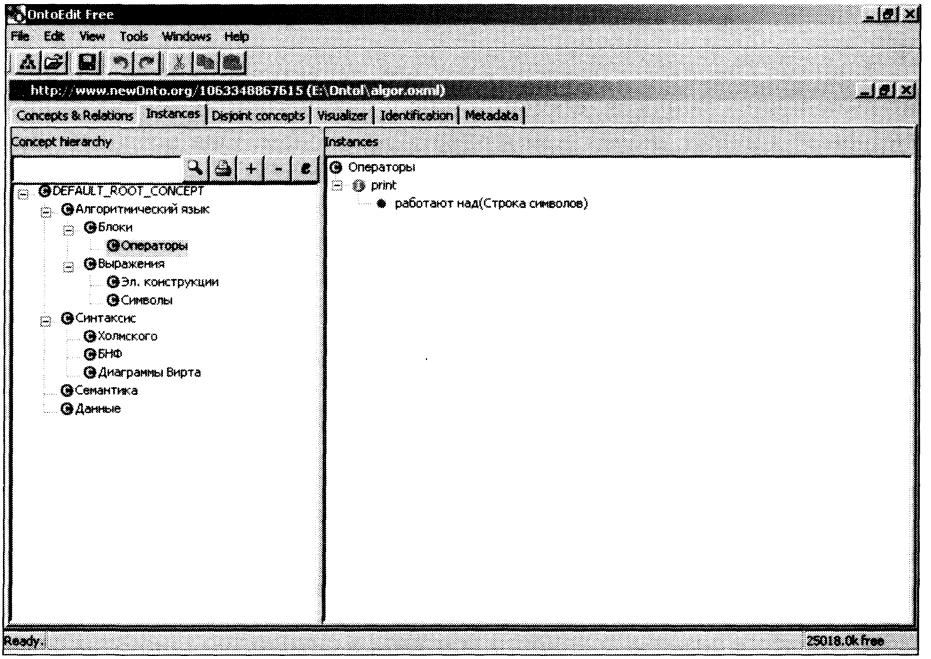


Рис. 5.11. Описание экземпляров в редакторе онтологий OntoEdit

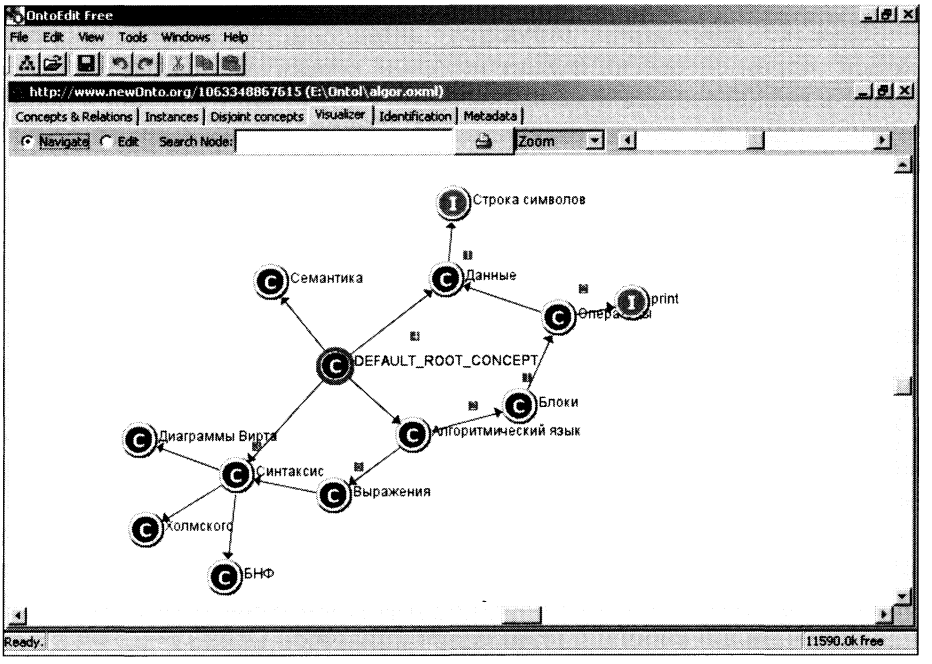


Рис. 5.12. Визуализация онтологии в редакторе OntoEdit

Система *OntoBroker* реализует машину дедуктивного вывода и интерфейс выполнения запросов, оперирующие с онтологией. В основе ее функционирования лежит анализ и интерпретация аксиом. Она позволяет верифицировать онтологию, выносить заключения о фактах и значениях атрибутов. Механизмы вывода не зависят от последовательности применения правил и последовательности утверждений внутри правил.

OntoBroker реализована на базе Java-технологии. Система может применяться в качестве автономного приложения или сервера выполнения запросов, использующего онтологию, а также библиотеки для средств дедуктивного вывода.

Интеллектуальная ИПС *SemanticMiner* построена в рамках архитектуры клиент-сервер. Ее основные функции:

- извлечение знаний из различных источников (текстовых документов, БД, web-страниц, экземпляров метаданных);
- формирование БЗ и ее визуализация;
- автоматическая кластеризация документов;
- поиск в БЗ и навигация по ней.

В процессе анализа исходного контента в онтологию заносятся ссылки на документы-источники. Это дает возможность автоматически распреде-

лять документы по кластерам. Онтология и связанные с ней механизмы вывода используются при поиске. Удобную навигацию по БЗ обеспечивают средства визуализации онтологии, позволяя «перемещаться» по ее элементам и выделять соответствующие им группы документов.

5.4.5. Обзор наиболее известных онтологических проектов

Как показывает анализ литературных источников, в настоящее время разрабатываются онтологии всех трех уровней: верхнего уровня, предметные и задачные. Однако наиболее активно создаются предметные онтологии.

Среди проектов *онтологий верхнего уровня* наиболее существенные результаты достигнуты в рамках *проекта Cyc* корпорации Cycorp (Техас, США). Данный проект направлен на формирование информационно-логической основы для создания и функционирования систем, реализующих механизм рассуждений. Эту основу составляет объемная онтология, содержащая описания общих понятий, отношений между ними и утверждений, считающихся априорно истинными (аксиом), а также машину вывода. Онтология *Cyc* базируется на ядре, включающем миллион утверждений, введенных в БЗ вручную. Версия *Cyc* с открытым исходным кодом (проект OpenCyc) на сегодняшний день содержит описания 6000 концептов и свыше 60000 аксиом, отражающих непротиворечивые знания о мире.

Машина вывода может применяться для получения новых утверждений. Знания в *Cyc* представляются на специально разработанном декларативном языке CycL в виде утверждений логики первого порядка.

Онтология *Cyc* организована в виде совокупности модулей, называемых микротеориями. Каждая микротеория фиксирует знания и утверждения о некоторой отдельной области, например, о пространстве, времени, причинности. Причем для одной области может существовать множество микротеорий, отражающих точки зрения специалистов, моделирующих эту область. В этом смысле *Cyc* является не монолитной онтологией, а сетью микротеорий для множества областей, представляющих разные онтологические сообщества, сформировавшиеся в этих областях.

Другим примером онтологии верхнего уровня является модель *GUM* (Generalized Upper Model)*. Цель проекта GUM — поддержка обработки ЕЯ (английского, немецкого и итальянского). Модель GUM развивает идеи онтологии Penman Upper Model. Уровень абстракции при представлении онтологии GUM лежит между лексическими и концептуальными знаниями. Это позволяет упростить создание на ее базе естественно-языкового интерфейса. Модель GUM содержит раздельные таксономии понятий и связей.

* http://www.darmstadt.gmd.de/publish/komet/gen-um/newUM.html.

Онтологии предметного уровня разрабатываются в рамках *проекта TOVE* (Toronto Virtual Enterprise), целями которого являются:

- создание средств представления общей терминологии для ПрО, приложения которой могут совместно использоваться и адекватно пониматься каждым членом сообщества;
- построение точных и непротиворечивых определений терминов на основе логики первого порядка;
- обеспечение возможностей описания семантики с помощью множества аксиом, которые автоматически позволяют получать ответы на вопросы о ПрО.

Онтология TOVE представляет собой интегрированную модель ПрО, состоящую из онтологий операций, состояний и времени, организации, ресурсов, продуктов, сервиса, производства, цены, количества.

Проект KACTUS выполняется в рамках программы Европейского Союза ESPIRIT [187]. Его цель заключается в создании методологии многократного применения знаний о технических системах на протяжении их жизненного цикла. Другими словами, данная методология позволяет использовать одни и те же БЗ при проектировании, оценке, эксплуатации, сопровождении, репроектировании и обучении. KACTUS поддерживает интегрированный подход, охватывающий как производственные и технологические методы, так и методы инженерии знаний на базе создания онтологической и вычислительной основы для многократного использования полученных знаний параллельно разными приложениями. Это достигается путем построения онтологий ПрО, ориентированных на все этапы жизненного цикла технических систем. Кроме того, в KACTUS сделана попытка объединить эти онтологии с существующими стандартами (например, STEP [202]).

Основным формализмом в KACTUS является язык CML (Conceptual Modeling Language). В рамках KACTUS создан комплекс инструментальных средств для просмотра, редактирования и управления онтологиями. С помощью него можно просматривать, редактировать и анализировать онтологии на основе разных формализмов, организовывать библиотеки онтологий, обмениваться данными между онтологиями, выполнять их преобразования и др.

Проект (KA)^2 — аннотация знаний сообществом приобретения знаний (Knowledge Annotation Initiative of the Knowledge Community) — направлен на развитие технологий интеллектуального поиска в Internet и автоматического извлечения знаний из web-ресурсов. В рамках данного проекта выделяются три направления исследований [190]:

1) онтологический инжиниринг;

2) автоматическое аннотирование web-страниц;

3) построение и выполнение запросов, относящихся к информации, представленной на web-страницах, и вывод ответов с использованием онтологических знаний.

Первое направление является основополагающим. Предполагается, что сообщество $(KA)^2$ должно разработать собственную достаточно общую систему онтологий с помощью средств Ontolingua (см. далее). На сегодняшний день построены восемь онтологий, рассматриваемых как разделы общей онтологии: онтологии организации, проекта, персоны, направления исследований, публикации, события (например, конференции, семинара и т. д.), исследовательского продукта и исследовательской группы.

Экземпляры онтологий размещаются на web-страницах. Такие страницы аннотируются с помощью нового типа HTML-тэгов (ONTO), информация в пределах которых обрабатывается специальным средством – системой Ontocrawler. При этом в зависимости от выразительных средств используемой онтологии может выводиться новая информация, релевантная запросу, но в явном виде не присутствующая на web-страницах.

Система Ontocrawler создается в рамках *проекта Ontobroker* инициативы $(KA)^2$. Программная реализация Ontobroker включает три основные подсистемы:

1) интерфейс построения запросов (query interface);

2) машину вывода ответов (inference engine);

3) машину доступа к Internet-ресурсам, служащую для извлечения знаний из Internet и их накопления.

Для спецификации онтологий разработан специальный язык представления знаний. Подмножество этого языка служит для формирования запросов, а язык аннотирования — для снабжения web-документов онтологической информацией.

В рамках *проекта SHOE* (Simple HTML Ontology Extensions) [206] создаются средства аннотирования web-страниц, позволяющие вносить в них семантическое содержание, доступное для обработки интеллектуальными агентами. Для этого SHOE дополняет HTML набором специальных тэгов для представления знаний, отличных от тэгов HTML.

Основным компонентом SHOE является онтология, представляющая сведения о некоторой ПрО. Используя эти сведения, средства поиска информации и выполнения запросов обеспечивают получение более релевантных результатов по сравнению с существующими поисковыми машинами. SHOE позволяет находить знания с помощью таксономий и правил вывода, представленных в онтологии.

Система *Ontolingua,* разработанная в Knowledge System Laboratory (KSL) Отделения информатики Стенфордского университета, представляет собой инструментальное средство для совместного формирования, просмотра, редактирования и использования онтологий в среде WWW [182, 204, 207]. Язык Ontolingua реализует принципы объектно-ориентированного подхода и является расширением языка KIF.

На сервере Ontolingua представлен ряд онтологических библиотек по различным областям знаний. С помощью системы Ontolingua KSL выполняет проект по формированию широкомасштабного репозитория выразительных и многократно используемых знаний (Large Scale Repository of Highly Expressive Reusable Knowledge). Инструментарий поддерживает взаимодействие групп разработчиков, тестирование и выявление ошибок в онтологиях, объединение онтологий, поиск знаний на основе онтологий, а также перевод информации с некоторых языков представления знаний в формат Ontolingua.

5.4.6. Примеры использования онтологий

Рассмотрим простые примеры использования онтологий при решении задач информационного поиска и уточнения смысла естественно-языковых суждений. Эти примеры имеют условный характер. Они иллюстрируют роль онтологий, но не раскрывают механизм их применения.

Пример 1. Исходный запрос: «Какие существуют фазовые состояния?». Для расширения запроса система использует отношение род–вид (R_1), экземпляры которого представлены в онтологии:

R_1(«фазовое состояние», «твердое вещество»);

R_1(«фазовое состояние», «жидкость»);

R_1(«фазовое состояние», «газ»);

R_1(«фазовое состояние», «плазма»).

Ответом на запрос служит множество {«твердое вещество», «жидкость», «газ», «плазма»}.

Пример 2. Исходный запрос: «электрическая емкость». Для расширения запроса система использует отношение синонимии R_s, отраженное в онтологии: R_s(«электрическая емкость», «электрический конденсатор»). Расширенный запрос имеет вид: «электрическая емкость» $\vee$ «электрический конденсатор».

Пример 3. Исходный запрос: «электромагнитная индукция». Для расширения запроса система использует отношение ассоциации R_a, представленное в онтологии: R_a(«электромагнитная индукция», «Фарадея — Максвелла — Ленца закон»). Расширенный запрос имеет вид: «электромагнитная индукция» $\vee$ «Фарадея — Максвелла — Ленца закон».

Пример 4. Исходный запрос: «электронное облако». Для дополнения запроса система использует отношение часть—целое (R_2), отраженное в онтологии: R_2(«электронное облако», «атом»). Уточненный запрос имеет вид: «электронное облако» & «атом».

Средства обработки запросов, соответствующих примерам 1—4, могут быть реализованы на основе онтологии физико-технических характери-

стик, развиваемой в рамках проекта по созданию виртуального фонда естественнонаучных и научно-технических эффектов «Эффективная физика»* [184]. В настоящее время данная онтология включает более 1200 свойств и отношений, представленных в виде таксономий с определенными на них ограничениями целостности, отражающими совместимость характеристик. Описание каждого эффекта в виртуальном фонде имеет содержательную и формализованную части. Формализованное описание эффекта включает описания его входной и выходной систем, а также преобразования входной системы в выходную. Обе системы состоят из объектов и отношений, представляемых наборами характеристик, выбираемых из таксономий, уровни которых соотносятся с разными степенями определенности (общности) свойств и отношений. Выбранные характеристики могут быть положительными или отрицательными и выражаться качественно или количественно.

Рассмотрим примеры, иллюстрирующие использование онтологий для уточнения смысла естественно-языковых высказываний.

Пример 5. Имеются два похожих предложения на английском языке:

- Fred saw the plane flying over Zurich;
- Fred saw the mountains flying over Zurich.

В обоих случаях программа грамматического разбора порождает две синтаксические схемы. В первой фраза «flying over Zurich» относится к подлежащему «Fred», во второй — к дополнениям «plane» для первого предложения и «mountains» для второго. Средства грамматического разбора не позволяют принять решение, какой вариант для какого предложения является верным.

Преодоление подобной проблемы обеспечивает использование онтологии общих понятий. В частности, в онтологии *Cyc* термину «plane» (самолет) приписано свойство «flying» (полеты). Между словом «Fred» (имя человека) и свойством «flying» явная связь не установлена. В то же время она не исключается, поскольку человек может летать с помощью технических средств. В свою очередь, между термином «mountain» (гора) и свойством «flying» связи нет и быть не может (несовместимость объектов и свойств также отражается в онтологии). Таким образом, пользуясь знаниями о понятиях и их свойствах, представленных в онтологии, система в состоянии установить, что в первом предложении, скорее всего, говорится о том, что Фред видел самолет, летающий над Цюрихом, а во втором о том, что Фред видел горы, сам летая над Цюрихом.

Пример 6. В библиографическом описании учебного пособия указано, что оно «предназначено для студентов вузов со знанием алгебры» (фраза взята из ГОСТ 7.82-2001). При грамматическом разборе данной фразы воз-

* http://www.effects.ru.

никает неопределенность: к каким сущностям (студентам или вузам) относится «со знанием алгебры»?

Онтология общих понятий фиксирует, что вузы не могут знать алгебру, а студенты могут знать. Поэтому «со знанием алгебры» отражает требование к подготовленности студентов.

Пример 7. Фраза из отчета о футбольном матче: «Футболисты покинули поле без голов». Разрешение присущей ей двусмысленности формальными средствами достаточно непросто. Онтология общих понятий позволяет установить две группы отношений. В первую входят отношения R_3(«футболист», «человек») и R_4(«человек», «голова»), где R_3 — отношение вид—род, R_4 — отношение целое—часть. Вторую группу образуют отношения R_a(«футболист», «игра»), R_a(«футболист», «гол»), R_a(«игра», «гол»), R_a(«игра», «поле»), R_a(«футболист», «поле»). Вторая группа является более сильной, так как входящие в нее ассоциации представляют общую ПрО (футбол) и покрывают большее число терминов предложения (футболисты, поле, голы). С учетом сказанного система может сделать заключение, что по всей видимости в предложении имеется в виду, что футболисты покинули поле, не забив мячей, а с их головами ничего страшного не случилось.

Основные выводы

1. Онтология представляет собой точную (выраженную формальными средствами) спецификацию концептуализации ПрО. На метауровне онтология является разновидностью сетевой модели знаний о ПрО.

2. Онтологии как способ представления знаний обеспечивают их согласование и интеграцию. Они позволяют зафиксировать согласованные интерпретации применяемых терминов. В этом плане важную роль играют механизмы интеграции онтологий.

3. К основным задачам, решаемым с помощью онтологий, относятся создание и использование БЗ, организация семантического поиска в БД, БЗ и Internet, реализация механизмов рассуждений для различных классов ИАС, представление смысла в метаданных об ИР и др.

4. Выделяют три уровня онтологий: общие (верхнего уровня), предметные и задачные.

5. Процесс разработки онтологий регламентирован стандартом IDEF5.

6. В настоящее время известно несколько методологий создания онтологий, поддерживаемых соответствующими языковыми и инструментальными средствами (например, Ontology Design Environment, Smarter Knowledge Suite и др.). Однако ни одна из них не достигла уровня стандарта де-факто.

7. К числу наиболее значимых международных проектов создания онтологий верхнего уровня относятся проекты *Cyc* и GUM.

8. Проведенный анализ показывает, что онтологические технологии активно развиваются и находят все новые области применения. Их главное назначение — повышение уровня интеллектуальности различных классов ИС за счет отражения семантики представляемых в их БД понятий, атрибутов и отношений. Это способствует превращению БД в БЗ, в которых можно вести поиск по смыслу, а также выводить на их основе новые знания, изначально не представленные в них в явном виде.

Вопросы для самопроверки

1. Что понимается под концептуализацией?
2. Охарактеризуйте различные интерпретации понятия «онтология».
3. Какие основные классы задач решаются с использованием онтологий?
4. Какова роль онтологий в методах поиска информации по смыслу (в том числе при поиске в Internet)?
5. Как используются онтологии в БЗ?
6. Как используются онтологии для представлении смысла в метаданных об ИР?
7. Что может обеспечить онтология в ИАС, поддерживающей взаимодействие множества специалистов?
8. Какова роль онтологий в CALS-технологиях?
9. Как представляется модель онтологии?
10. Каким видам ИС соответствуют граничные варианты модели онтологии?
11. Что такое модель расширенной онтологии? Охарактеризуйте ее компоненты.
12. Какие этапы построения онтологии предусмотрены стандартом IDEF5?
13. Для чего предназначены языки SL и EL?
14. Какие типы диаграмм предусмотрены в IDEF5?
15. Какие этапы построения онтологии поддерживаются инструментальной средой Ontology Design Environment?
16. Какие продукты включает пакет Smarter Knowledge Suite?
17. Опишите структуру онтологий, с которыми работает редактор OntoEdit?
18. Каково назначение онтологий верхнего уровня? Приведете примеры таких онтологий.
19. Каково назначений онтологий предметного уровня? Приведете примеры таких онтологий.

5.5. Основы технологии баз знаний*

Чтобы разумно действовать, одного разума недостаточно.

Ф.М. Достоевский

5.5.1. Общие положения

Создание БЗ и в теории, и в практике ИИ сегодня является проблемой такой же важности, как в свое время в информационных технологиях проблема создания БД.

Под базой знаний понимается семантическая модель, предназначенная для представления в ЭВМ знаний, накопленных человеком в определенной ПрО**. На технологическом уровне БЗ рассматривается как хранилище (репозиторий) сложно структурированных информационных единиц (знаний).

Базы знаний подразделяются на замкнутые и открытые [10]. Интерпретация содержимого *замкнутой БЗ* в процессе функционирования включающей ее интеллектуальной системы не изменяется. Логический вывод в такой БЗ эквивалентен выводу в формальной системе и обладает свойством монотонности.

Противоположные черты присущи *открытой БЗ.* Охватывающая ее интеллектуальная система может пополнять и модифицировать содержимое БЗ, а также удалять знания из нее. Вывод в открытой БЗ является немонотонным.

Говоря о БЗ, мы всегда будем соотносить ее со знаниями о некоторой ПрО (одной или нескольких). При этом под ПрО может пониматься и некоторый класс решаемых задач.

По аналогии с технологией БД будем различать собственно информационное хранилище знаний (БЗ) и ***систему управления БЗ (СУБЗ)***, обеспечивающую набор типовых функций хранения и манипулирования знаниями. С парой БЗ—СУБЗ может взаимодействовать прикладная интеллектуальная система, использующая содержимое БЗ и средства СУБЗ для решения каких-либо предметных задач.

Обобщенная структура БЗ изображена на рис. 5.13. Математически она представляется шестеркой

---

* Содержание параграфа соответствует направлениям исследований в области ИИ 2.1.2 и 2.3.

** *Першиков В.И., Савинков В.М.* Толковый словарь по информатике. — М.: Финансы и статистика, 1991. С. 31.

$$(M_1, M_2, M_3, I_1, I_2, I_3), \qquad (5.23)$$

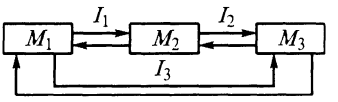


Рис. 5.13. Обобщенная структура БЗ

где M_1 — база глубинных знаний, представляющая понятийные структуры ПрО; M_2 — база фактов; M_3 — база метазнаний; I_1 — интерфейсы между M_1 и M_2; I_2 — интерфейсы между M_2 и M_3; I_3 — интерфейсы между M_1 и M_3.

База глубинных знаний M_1 состоит из двух компонентов:

$$M_1 = (M_{11}, M_{12}). \qquad (5.24)$$

Здесь M_{11} — часть хранилища знаний, содержащая описания единиц знаний, образующих понятийные структуры ПрО; M_{12} — сеть фреймов над понятийными структурами.

База фактов M_2 соответствует части хранилища знаний, содержащей эмпирические данные о ПрО, параметры наблюдаемых ситуаций и т. д.

База метазнаний включает три компонента:

$$M_3 = (M_{31}, M_{32}, M_{33}), \qquad (5.25)$$

где M_{31} — база правил для данной ПрО; M_{32} — база метаправил, метаметаправил и т. д.; M_{33} — стратегия управления правилами и метаправилами.

Интерфейсы I_1, I_2 и I_3 представлены парами компонентов, соответствующими направленности связей между взаимодействующими блоками БЗ:

$$I_1 = (I_{11}, I_{12}), \qquad (5.26)$$
$$I_2 = (I_{21}, I_{22}), \qquad (5.27)$$
$$I_3 = (I_{31}, I_{32}). \qquad (5.28)$$

Здесь I_{11} — интерфейс, связывающий M_1 и M_2; I_{12} — интерфейс, связывающий M_2 и M_1; I_{21} — интерфейс, связывающий M_2 и M_3; I_{22} — интерфейс, связывающий M_3 и M_2; I_{31} — интерфейс, связывающий M_1 и M_3; I_{32} — интерфейс, связывающий M_3 и M_1.

Наиболее сложной проблемой является представление глубинных знаний (M_1). Технология построения M_1 непосредственно связана с выбором модели представления знаний о ПрО. Соответствующие вопросы были рассмотрены в предыдущих параграфах данной главы. В настоящее время для организации M_1 используется технология объектно-ориентированных БД [223]. База фактов M_2, как правило, реализуется на основе технологии реляционных БД. Поэтому I_1 — это интерфейсы между реляционными и объектно-ориентированными моделями.

Для построения базы метазнаний M_3 в последние годы все чаще используются семантические сети и онтологии.

5.5.2. Система операций для работы со знаниями в базе знаний

Остановимся на важных, но недостаточно освещенных в литературе вопросах формирования системы операций для работы со знаниями. Рассмотрим подходы к решению этой проблемы на примере обобщенной модели представления знаний о ПрО *М*4, охарактеризованной в § 5.3.

Система операций для работы со знаниями в БЗ является *многоуровневой.* Первый уровень образуют интерфейсные операции, обеспечивающие ввод и коррекцию знаний в БЗ в процессе диалога с пользователем интеллектуальной системы или приема информации из иных источников. На втором уровне располагаются элементарные операции, отражающие специфику взаимосвязи базисных компонентов информационных структур (вещей, свойств и отношений). Третий уровень содержит комплексные операции. К ним относятся операции верификации БЗ (выявление ошибок и неточностей, разрешение противоречий), а также операции поиска, извлечения, пополнения и систематизации знаний.

Далее будут рассмотрены некоторые наиболее важные операции второго и третьего уровней.

5.5.3. Элементарные операции

К *операциям второго уровня* относятся различные виды абстракции, конкретизации, формализации и интерпретации. Данные операции представляют собой отражение принципа взаимоперехода вещей, свойств и отношений. На основе элементарных операций строятся другие механизмы обработки знаний.

К системе операций второго уровня предъявляются три основных требования:

1) полнота в смысле формальной логики;

2) обеспечение обработки знаний на разных ступенях детальности их представления;

3) работа с единым набором информационных структур (вещь, свойство, отношение).

Операции перехода от вещей к свойствам и от свойств к вещам осуществляются с помощью широко известных логических приемов абстракции и конкретизации. ***Абстракцией*** называют результат мысленного отвлечения (абстрагирования) тех или иных определенных свойств от множества свойств исследуемого конкретного предмета. Абстракция может выступать в формах чувственно-наглядного образа, суждения, понятия, категории.

В самой широкой трактовке абстракция представляет собой переход от одной вещи или множества вещей к другой — абстрактной вещи или со-

вокупности вещей, обладающих выделенными общими свойствами исходных вещей. Для формального выражения операции абстракции используем обозначения вещей, в которых характеризующие их свойства будут указываться в скобках за символами самих вещей, например: $A_i(P_{i1}, P_{i2}, \ldots, P_{in})$ — вещь A_i, обладающая свойствами $P_{i1}, P_{i2}, \ldots, P_{in}$.

Специфика *первой разновидности абстракции* (5.29) заключается в том, что среди отбрасываемых свойств вещи A_i нет ни одного качества, т. е. качественные определенности исходной конкретной и абстрактной вещей совпадают (рис. 5.14, *а*). В этом смысле (5.29) можно сравнить с рассмотрением изучаемой вещи через лупу или микроскоп, когда только подчеркиваются некоторые особо интересующие исследователя свойства, но еще не делаются обобщающие выводы об их сути:

$$L_a^{1}[A_i(P_{i1}, P_{i2}, \ldots, P_{in})] \rightarrow A_i(P_{i1}, P_{i2}, \ldots, P_{im}), \qquad (5.29)$$

где L_a^{1} — оператор абстрагирования типа 1; n — количество свойств исходного представления вещи A_i, $n > 1$; m — количество свойств формируемого представления вещи A_i, $1 \le m < n$; символ $\rightarrow$ обозначает переход от одного представления вещей к другому.

При *абстракции второго типа* (5.30) проводится отвлечение как от несущественных свойств вещи, так и от ее качеств (рис. 5.14, *б*):

$$L_a^{2}[A_i(P_{i1}, P_{i2}, \ldots, P_{in})] \rightarrow A_o(P_{o1}, P_{o2}, \ldots, P_{om}). \qquad (5.30)$$

Здесь L_a^{2} — оператор абстрагирования типа 2; A_o — абстрактная вещь-носитель выделенных свойств $P_{o1}, P_{o2}, \ldots, P_{om}$.

Вырождение схемы (5.30), заключающееся в отвлечении от определенного носителя выделенных свойств и переходе к подразумеваемой вещи-переменной, отражается в *третьем варианте абстракции* (рис. 5.14, *в*). Трактовка подобной разновидности абстракции полностью идентична предыдущему случаю, а сам третий тип введен нами исключительно по причине его связи с обратным абстракции процессом конкретизации, обсуждаемым позже. Формальное выражение абстракции типа 3 может быть получено из (5.30) путем замены абстрактной вещи A_o на переменную (неопределенную) подразумеваемую вещь X и переобозначения индексов ее свойств с P_{oj} на P_{xj}.

Наконец, последний, *четвертый, вариант абстракции свойств одной вещи* (рис. 5.14, *г*) наиболее близок к предельной трактовке рассматриваемой операции как непосредственного перехода базисных категорий согласно (5.14):

$$L_a^{4}[A_i(P_{i1}, P_{i2}, \ldots, P_{in})] \rightarrow A_k, A_{k+1}, \ldots, A_{k+m-1}, \qquad (5.31)$$

где L_a^{4} — оператор абстрагирования типа 4; n — количество свойств исходного представления вещи A_i, $n \geqslant 1$; $A_k, A_{k+1}, \ldots, A_{k+m-1}$ — вещи, соответст-

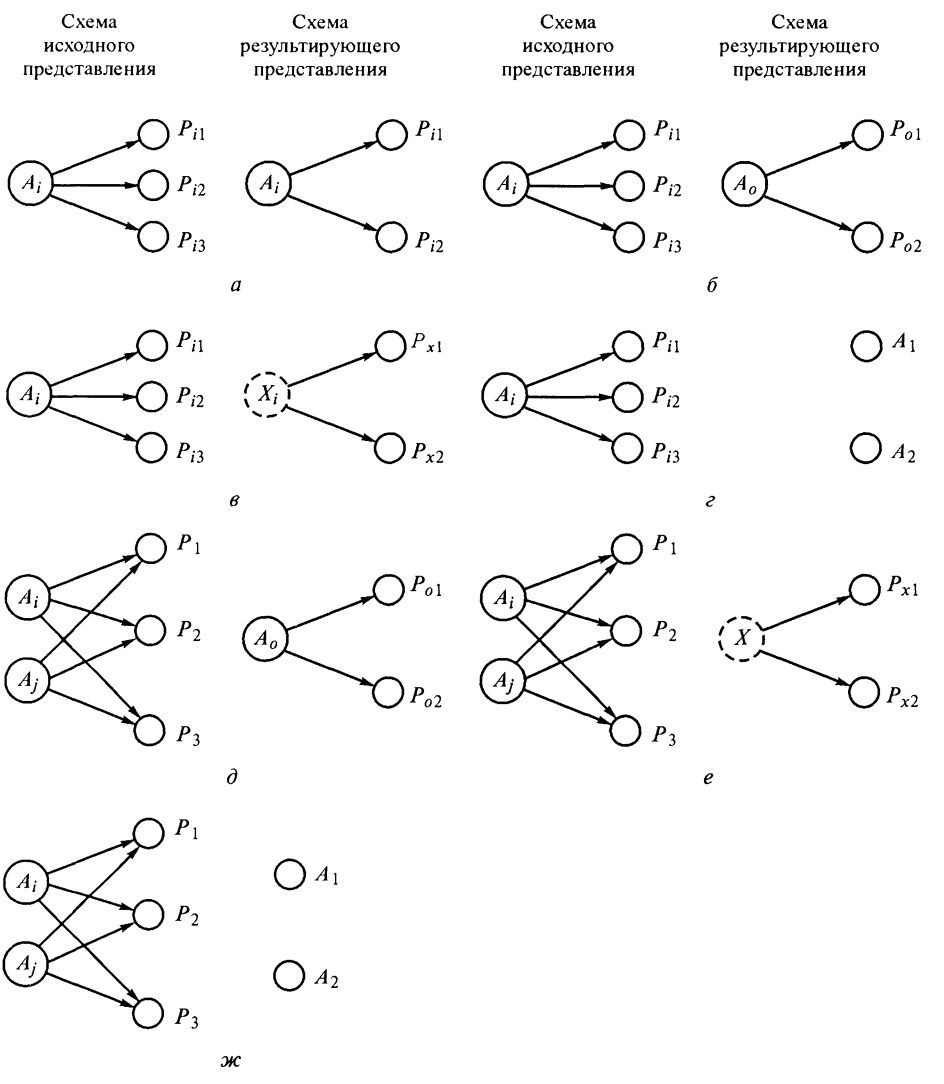


Рис. 5.14. Схемы разновидностей операции абстракции

вующие выделенным свойствам, рассматриваемым вне их вещи-носителя A_i; m — количество выделенных свойств A_i, $1 \leqslant m \leqslant n$.

Типы абстракции 5–7 (рис. 5.14, *д–ж*) представляют собой аналоги типов 2—4 для случаев абстракции класса вещей (формальное выражение абстракции типа 6 может быть получено из (5.32) путем замены абстрактной вещи A_o на переменную вещь X и переобозначения индексов ее свойств с P_{oj} на P_{xj}):

$$L_a^5[A_1(P_{11}, P_{12}, \ldots, P_{1n_1}), A_2(P_{21}, P_{22}, \ldots, P_{2n_2}), \ldots,$$
$$A_n(P_{n1}, P_{n2}, \ldots, P_{nn_n})] \to A_o(P_{o1}, P_{o2}, \ldots, P_{om}); \quad (5.32)$$
$$L_a^7[A_1(P_{11}, P_{12}, \ldots, P_{1n_1}), A_2(P_{21}, P_{22}, \ldots, P_{2n_2}), \ldots,$$
$$A_n(P_{n1}, P_{n2}, \ldots, P_{nn_n})] \to A_k, A_{k+1}, \ldots, A_{k+m-1}. \quad (5.33)$$

Здесь L_a^5, L_a^7 — операторы абстрагирования типов 5 и 7; n — количество элементов в абстрагируемом множестве вещей, $n > 1$; n_i ($i = \overline{1, n}$) — количество свойств представления i-й вещи данного множества, $n_i \geqslant 1$; m — количество свойств абстрактной вещи A_o (5.32) или количество выделенных общих свойств класса вещей, рассматриваемых в отрыве от их вещей-носителей, т. е. как отдельные самостоятельные вещи (5.33), $m \leqslant \min\limits_{i=\overline{1,n}}(n_i)$.

Операцией, обратной абстракции, является ***конкретизация.*** Традиционно конкретизация трактуется как переход от одного или множества свойств к вещи-носителю этих свойств. Конкретным же объектом называют материальный предмет во всем его многообразии признаков, свойств, связей и отношений.

Свойства, оторванные от их вещей-носителей, интерпретируются как отдельные самостоятельные вещи. Согласно данному положению переход от свойств к вещам практически невозможен, так как свойства вне вещей, т. е. сами по себе, не существуют, ибо, в свою очередь, тут же превращаются в вещи. Для преодоления подобной трудности были введены два особых варианта абстракции (3-й и 6-й), суть которых состоит в формировании подразумеваемого объекта-переменной X — носителя выделенных свойств. Ясно, что именно этот виртуальный объект предназначен для использования в качестве недостающего звена-фиксатора свойств-аргументов при традиционном понимании конкретизации. Отметим, что предлагаемый способ выражения операции конкретизации имеет еще одно преимущество: сохраняет отношения, в которых рассматриваются свойства вещи X. Формально схему конкретизации переменной вещи X представим в виде:

$$L_c[X(P_{x1}, P_{x2}, \ldots, P_{xm})] \to A_i, A_{i+1}, \ldots, A_{i+n-1}, \quad (5.34)$$

где L_c — оператор конкретизации; X — подразумеваемая переменная вещь-носитель выделенных свойств $P_{x1}, P_{x2}, \ldots, P_{xm}$; m — количество выделенных свойств; $A_i, A_{i+1}, \ldots, A_{i+n-1}$ — результат конкретизации: n конкретных вещей-носителей выделенных свойств.

Как видим, результатом (5.34) является не одна, а множество (n) вещей-носителей свойств $P_{x1}, P_{x2}, \ldots, P_{xm}$. Заметим, что приведенная трактовка перехода от свойств к вещам позволяет рассматривать схему (5.34) как частный случай поиска аналогии по атрибутивному основанию [218, 220].

Рассмотренные операции абстракции и конкретизации охватывают два аспекта взаимоперехода базисных категорий *M*4: от вещей к свойствам и от свойств к вещам. Механизмы же перехода от вещей к отношениям и наоборот традиционно называют формализацией и интерпретацией.

В самом общем случае ***формализация*** заключается в анализе множества вещей-коррелятов определенной совокупности отношений, отвлечении от несущественных и выделении подмножества значимых в данной ситуации отношений. Кроме того, в ходе формализации возможна замена конкретных вещей-коррелятов отношений подразумеваемыми переменными объектами, т. е. переход к «чистой» структуре, и, в пределе, полный отрыв вычленяемых отношений от носителей и превращение их в отдельные самостоятельные вещи.

Для выражения операций формализации и интерпретации используем обозначения соотносимых вещей, в которых множество отношений будет указываться в скобках между символами самих вещей, например $A_i(R_{i1}, R_{i2}, \ldots, R_{in})A_j$: отношения $R_{i1}, R_{i2}, \ldots, R_{in}$ связывают вещи A_i и A_j.

Первый вариант формализации во многом похож на подобный ему тип абстракции, образно сравниваемый с изучением вещей через лупу или микроскоп:

$$L_f^1[A_i(R_{i1}, R_{i2}, \ldots, R_{iv})A_j] \to A_i(R_{i1}, R_{i2}, \ldots, R_{iu})A_j, \qquad (5.35)$$

где L_f — оператор формализации типа 1; v — количество отношений, связывающих вещи A_i и A_j, $v > 1$; u — количество выделенных отношений, $1 \leqslant u < v$.

Специфика данной разновидности формализации состоит в том, что выделяемые в результате ее отношения установлены по качествам вещей-коррелятов, а среди отбрасываемых отношений нет ни одного существенного, что, естественно, гарантирует сохранение качественной определенности соотносимых объектов. На рис. 5.15, *а* показана схема, иллюстрирующая (5.35) на примере внешних отношений.

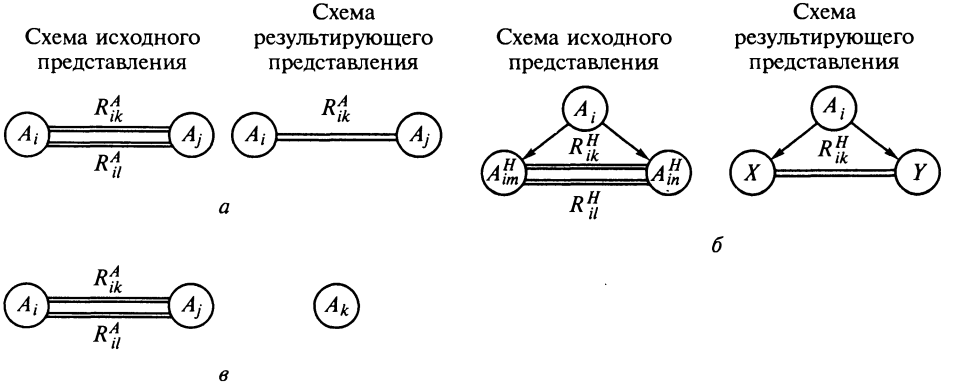


Рис. 5.15. Схемы разновидностей операции формализации

В соответствии со *вторым вариантом формализации* конкретные вещи-корреляты выделяемых отношений заменяются подразумеваемыми объектами X и Y, т. е. осуществляется переход к «чистой структуре»:

$$L_f^2[A_i(R_{i1}, R_{i2}, \ldots, R_{iv})A_j] \to X(R_{x1}, R_{x2}, \ldots, R_{xu})Y. \qquad (5.36)$$

Здесь L_f^2 — оператор формализации типа 2; X и Y — подразумеваемые вещи-переменные.

Схема, иллюстрирующая (5.36) на примере внутренних отношений, изображена на рис. 5.15, *б*.

Наконец, *третья разновидность формализации* характеризуется полным отрывом выделяемых отношений от их вещей-носителей, обусловливающим рассмотрение этих отношений как отдельных самостоятельных вещей:

$$L_f^3[A_i(R_{i1}, R_{i2}, \ldots, R_{iv})A_j] \to A_k, A_{k+1}, \ldots, A_{k+u-1}, \qquad (5.37)$$

где L_f^3 — оператор формализации типа 3; v — количество отношений, связывающих вещи A_i и A_i, $v \geqslant 1$; u — количество выделенных отношений, $1 \leqslant u \leqslant v$.

На рис. 5.15, *в* приведена схема, иллюстрирующая (5.37) на примере внешних отношений.

Для анализа схемы обратной формализации операции ***интерпретации*** в качестве исходного представления удобнее всего использовать результат формализации типа 2 (5.36):

$$L_i[X_i(R_{x1}, R_{x2}, \ldots, R_{xu})Y] \to A_i, A_{i+1}, \ldots, A_{i+2k-1}. \qquad (5.38)$$

Здесь L_i — оператор интерпретации; X, Y — подразумеваемые вещи-переменные, между которыми существуют отношения R_{x1}, R_{x2}, ... , R_{xu}; A_i, A_{i+1}, ... , A_{i+2k-1} — $2k$ вещей, между парами которых (A_i и A_{i+1}, A_{i+2} и A_{i+3}, ... , A_{i+2k-2} и A_{i+2k-1}) существуют отношения R_{x1}, R_{x2}, ... , R_{xu}.

Суть интерпретации заключается в переходе от объектов-переменных X и Y к конкретным вещам-коррелятам некоторой заранее фиксированной совокупности отношений R_{x1}, R_{x2}, ... , R_{xu}, причем очевидно, что условию (5.38) может удовлетворять не одна пара, а множество вещей. В этом плане интерпретацию правомерно трактовать как частный случай поиска аналогии по реляционному основанию [218, 220].

Несмотря на кажущуюся общность абстракции и формализации, между данными операциями существуют достаточно важные различия, одним из которых является то, что формализация никогда не исходит из анализа класса вещей. Разумеется, сделанное замечание отнюдь не отрицает весьма распространенной на практике ситуации, когда изучаются тождественные отношения, образующие не совпадающие друг с другом пары вещей, ибо каждую из подобных пар можно трактовать как одну вещь-систему, форми-

руемую связывающим корреляты отношением и характеризующуюся фактом наличия этого отношения. Таким образом, тождественные отношения между не совпадающими вещами представляют собой не что иное, как тождественные свойства рассматриваемых целостно пар вещей, т. е. фактически имеет место абстракция, а не формализация.

Таким образом, формализация не допускает одновременного анализа совокупности вещей-носителей отношений, а требует отдельного исследования каждой системы объектов, т. е. является более сложным процессом, чем абстракция. Вместе с тем, описываемые операции не следует противопоставлять друг другу, ибо на практике они, как правило, используются последовательно. Кроме того, в технологическом плане можно предложить и вариант совместного или параллельного действия абстракции и формализации, например, отвлечение от свойств и отношений и выделение свойств назвать обобщенной абстракцией, а отвлечение от свойств и отношений и выделение отношений — обобщенной формализацией.

5.5.4. Комплексные операции

Верификация знаний

Реализация эффективных методов и алгоритмов ***верификации знаний*** является одной из ключевых проблем, связанных с развитием технологий БЗ. Необходимость верификации БЗ обусловлена тем, что ее содержание формируется за счет интеграции сведений из разнородных источников, отличающихся различными степенями достоверности, полноты и точности.

Традиционно верификация включает контроль синтаксиса представления информации на входе в ИАС и проверку выполнения фиксированного множества ограничений целостности. Возможности подобных средств верификации являются недостаточными. Они не позволяют идентифицировать неявные семантические ошибки, оценивать корректность поступившей информации в контексте текущего состояния БЗ, переоценивать и модифицировать имеющиеся сведения, согласуя их с новыми знаниями. Сказанное подчеркивает актуальность развития методов интеллектуальной верификации знаний, обеспечивающих контроль состояния БЗ не только на синтаксическом, но и на семантическом уровне [214].

Методы интеллектуальной верификации, предусмотренные в модели *М*4, подразделяют на четыре класса [215]:

1) методы проверки выполнения базовых (независимых) ограничений целостности;

2) методы анализа структурной семантики БЗ;

3) методы анализа семантических зависимостей в БЗ;

4) методы разрешения противоречий.

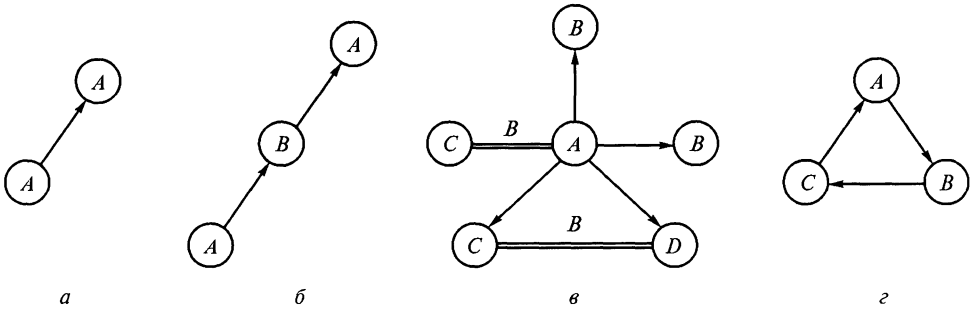


Рис. 5.16. «Сомнительные» семантические структуры:
а — рефлексия; *б* — симметрия; *в* — «персечение категорий»; *г* — «круг»

Методы первого класса обеспечивают проверку соответствия описания вещи структуре ее определенности, принятой в *М*4. Элементы определенности (нечеткие свойства и отношения) контролируются по отдельности, независимо друг от друга. Реализация подобных методов выполняется традиционными средствами и не требует пояснений.

Методы анализа структурной семантики БЗ позволяют выявлять и оценивать допустимость «сомнительных» семантических структур в БЗ. К таким структурам относятся:

- рефлексия, т. е. описание вещи через соответствующий ей объект (рис. 5.16, *а*);
- симметрия, т. е. описание вещи *A* через объект *B*, а вещи, соответствующей *B*, через объект *A* (рис. 5.16, *б*);
- «пересечение категорий», т. е. наличие в описании вещи нескольких ссылок на один и тот же объект в разных формах (как свойство, как отношение, как часть) — рис. 5.16, *в*;
- «круг», т. е. описание вещи *A* через объект *B*, вещи, соответствующей *B*, через объект *C*, а вещи, соответствующей *C*, через объект *A* (рис. 5.16, *г*).

Методы анализа структурной семантики для модели *М*4 подробно описаны в [220].

Методы анализа семантических зависимостей в БЗ основываются на рассмотрении полного множества возможных сочетаний элементов представлений пары свойств или отношений, принципе взаимообоснования свойств и отношений, а также выделении множества семантических признаков противоречивости. Анализ включает три этапа:

1) выявление безусловно противоречивых и безусловно допустимых сочетаний;

2) уточнение меры совпадения элементов обоснований свойств или отношений в оставшихся сочетаниях и преобразование их к комбинациям безусловно противоречивых и допустимых сочетаний;

3) выбор сочетаний, позволяющих провести их модификацию для преодоления обнаруженных несогласованностей.

Подобный подход выходит за рамки «чистой» идентификации противоречивости и также затрагивает вопросы разрешения противоречий.

Отметим, что множество семантических признаков противоречивости может быть динамическим. Часть таких признаков задается формально априори, другие могут определяться в процессе самообучения средств верификации.

Исходным материалом для верификации служит пара членов одного из нечетких множеств свойств или отношений вещи X и Y. В табл. 5.5 приведены 32 возможных варианта соотношений между X и Y, деление на которые базируется на следующих пяти факторах:

- совпадение X и Y, имеющее место при совпадении ссылок X и Y на задающие их объекты и, если X, Y – отношения, при совпадении ссылок на объекты, задающие корреляты данных отношений (столбец 2);
- тождество ссылок обоснований $B'(X)$ и $B'(Y)$, имеющее место при совпадении объектов, соответствующих родовым понятиям X и Y (столбец 3);
- совпадение временных меток $T(X)$ и $T(Y)$ (столбец 4);
- совпадение индексов распределенности, имеющее место при одновременном тождестве значений индексов в обоих парах (экстенсиональной и интенсиональной), т. е. I—I, II—II, III—III, IV—IV (столбец 5);
- обозначенное «1» совпадение качеств распределенности $q(X)$ и $q(Y)$ (++, – –) и обозначенное «0» несовпадение $q(X)$ и $q(Y)$ (+–, –+) (столбец 6).

Разделим все множество вариантов табл. 5.5 на два больших класса. К первому отнесем случаи тождества анализируемых элементов описания (варианты 1—16), т. е. когда $X = Y$ (например, «Небо утром голубое»/«Небо днем голубое», «Иван говорит, что небо голубое»/«Петр говорит, что небо не голубое», «Небо всегда голубое»/«Небо голубое только в ясную погоду днем»). Ко второму — сочетания различных свойств или отношений (варианты 17—32), т. е. когда $X \neq Y$ («Небо утром было голубым»/«Небо днем стало серым», «Иван говорит, что небо голубое»/«Петр говорит, что небо не серое», «Небо иногда бывает голубым»/«Небо иногда бывает серым»). Кроме того, в каждом из образованных вариантами 1—16 и 17—32 классов вычленим по два подкласса, соответствующих совпадению и несовпадению обосновывающих X и Y ссылок на родовые понятия Z_x и Z_y: $Z_x = Z_y$ (варианты 9—16, 25—32) и $Z_x \neq Z_y$ (варианты 1—8, 17—24). Рассмотрим выделенные подклассы вариантов из табл. 5.5.

Варианты 1—8 и 17—24. Описание вещи парой свойств или отношений, обоснованных нетождественными ссылками на родовые понятия, представляет собой разностороннюю, разноплановую характеристику данной вещи, что совершенно естественно и широко распространено на практике.

Таблица 5.5

Вариант	Совпадение X и Y	Тождество ссылок $B'(X)$ и $B'(Y)$	Совпадение $T(X)$ и $T(Y)$	Совпадение индексов распределенности	Соотношение $q(X)$ и $q(Y)$
1–8	+	–	*	*	*
9	+	+	–	+	1
10	+	+	–	+	0
11	+	+	–	–	1
12	+	+	–	–	0
13	+	+	+	–	1
14	+	+	+	–	0
15	+	+	+	+	1
16	+	+	+	+	0
17–24	–	–	*	*	*
25	–	+	–	+	1
26	–	+	–	+	0
27	–	+	–	–	1
28	–	+	–	–	0
29	–	+	+	–	1
30	–	+	+	–	0
31	–	+	+	+	1
32	–	+	+	+	0

Примечание. «+» — совпадение (тождество); «–» — несовпадение (различие); «1» — совпадающие качества распределенности; «0» — противоположные качества распределенности; «*» — соотношение не играет роли.

Отметим специфику вариантов 1—8, состоящую в том, что исходя из совокупности одинаковых свойств или отношений какой-либо вещи, зафиксированных различными исследователями в разное время, можно получить заключение, обобщающее подобную совокупность, т. е. осуществить вывод через *операцию абстракции обоснований.* Используя в качестве основы дефиниции свойства или отношения X определение (5.16), выразим схему абстракции обоснований в следующем виде:

$$L_b[(I(X_1), B(X_1)), (I(X_2), B(X_2)), \ldots, (I(X_n), B(X_n))] \rightarrow (I(X_0), \tilde{B}(X_0)), \quad (5.39)$$

где $X_1 = X_2 = \ldots = X_n = X_0 - n+1$ тождественных свойств или отношений; L_b — оператор абстрагирования обоснований; $I(X_i)$ — ссылка на объект, соответствующий свойству или отношению X_i; $B(X_1)$, $B(X_2)$, ... , $B(X_n)$ — не совпадающие друг с другом обоснования свойств или отношений X_1, X_2, ... , X_n; $\tilde{B}(X_0)$ — результирующее обоснование.

Варианты 9—16. Вещь A описывается парой свойств или отношений X и Y, рассматриваемых в одном отношении или установленных по одним и тем же свойствам, т. е. имеют место формальные признаки противоречия. Вместе с тем, очевидно, что безусловно противоречивы только представления в вариантах 15 и 16, так как при всех прочих сочетаниях обоснований X и Y в них есть попарно не тождественные составляющие. Кроме того, поскольку $X = Y$, содержательная интерпретация противоречий ограничена двумя фиксированными типами: во-первых, констатацией одновременного наличия или отсутствия того или иного свойства или отношения («Небо голубое»/«Небо не голубое») и, во-вторых, указанием различной степени существенности (информативности) свойства или отношения («Небо всегда голубое»/«Небо иногда голубое»). Противоречия первого типа назовем сильными (вариант 16), второго типа — слабыми (вариант 15).

Итак, варианты 15 и 16 безусловно соответствуют противоречию X и Y, а варианты 9—14 могут быть как противоречивыми, так и непротиворечивыми, т. е. выявление их характера требует более детального анализа. Ясно, что суть подобного анализа должна заключаться в уточнении меры близости временных меток и индексов предикации X и Y, выделении их «пересекающихся» частей и преобразовании соотношения X и Y к комбинации непротиворечивых и безусловно противоречивых (варианты 15, 16) элементов описания.

Так, для исключения конфликта (вариант 9) X и Y, значения индексов распределенности которых совпадают ($x_E(X) = x_E(Y)$ и $x_I(X) = x_I(Y)$), достаточно объединить несовпадающие временные метки $T(X)$ и $T(Y)$, например, с помощью следующих видов операции абстракции обоснований (5.39):

- «поглощение» одной временной меткой ($T(X)$) другой метки ($T(Y)$)

$$L_b^T[(I(X), T(X)), (I(Y), T(Y))] \to (I(X), T(X)); \qquad (5.40)$$

- «склейка» временных меток

$$L_b^T[(I(X), T(X)), (I(Y), T(Y))] \to (I(X), \tilde{T}(X)); \qquad (5.41)$$

где $X = Y$, $I(X) = I(Y)$; L_b^T — оператор абстрагирования временных меток обоснований; $\tilde{T}(X)$ — результирующая временная метка.

Как видим, целью подобных операций является переход от противоречивой пары X–Y к одному свойству или отношению X, обоснование которого включает в себя временной аспект тождественного ему элемента описания.

В вариантах 9—14 X и Y противоречат друг другу, если их временные метки «пересекаются» («+» в табл. 5.6), а значения в обеих парах индексов распределенности удовлетворяют условиям, приведенным в табл. 5.7, 5.8 («+»).

Таблица 5.6

$T(Y)$	$T(X)$						
	p	i	f	pi	pf	if	u
p	+	–	–	+	+	–	+
i	–	+	–	+	–	+	+
f	–	–	+	–	+	+	+
pi	+	+	–	+	+	+	+
pf	+	–	+	+	+	+	+
if	–	+	+	+	+	+	+
u	+	+	+	+	+	+	+

Примечание. Символы p, i и f обозначают прошедшее, настоящее и будущее время.

Таблица 5.7

$x_I(Y)$	$x_I(X)$			
	I	II	III	IV
I	+	+	+	+
II	+	+	+	+
III	+	+	–	+
IV	+	+	+	+

Таблица 5.8

$x_E(Y)$	$x_E(X)$			
	I	II	III	IV
I	+	+	+	–
II	+	+	+	–
III	+	+	–	–
IV	–	–	–	–

В ряде случаев сформулированный критерий нетрудно уточнить путем конкретизации обусловливающих существование конфликта составляющих обоснований X и Y. Так, например, анализируя пару суждений «Утром небо было голубое» и «Утром и днем небо не было голубым», можно выявить сильное противоречие между оценками неба утром. Что же касается голубизны дневного неба, то этот факт первым суждением не фальсифицируется, и, следовательно, его правомерно выделить и выразить в форме третьего, не конфликтующего с другими характеристиками неба, свойства («Днем небо не было голубым»).

Итог анализа вариантов 9—16 в виде схемы классификации типов сочетаний X и Y показан на рис. 5.17. В целом же сильные и слабые противо-

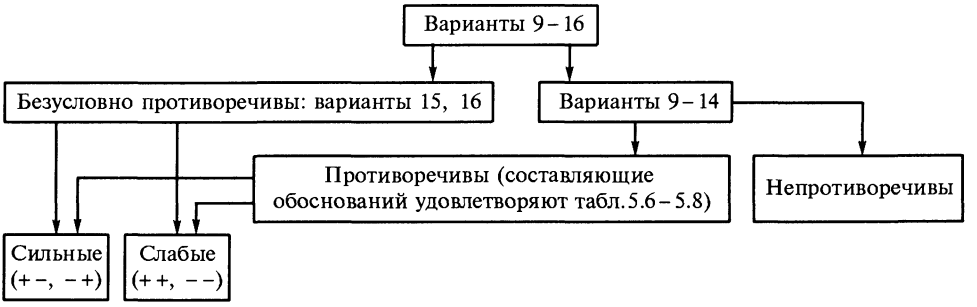


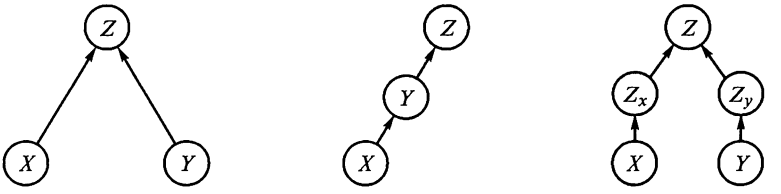
Рис. 5.17. Типы сочетаний X и Y в вариантах 9—16

речия вариантов 9–16 ($X = Y$) далее условно будем называть противоречиями количества или оценки, так как фактически между собой конфликтуют не сами тождественные X и Y, а оценки их одновременного наличия или отсутствия (слабые противоречия) либо степени принадлежности и непринадлежности какой-нибудь вещи (сильные противоречия).

Варианты 25—32. Единственное отличие вариантов 25—32 от вариантов 9—16 заключается в несовпадении элементов определенности вещи $X \neq Y$. Однако, невзирая на это обстоятельство, варианты 25—32 также обладают формальными признаками противоречия, отражаемыми соответствующим законом (более одного свойства или отношения, рассматриваемых в одном отношении или установленных по одним и тем же свойствам). Близость вариантов 25—32 и 9—16 во многом обусловливает то, что сформулированный выше для вариантов 9—14 критерий выявления противоречий между X и Y справедлив и применительно к вариантам 25—30, а в двух последних вариантах (31, 32) X и Y безусловно противоречат друг другу.

Среди вариантов 25—32 также можно выделить классы сильных и слабых противоречий. К сильным противоречиям относятся ситуации совпадения качеств распределенности $q(X)$ и $q(Y)$ (++, – –: варианты 25, 27, 29, 31), к слабым — сочетания противоположных $q(X)$ и $q(Y)$ (+–, –+: варианты 26, 28, 30, 32).

При использовании способа выражения обоснований с помощью ссылок на родовые понятия X и Y в практическом плане целесообразно дифференцировать три вида схем выявления «общего рода»: «пересечение на первом уровне» (рис. 5.18, *а*), «общую ветвь» (рис. 5.18, *б*) и «пересечение на втором и последующих уровнях» (рис. 5.18, *в*) (для сокращения объема иллюстраций на рис. 5.18 не показана описываемая вещь, а связи между вершинами обозначают фиксируемые посредством составных свойств ссылки на родовые понятия). В результате анализа обоснований $B'(X)$ и $B'(Y)$ по схемам рис. 5.18 выделяются четыре класса ситуаций (табл. 5.9).



а *б* *в*

Рис. 5.18. Схемы выявления «общего рода»:

а — «пересечение на первом уровне»; *б* — «общая ветвь»; *в* — «пересечение на втором и последующих уровнях»

Таблица 5.9

Название схемы определения тождества $B'(X) = B'(Y)$	Соотношение качеств распределенности $q(X)$ и $q(Y)$		
	++	– –	+–/–+
Пересечение на первом уровне (см. рис. 5.18, *а*)	П	УП	И/И
Общая ветвь (см. рис. 5.18, *б*)	И	И	П/
Пересечение на втором и последующих уровнях (см. рис. 5.18, *в*)	П	УП	И/И

Примечание. «П» — противоречие; «И» — избыточность; «УП» — условное противоречие.

1. Противоречие X и Y, образуемое парой исключающих друг друга свойств или отношений, рассматриваемых в одном и том же отношении или установленных по одним и тем же свойствам. Например, «Небо голубое»/«Небо серое», «Железо — металл»/«Железо — не химический элемент».

2. Избыточность описания вещи, возникающая при наличии в нем элементов, принадлежность которых может быть выведена логическим путем на основании знаний о принадлежности прочих элементов определенности данной вещи. Например, «Небо голубое»/«Небо цветное (обладает цветом)», «Железо — металл»/«Железо — химический элемент», «Серебро — проводник»/«Серебро не является диэлектриком», «Сера не является металлом»/«Сера не является щелочным металлом».

3. Условное противоречие X и Y, способное в зависимости от ряда обстоятельств (существование отличного от X и Y (Z_x и Z_y) вида понятия Z) реализоваться как обычное противоречие или допустимый вариант представления. Например, «Изучаемое вещество — не проводник»/«Изучаемое вещество — не диэлектрик» (если имеется информация о том, что помимо проводников и диэлектриков существует третья разновидность веществ — полупроводники, имеем допустимый вариант, иначе — противоречие).

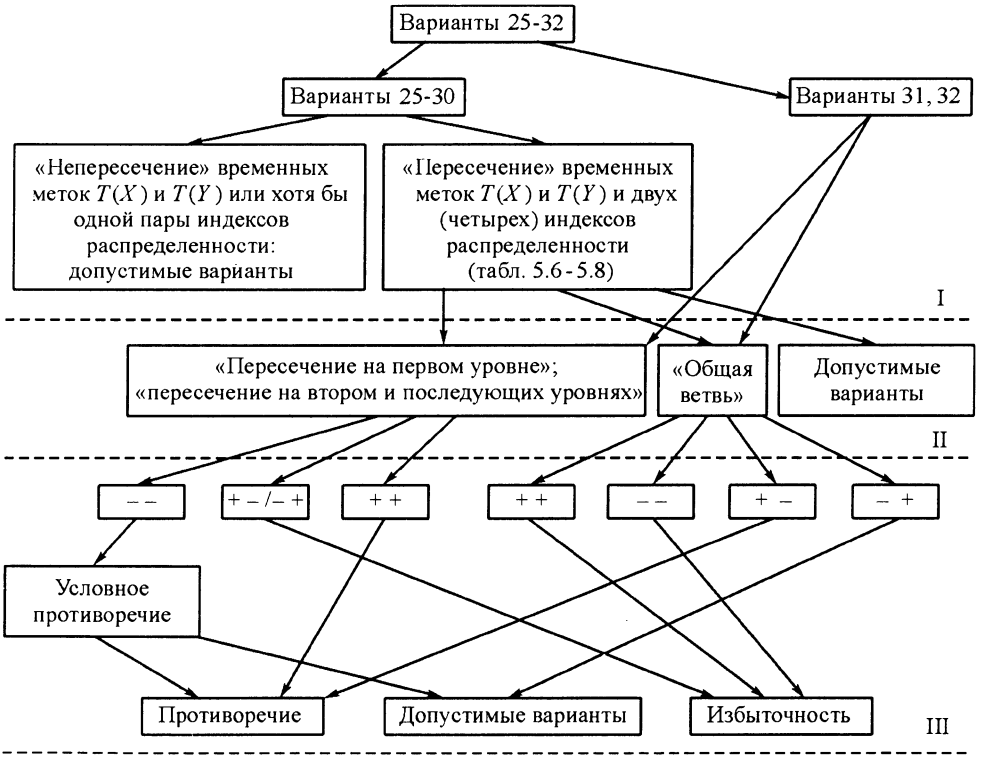


Рис. 5.19. Схема анализа вариантов 25—32

4. Допустимые структуры описания.

Последовательность этапов верификации вариантов 25—32 иллюстрирует рис. 5.19. Разделение «сомнительных» и допустимых соотношений X и Y проводится на трех ступенях анализа (обозначенных римскими цифрами): определении «пересечения» временных меток и индексов распределенности (для вариантов 25—30), выявлении общности ссылок на родовые понятия согласно схемам рис. 5.18 и интерпретации характеризующих противоречивость и избыточность условий.

Разрешение противоречий в базе знаний

В рамках модели $M4$ противоречие соотносится с описанием вещи и имеет место при наличии в нем семантически несовместимых элементов. Традиционно рассматриваются *бинарные противоречия,* обусловленные несовместимостью пары свойств или отношений. Схема, иллюстрирующая классификацию бинарных противоречий элементов описания X и Y, показана на рис. 5.20. В классификации используются четыре основания.

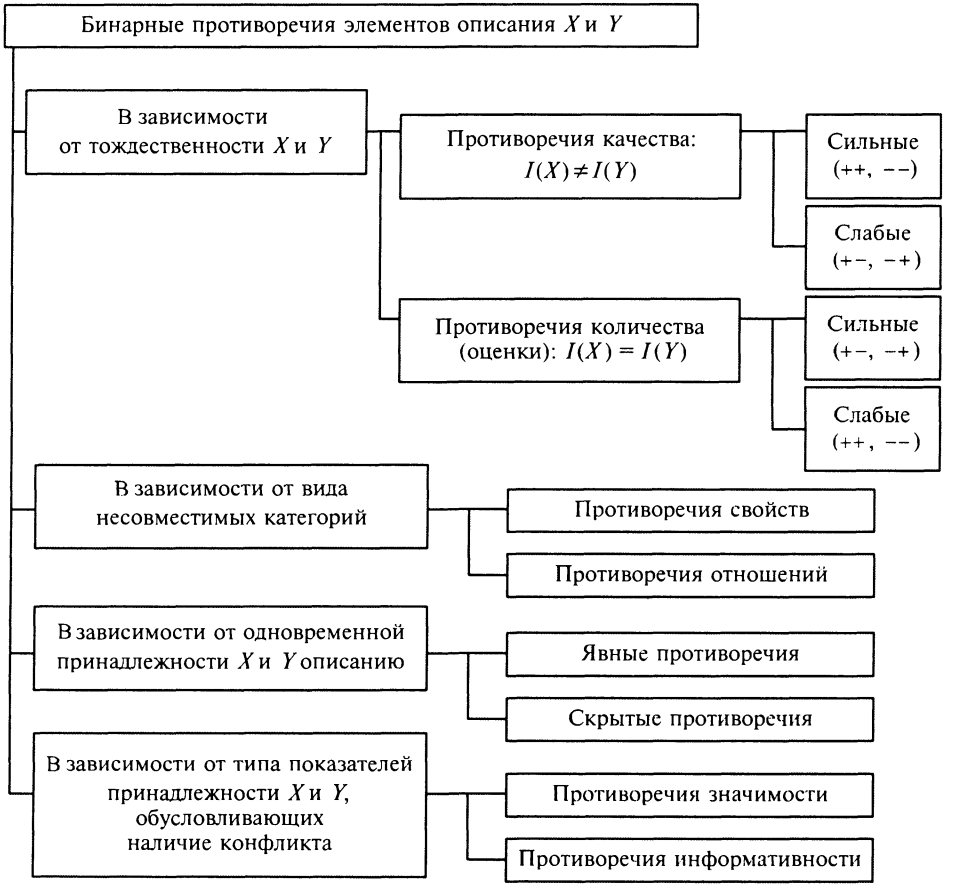


Рис. 5.20. Классификация бинарных противоречий элементов описания вещи в модели *M*4

Во-первых, в зависимости от тождественности конфликтующих элементов выделяются *противоречия качества* (X и Y не тождественны) и *противоречия количества или оценки* (X и Y тождественны). В данном случае имеется в виду качественная тождественность, определяемая совпадением первых членов пары (5.16). Так, X и Y тождественны в качественном отношении, если $I(X) = I(Y)$, т. е. X и Y ссылаются на один и тот же объект. Сочетания качеств распределенности $q(X)$ и $q(Y)$ в каждом из указанных случаев порождают еще пару подклассов: *сильные противоречия качества* ($q(X) = q(Y)$), *слабые противоречия качества* ($q(X) \neq q(Y)$), *сильные противоречия количества* ($q(X) \neq q(Y)$) и *слабые противоречия количества* ($q(X) = q(Y)$). Оригинальность данного аспекта классификации связана с ис-

пользованием общих формальных признаков противоречивости и избыточности. Последняя в рамках классификации ассоциируется с подклассами слабых противоречий при совпадении временных меток, индексов распределенности и интегральных показателей устойчивости X и Y.

Во-вторых, в зависимости от вида несовместимых категорий выделяются *противоречия свойств* (базовых и составных) и *противоречия отношений* (внутренних и внешних).

В-третьих, одновременная принадлежность описанию вещи пары взаимоисключающих друг друга свойств или отношений X и Y представляет собой *явное противоречие.* На практике чаще встречаются *скрытые или потенциальные противоречия,* состоящие в том, что конфликтующие свойства или отношения характеризуют разные вещи, между которыми существуют семантические связи, обусловливающие возможность одновременного наличия несовместимых элементов в одном или обоих описаниях. Например, конфликтующие элементы приписаны вещам, одна из которых соответствует родовому понятию другой. С помощью операций преобразования описаний скрытые противоречия могут быть превращены в явные.

В-четвертых, в зависимости от типа показателей принадлежности X и Y, обусловливающих наличие конфликта, выделяются *противоречия значимости* и *противоречия информативности.*

Исходной информацией для *разрешения противоречия* является пара элементов описания вещи, семантическая несовместимость которых была определена с помощью методов анализа семантических зависимостей в БЗ. Цель разрешения противоречия состоит в устранении семантической несовместимости за счет преобразования описания.

Традиционно пара противоречащих друг другу элементов описания рассматривается как симметричная структура. Это не позволяет учитывать предысторию конфликта, сопутствующие его возникновению обстоятельства. В большинстве случаев причина конфликта свойств или отношений состоит в выдвижении гипотезы о принадлежности изучаемой вещи какого-либо нового элемента описания, фальсифицирующего или опровергающего некоторые ранее известные ее характеристики.

Рассмотрим подход к разрешению бинарных противоречий, основывающийся на *несимметричной интерпретации конфликтующих элементов описания* как старого (известного) и нового знаний [213, 215, 220]. Возможность учета при разрешении противоречия его предыстории, обеспечиваемая подобной интерпретацией, имеет важное значение при верификации динамических БЗ.

Пусть X будет обозначать старый, а Y – противоречащий ему новый элемент описания. Обобщение возможных исходов разрешения конфликта X и Y позволяет выделить два класса стратегий: разрешение на фиксирован-

ном уровне и разрешение с выходом на другие уровни. Уровень в данном случае ассоциируется с границей используемого при анализе фрагмента модели представления ПрО. В стратегиях первого класса этой границей является описание вещи-носителя X и Y (рис. 5.21). При разрешении противоречия ей не приписывается никаких свойств и отношений, заменяющих X и Y и способных снять или «смягчить» исходный конфликт, т. е. единственный путь его преодоления — изменение обоснований или исключение самих X и Y. В стратегиях второго класса это ограничение снимается, т. е. анализ выходит за рамки отдельного описания. Таким образом, от выбора уровня (класса стратегий) зависит объем знаний, привлекаемых при разрешении противоречий, что, в свою очередь, влияет на глубину анализа.

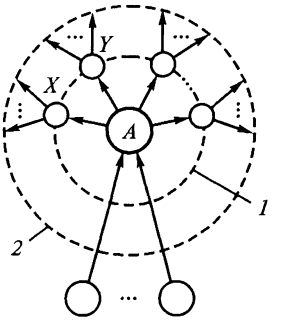


Рис. 5.21. Границы фрагментов модели ПрО, используемых в стратегиях разрешения противоречий на фиксированном уровне и с выходом на другие уровни:

A — вещь-носитель несовместимых элементов описания X и Y; *1*, *2* — границы фрагментов модели ПрО, используемых при разрешении противоречий на фиксированном уровне (*1*) и с выходом на другие уровни (*2*)

В классе *стратегий разрешения противоречий на фиксированном уровне* на основе аналогии с принятием решений человеком в конфликтных ситуациях выделены четыре базовые схемы:

1) «консерватизм и недоверие»;
2) «частичная фальсификация и прагматизм»;
3) «наивная переоценка и вера»;
4) «полная фальсификация».

В рамках *первой стратегии* абсолютный приоритет имеет старый элемент описания X, что обеспечивает запрет на доминирование X новой характеристикой Y. В результате сколько бы ни было конфликтов новых элементов со старым, последний никогда не будет подвергнут ревизии. При использовании этой стратегии текущее представление ПрО обладает максимальной устойчивостью.

Рассматриваемая стратегия включает две разновидности. Первая формально представляется выражением

$$L_D^{10}[(I(X), B(X)), (I(Y), B(Y))] \rightarrow (I(X), B(X)), \qquad (5.42)$$

где L_D^{10} — оператор разрешения противоречий на фиксированном уровне типа 10 (смысл верхнего индекса пояснен ниже).

Обозначим нулем исключаемый из описания (X) или отвергаемый новый (Y) элемент, единицей — элемент, остающийся в описании (X) или

включаемый в него (Y) с сохраняющимся неизменным обоснованием $B(X)$ или $B(Y)$, а двойкой — элемент, обоснование которого модифицируется. Тогда рассматриваемые стратегии можно представить кодами ij ($i, j \in \{0, 1, 2\}$), в которых первая позиция соответствует X, а вторая — Y. Подобный код указан в качестве верхнего индекса оператора разрешения противоречий в (5.42).

Выражение (5.42) соответствует варианту, при котором все факты фальсификации X система забывает, т. е. фальсификация не приводит к изменению его обоснования.

Вторая разновидность стратегии отличается тем, что несмотря на разрешение противоречия в пользу старой характеристики факт ее фальсификации фиксируется в модели, проявляясь в модификации обоснования X. В содержательном плане подобная модификация отражает снижение уверенности в принадлежности X. Соответствующее преобразование имеет следующий вид:

$$L_D^{20}[(I(X), B(X)), (I(Y), B(Y))] \to (I(X), \tilde{B}(X)). \qquad (5.43)$$

Здесь L_D^{20} — оператор разрешения противоречий на фиксированном уровне типа 20; $\tilde{B}(X)$ — модифицированное обоснование элемента описания X.

Особенность *стратегии «частичной фальсификации и прагматизма»* состоит в том, что оба изначально конфликтующих элемента остаются в описании. Разрешение противоречия достигается за счет модификации компонентов их обоснований. Данная стратегия является наиболее часто применяемой и сложной в реализации. Модификация может коснуться либо одного из обоснований $B(X)$ или $B(Y)$, либо обоих. Таким образом, выделяются три разновидности стратегии:

- «прагматичный консерватизм» ($B(X)$ не меняется, $B(Y)$ модифицируется);
- «прагматичное доверие» ($B(X)$ модифицируется, $B(Y)$ не меняется);
- «прагматичный баланс» (модифицируются и $B(X)$, и $B(Y)$).

Соответствующие схемы формально описывают выражения:

$$L_D^{12}[(I(X), B(X)), (I(Y), B(Y))] \to \{(I(X), B(X)), (I(Y), \tilde{B}(Y))\}, \qquad (5.44)$$

$$L_D^{21}[(I(X), B(X)), (I(Y), B(Y))] \to \{(I(X), \tilde{B}(X)), (I(Y), B(Y))\}, \qquad (5.45)$$

$$L_D^{22}[(I(X), B(X)), (I(Y), B(Y))] \to \{(I(X), \tilde{B}(X)), (I(Y), \tilde{B}(Y))\}, \qquad (5.46)$$

где L_D^{12}, L_D^{21}, L_D^{22} — операторы разрешения противоречий на фиксированном уровне типов 12, 21 и 22; $\tilde{B}(Y)$ — модифицированное обоснование элемента описания Y.

Третья стратегия представляет собой инверсию первой. Здесь абсолютный приоритет отдается новому элементу Y. Представление ПрО при

использовании этой стратегии является максимально подвижным. Каждое новое знание вызывает его изменение.

Данная стратегия имеет две разновидности. При первой обоснование включаемого в описание элемента Y не изменяется, при второй – модифицируется с учетом обоснования исключаемого элемента X. Содержательно такая модификация отражает влияние старого знания на новое, обеспечивая преемственность представлений о ПрО. Соответствующие преобразования формально выражаются в следующем виде:

$$L_D^{01}[(I(X), B(X)), (I(Y), B(Y))] \to (I(Y), B(Y)), \tag{5.47}$$

$$L_D^{02}[(I(X), B(X)), (I(Y), B(Y))] \to (I(Y), \tilde{B}(Y)). \tag{5.48}$$

Здесь L_D^{01}, L_D^{02} — операторы разрешения противоречий на фиксированном уровне типов 01 и 02.

В рамках *четвертой стратегии* из описания исключаются как фальсифицируемый, так и фальсифицирующий элементы. Формально эта стратегия описывается выражением

$$L_D^{00}[(I(X), B(X)), (I(Y), B(Y))] \to \varnothing, \tag{5.49}$$

где L_D^{00} — оператор разрешения противоречий на фиксированном уровне типа 00.

Способами модификации обоснований являются *разрешение противоречий во времени и в пространстве*. Эти методы базируются на предположении о том, что изначально указаны чересчур широкие значения временных меток и индексов распределенности, и наличие конфликта между X и Y свидетельствует о необходимости корректировки данных компонентов их обоснований. Цель модификации в подобных ситуациях состоит в выборе непересекающихся интервалов временных меток и несовместимых индексов предикации.

Если временные метки определить как множества, членами которых являются признаки, обозначающие фиксированные непересекающиеся временные интервалы (например, прошедшее, настоящее и будущее), то для «разнесения» временных меток $T(X)$ и $T(Y)$ можно использовать операции, аналогичные вычитанию множеств и заключающиеся в отбрасывании от модифицируемой метки части, пересекающейся с меткой конфликтующего элемента:

$$\tilde{T}(X) = T(X) \setminus T(Y),$$

$$\text{если } T(Y) \subset T(X) \vee T(Y) \cap T(X) \neq \varnothing \;\&\; T(Y) \neq T(X); \tag{5.50}$$

$$\tilde{T}(Y) = T(Y) \setminus T(X),$$

$$\text{если } T(X) \subset T(Y) \vee T(X) \cap T(Y) \neq \varnothing \;\&\; T(X) \neq T(Y); \tag{5.51}$$

$$\tilde{T}(X) = T(X) \setminus T(Y),\ \tilde{T}(Y) = T(Y) \setminus T(X),$$
$$\text{если } T(X) \cap T(Y) \neq \varnothing \ \&\ T(X) \neq T(Y), \qquad (5.52)$$

где $\tilde{T}(X)$, $\tilde{T}(Y)$ — модифицированные временные метки элементов описания X и Y.

Модификация (5.50) может применяться в рамках стратегий (5.43) и (5.45), модификация (5.51) — в рамках стратегий (5.44) и (5.48), модификация (5.52) — в рамках стратегии (5.46).

Очевидно, что преобразования типа (5.50)–(5.52) имеют смысл, если их результат — не пустое множество: $\tilde{T}(X) \neq \varnothing$, $\tilde{T}(Y) \neq \varnothing$. Данное требование накладывает ограничение на применимость тех или иных стратегий (разумеется, только в части, связанной с разрешением противоречий во времени). Например, если используется прагматичная стратегия и $T(X) \subseteq T(Y)$, то схема (5.45) неприменима, а допустимыми схемами являются (5.44) и (5.46), причем (5.44) неприменима в случае $T(X) = T(Y)$.

Основой разрешения противоречий в пространстве является «разнесение» X и Y между частями исходной вещи (интенсиональный аспект) или вещами, соответствующими объему понятия об этой вещи (экстенсиональный аспект), в результате чего каждой конкретной вещи приписывается или X, или Y, но не оба элемента одновременно. Подход к реализации операций модификации индексов распределенности предложен в [220].

Модификация обоснования не обязательно сводится к разрешению противоречия только во времени или только в пространстве, хотя формально для преодоления несовместимости X и Y достаточно исключить пересечение хотя бы одной пары соответствующих компонентов обоснований $B(X)$ и $B(Y)$. Методы разрешения противоречий во времени и в пространстве могут применяться совместно, при этом стратегии (5.42)–(5.49) должны использоваться синхронно, т. е. при модификации временных меток и индексов распределенности данной пары элементов описания должна применяться одна и та же стратегия.

Схема, иллюстрирующая классификацию стратегий разрешения противоречий на фиксированном уровне и использующая введенную кодировку, показана на рис. 5.22.

Применяемую стратегию предварительно можно выбрать с помощью способа, основывающегося на отношении доминирования одним членом нечеткого множества другого. При разрешении противоречия доминирующий элемент (нечеткое свойство или нечеткое отношение) обладает большей устойчивостью. В стратегиях 10 и 20 он остается в описании, а доминируемый отвергается. В стратегиях 01 и 02 доминирующий элемент включается в описание, а доминируемый исключается. В стратегиях 12 и 21 обоснование доминирующего элемента не изменяется, а обоснование доминируемого модифицируется.

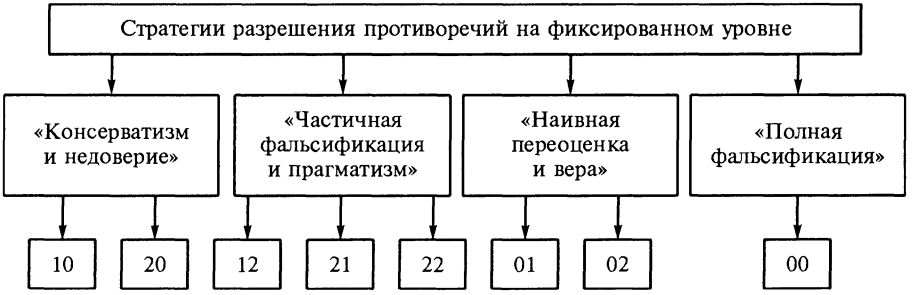


Рис. 5.22. Классификация стратегий разрешения противоречий на фиксированном уровне

Отношение доминирования правомерно задавать по-разному. Простейший вариант — сравнение весов и отдание предпочтения элементу с существенно бóльшим значением. Так, при использовании значимого веса X доминирует Y, если $\mu_M(X) >> \mu_M(Y)$, Y доминирует X, если $\mu_M(Y) >> \mu_M(X)$. При незначительной разнице весов отношение доминирования не устанавливается. В этом случае применяются стратегии 00 и 22. При использовании информативного веса в приведенных неравенствах μ_M меняется на μ_I.

Для одновременного учета нескольких аспектов нечеткости X и Y (в частности, значимости и информативности) могут применяться различные приемы синтеза интегральных оценок: сложение или умножение исходных значений весов, выбор из них максимума или минимума и др. Соответствующие преобразования хорошо освещены в литературе по теории нечетких множеств [219, 221, 222].

Введя показатель, отражающий степень доминирования, данный подход можно использовать для *формирования предпочтений на множестве допустимых стратегий.* Для упрощения изложения будем рассматривать только значимые веса. Определим два варианта показателя доминирования Δ_1 и Δ_2:

$$\Delta_1 = \mu_M(X) - \mu_M(Y); \tag{5.53}$$

$$\left.\begin{aligned}\Delta_2 &= 1 - \mu_M(Y)/\mu_M(X), \text{ если } \mu_M(X) \geqslant \mu_M(Y),\\ \Delta_2 &= \mu_M(X)/\mu_M(Y) - 1, \text{ если } \mu_M(X) < \mu_M(Y).\end{aligned}\right\} \tag{5.54}$$

Поскольку веса принадлежат диапазону]0; 1], то Δ_1, $\Delta_2 \in]-1; 1[$. Можно продемонстрировать, что Δ_2 лучше учитывает семантику разрешения противоречия, однако Δ_1 более адекватен при малых значениях весов, когда выполняется неравенство

$$\max(\mu_M(X), \mu_M(Y)) \leqslant \lambda, \tag{5.55}$$

где $\lambda \in]0; 0{,}3]$ — пороговое значение веса.

Будем исходить из того, что при $\mu_M(X) \geqslant \mu_M(Y)$ применимы стратегии 10, 20, 12, 22 и 00, в противном случае действуют стратегии 01, 02, 21, 22 и 00. Построим шкалу значений показателя доминирования (рис. 5.23) и для каждой стратегии экспертным путем определим функцию предпочтения, отображающую показатель доминирования в единичный интервал:

$$\mu^{ij}: \Delta \to [0; 1], \tag{5.56}$$

где ij — код стратегии; Δ — показатель доминирования (Δ_1 или Δ_2); символ $\to$ обозначает отображение.

Значения $\mu^{ij}(\Delta)$ отражают предпочтительность использования стратегии ij в зависимости от показателя доминирования. С помощью функций (5.56) для вычисленного на основе (5.53) или (5.54) значения Δ множество допустимых стратегий упорядочивается в соответствии с убыванием μ^{ij} ($\Delta = \text{const}$). Стратегии ij и kl, для которых при фиксированном Δ $\mu^{ij}(\Delta) = \mu^{kl}(\Delta)$, считаются одинаково предпочтительными, соответствующее отношение для них не устанавливается, а выбор между ними производится исходя из приоритетов, заданных пользователем БЗ.

Стратегия 00 может действовать только в области малых весов, т. е. при выполнении условия (5.55).

При выборе применяемой стратегии и определении предпочтений на множестве допустимых стратегий наряду с весами конфликтующих свойств или отношений X и Y могут учитываться их взаимосвязи с другими элементами описания рассматриваемой вещи. Последнее имеет особое значение при разрешении слабых противоречий. При анализе избыточных элементов и выборе одного из них ключевую роль, как правило, имеет то, насколько привычно и естественно употребление оставляемого свойства или отношения, как часто оно используется, в какой степени оно сочетается с прочими элементами описания, т. е. один из решающих факторов обусловливается вхождением данной характеристики в множество «приоритетных» определителей.

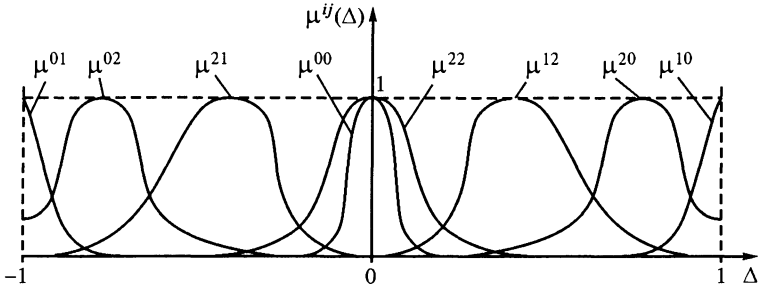


Рис. 5.23. Функции, отражающие предпочтительность использования стратегий разрешения противоречий на фиксированном уровне

Рассмотренные стратегии являются одношаговыми. На практике обновление представлений о ПрО — многоступенчатый процесс. Его типовую схему отражает *итерационный алгоритм совместного применения стратегий разрешения противоречий на фиксированном уровне* (рис. 5.24). Итерационный характер обусловлен тем, что конфликт старого и нового знаний рассматривается как последовательность актов фальсификации элементом Y элемента X. Действие алгоритма инициируется при поступлении информации о принадлежности описанию вещи нового свойства или отношения Y, противоречащего известному элементу X.

Алгоритм состоит из нескольких фаз. На первой сомнения в устойчивости X не допускаются, конфликт однозначно разрешается в его пользу, и новый элемент Y безусловно отвергается (применяется стратегия 10). Последующие фазы выполняются при появлении сомнений в устойчивости X. Если уверенность в его принадлежности превышает пороговое значение λ_1, то все еще превалирует консерватизм. Однако по сравнению с «чистым» исключением X частично фальсифицируется, т. е. незыблемость старой характеристики постепенно подвергается сомнению (используется стратегия 20). Накопление сомнений отражает снижение веса $\mu_M(X)$ (см. рис. 5.24, блок 10).

При выполнении условия $\lambda_2 < \mu_M(X) \leqslant \lambda_1$ применяется стратегия прагматичного консерватизма 12, предусматривающая включение в описание и модификацию обоснования Y. В случае попадания веса $\mu_M(X)$ в интервал $]\lambda_3; \lambda_2]$ действует стратегия 22. Стратегии 20, 12 и 22 могут использоваться неоднократно при обнаружении дополнительных аргументов в пользу Y (блоки 11, 12).

Пороговые значения λ_i, определяющие диапазоны применимости стратегий 20 (накопление сомнений), 12 (отражение в описании нового знания) и 22 (постепенное вытеснение старого знания новым), связаны неравенством: $0 \leqslant \lambda_3 \leqslant \lambda_2 \leqslant \lambda_1 \leqslant 1$ (рис. 5.25). При уменьшении веса ниже порога λ_3 выполняется последняя фаза алгоритма (см. рис. 5.24, блок 9), в рамках которой возможны три варианта. При первом противоречие окончательно разрешается с помощью стратегии 21, 02 или 01. Второй вариант соответствует стратегии 00, используемой при малых значениях весов (условие (5.55)), если до этого не применялись стратегии 12 и 22. Формально противоречие снято, и включению в описание Y в последующем ничто не мешает. Третий вариант представляет собой вырожденный случай стратегии 21, обусловленный отрицанием X (инверсией значения качества распределенности $q(X)$), за счет чего сильное противоречие X и Y преобразуется в слабое.

В основе рассмотренного алгоритма лежит моделирование процессов накопления сомнений и обновления представлений о ПрО, отражаемых в БЗ. Его параметры (пороговые значения λ_i) и процедуры модификации весов позволяют управлять устойчивостью текущих представлений и стилем принятия решений при разрешении противоречий.

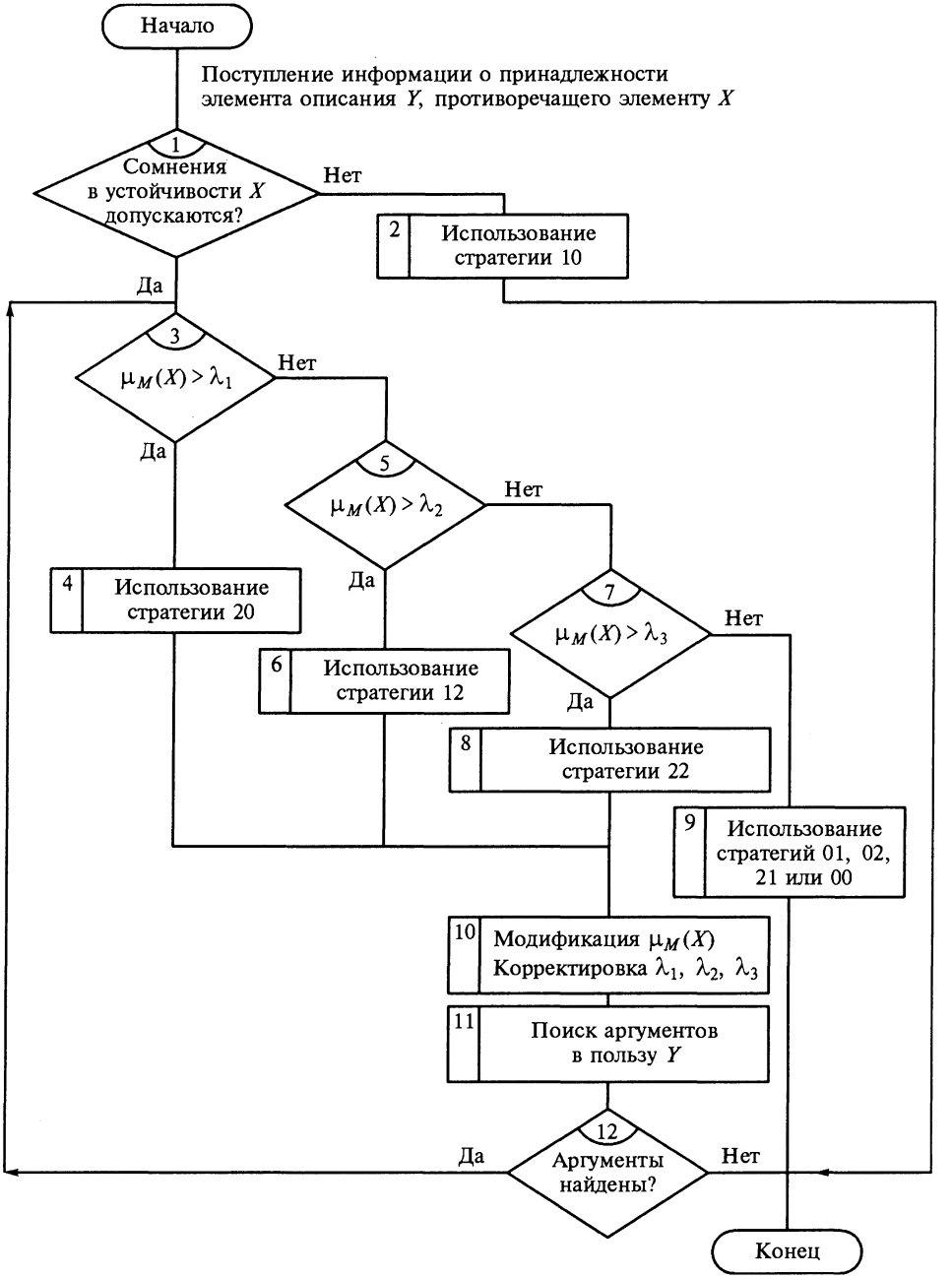


Рис. 5.24. Итерационный алгоритм совместного применения стратегий разрешения противоречий на фиксированном уровне

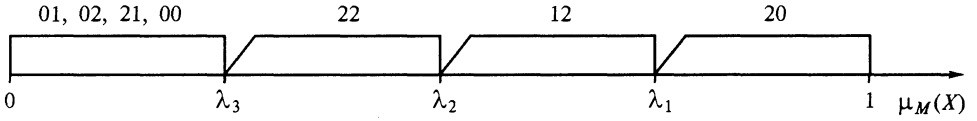


Рис. 5.25. Шкала пороговых значений

Обобщенная схема *стратегий разрешения противоречий с выходом на другие уровни* состоит в замене конфликтующей пары свойств или отношений X и Y альтернативным элементом описания Z. В качестве примера приведем рассуждение: «(небо) голубое, (небо) серое → (небо) цветное», в котором обоснования несовместимых элементов X («голубое») и Y («серое») ссылаются на общее родовое понятие Z («цветное»). Формально подобное преобразование выражается в следующем виде:

$$L_{DR}[(I(X), B(X)), (I(Y), B(Y))] \rightarrow (I(Z), B(Z)), \qquad (5.57)$$

где L_{DR} — оператор разрешения противоречий с выходом на другие уровни; символ → обозначает переход от одного представления вещей к другому.

Преобразования типа (5.57) можно представить через комбинацию элементарных операций над знаниями: включающей Z в описание и порождающей тем самым его избыточность конкретизации или интерпретации и исключающей конфликтующие элементы X и Y абстракции или формализации.

Схемы разрешения противоречий с выходом на другие уровни и на фиксированном уровне могут применяться совместно. Соответствующие преобразования порождаются сочетаниями (5.57) со стратегиями (5.42)—(5.49). В этих случаях отсутствие в результирующем описании изначально несовместимых характеристик X и Y не обязательно, так как разрешение противоречия может быть достигнуто за счет комбинации частичной компенсации их конфликта новым элементом Z (выход на другой уровень) с модификацией обоснований $B(X)$ и $B(Y)$ (фиксированный уровень).

Наследование в базе знаний

Наследованием назовем процесс расширения описания (доопределения) некоторой вещи A_i, базирующийся на знаниях исходных представлений данной вещи и какой-нибудь другой вещи A_j $(i \neq j)$, при котором соответствующие A_i и A_j объекты O_i и O_j являются соседями.

Согласно геометрической интерпретации уровня информационных структур $M4$ (см. рис. 5.6) нетождественные объекты O_i и O_j могут быть соседями только в двух случаях (рис. 5.26):

- соответствующая O_i вещь A_i определяется через свойства или отношения, задаваемые посредством ссылки на объект O_j;

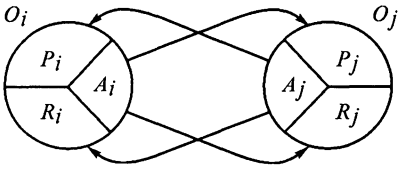


Рис. 5.26. Объекты семантической сети *M*4, являющиеся соседями

• соответствующая O_j вещь A_j определяется через свойства или отношения, задаваемые посредством ссылки на объект O_i.

Таким образом, наследование заключается в приписывании некоторой вещи A_i свойств или отношений, характеризующих вещь A_j, ссылки на соответствующий объект которой O_j выступают в роли элементов определенности A_i. В процессе наследования выделяются следующие основные стадии.

1. Анализ описания вещи A_i и выбор какого-либо ее элемента определенности (свойства или отношения) P_{ij} (R_{ij}).

2. Выполнение операций обобщенной абстракции или формализации, в результате чего производится отвлечение от прочих свойств и отношений A_i, а выбранное свойство или отношение рассматривается в отрыве от его вещи-носителя, т. е. как самостоятельная вещь A_j.

3. Анализ описания вещи A_j и выдвижение гипотезы о принадлежности каких-либо ее элементов определенности вещи A_i.

4. Перенос свойств или отношений с A_j на A_i, т. е. доопределение A_i.

5. Анализ итогового представления вещи A_i:

• выявление безусловно недопустимых ситуаций;

• идентификация и разрешение противоречий.

Подчеркнем, что верификация итоговой совокупности элементов описания A_i (5-я стадия) является неотъемлемой частью наследования, что фактически позволяет не разделять, а объединять достоверные и правдоподобные выводы. В этом плане, поскольку все операции верификации ассоциируются со своими результирующими значениями типа «успех»/«неуспех», причем «успех» верификации свидетельствует о подтверждении первоначальной гипотезы, а «неуспех» — об ее фальсификации, данные значения можно интерпретировать в качестве критериев правдоподобия соответствующих верификации операций наследования. Таким образом, любая операция наследования связывается с некоторой итоговой оценкой, отражающей правомерность сделанных заключений. Понятно, что влияние на представления вещей оказывают только «успешные» попытки наследования, а «неудачные» подлежат отмене, возвращающей измененные описания к их первоначальному виду.

Изложенные рассуждения, касающиеся совмещения наследования и верификации, выражают важный принцип построения моделей умозаключений, благодаря которому исчезает необходимость, «страхуясь» от ошибок, чрезмерно сужать множество используемых операций, ограничивая их чис-

ло только достоверными типами выводов. Действительно, поскольку каждое умозаключение завершается анализом и проверкой результирующих структур определенности, задача отсечения порождающих конфликтные ситуации механизмов решается автоматически на одном из этапов преобразования, т. е. сам процесс приобретает «обратную связь» и становится внутренне управляемым, причем в роли «минимизируемого сигнала ошибки» выступает итоговая оценка наличия или отсутствия противоречий.

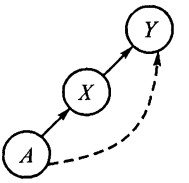


Рис. 5.27. Перенос наследуемого элемента определенности

Возвращаясь к пяти выделенным стадиям наследования, остановимся на аспектах переноса свойств и отношений вещи A_j на вещь A_i (4-я стадия). Пусть имеется некоторая вещь A, ее свойство или отношение X и приписанное соответствующей X вещи свойство или отношение Y (рис. 5.27).

В самом общем виде схема «успешного» наследования без учета возможного разрешения противоречий описывается выражением

$$L_F[\{(I(X), B(X))\}, \{(I(Y(X)), B(Y(X)))\}] \rightarrow \{(I(X), B(X)), (I(Y), B(Y))\}, \quad (5.58)$$

где L_F — оператор наследования.

Фигурные скобки в левой части (5.58) фиксируют характеристики двух различных вещей (A и X). Если же среди аргументов оператора наследования оставить представление только одной, доопределяемой вещи A, модифицированная схема преобразования, заключающегося в приписании данной вещи наследуемого свойства или отношения Y (4-я стадия), фактически будет соответствовать известным операциям конкретизации и интерпретации:

$$L_F[(I(X), B(X))] \rightarrow \{(I(X), B(X)), (I(Y), B(Y))\}. \quad (5.59)$$

Для построения классификации механизмов наследования воспользуемся двумя основаниями деления: во-первых, формой элемента описания X вещи A и, во-вторых, формой элемента описания Y вещи X. Комбинируя между собой существующие формы X и Y, получаем четыре разновидности наследования: наследование свойств по свойствам, свойств по отношениям, отношений по свойствам и отношений по отношениям.

Механизмы определения значений компонентов обоснования $B(Y)$ по значениям компонентов обоснований $B(X)$ и $B(Y(X))$ описаны в [213, 220]. Наиболее просто решается задача анализа временных меток $T(X)$ и $T(Y(X))$ и формирования результирующей метки $T(Y)$. Похожие процедуры описывались выше при обсуждении операций «склейки» и «поглощения» временных меток. Очевидно, что наследование правомерно проводить только тогда, когда значения $T(X)$ и $T(Y(X))$ «пересекаются» (см. табл. 5.6), при этом для

определения итоговой метки $T(Y)$ в $T(X)$ и $T(Y(X))$ достаточно вычленить общую часть (в смысле пересечения множеств):

$$T(Y) = T(X) \cap T(Y(X)). \qquad (5.60)$$

Правила преобразования формы наследуемого элемента Y приведены в табл. 5.10. Как видно из табл. 5.10, во всех вариантах, кроме переноса составных свойств типа P^{AI}, итоговая форма Y сохраняется, т. е. совпадает с формой $Y(X)$.

Таблица 5.10

Категория элемента описания, по которому производится наследование (перенос ... по ...)	Перенос				
	составных свойств типа P^{AI}	составных свойств типа P^{AH}	базовых свойств (P^P)	внешних отношений (R^A)	внутренних отношений (R^H)
по составным свойствам типа P^{AI}	P^{AI}	P^{AH}	P^P	R^A	R^H
по составным свойствам типа P^{AH}	P^{AH}	P^{AH}	P^P	R^A	R^H
по базовым свойствам (P^P)	P^P	–	–	–	–
по внешним отношениям (R^A)	R^A	–	–	–	–
по внутренним отношениям (R^H)	R^H	–	–	–	–

Примечание. «–» — нет наследования.

При наследовании составных свойств типа P^{AI} по базовым свойствам и отношениям часто применяется *комплексное наследование* или *замещение,* состоящее в последовательном выполнении операций наследования и абстракции (или формализации), причем в результате второй операции из представления вещи исключается свойство или отношение X, которое на первом этапе служит «мостиком» для переноса Y, т. е. Y как бы «вытесняет» (замещает) X. Например, «Небо голубое» → «Небо цветное», «Железо горит в кислороде» → «Железо реагирует с кислородом». Сокращенная и развернутая схемы замещения имеют следующий вид:

$$L_{Fa}[(I(X), B(X))] \rightarrow (I(Y), B(Y)); \qquad (5.61)$$

$$L_{Fa}[(I(X), B(X))] = L_a[L_F[(I(X), B(X))]] \rightarrow$$
$$\rightarrow L_a[(I(X), B(X)), (I(Y), B(Y))] \rightarrow (I(Y), B(Y)), \qquad (5.62)$$

где L_{Fa} — оператор замещения; L_F — оператор наследования (см. (5.59)); L_a — оператор абстракции (или формализации – L_f).

Основные выводы

1. И в теории, и в практике ИИ построение и использование БЗ является проблемой такой же важности, как в свое время в информационных технологиях проблема создания БД.

2. Обобщенная структура БЗ включает базу глубинных знаний, представляющую понятийные структуры ПрО, базу фактов, содержащую эмпирические данные о ПрО, базу метазнаний и совокупность интерфейсов между названными базами.

3. При реализации БЗ наиболее часто используются объектно-ориентированные и реляционные технологии, фреймовые и сетевые модели знаний, а также онтологический подход.

4. Система операций для работы со знаниями в БЗ является многоуровневой. Первый уровень образуют интерфейсные операции, второй — операции над базовыми информационными структурами, третий уровень включает комплексные операции верификации БЗ, поиска, извлечения, пополнения и систематизации знаний. На четвертом уровне располагаются операции, управляющие выполнением операций нижних уровней.

5. Основу для реализации операций второго уровня составляют логические приемы абстракции, конкретизации, формализации и интерпретации, обеспечивающие полноту в смысле формальной логики.

6. Интеллектуальная верификация знаний охватывает методы проверки выполнения базовых (независимых) ограничений целостности, методы анализа структурной семантики БЗ, методы анализа семантических зависимостей в БЗ и методы разрешения противоречий.

7. Методы анализа семантических зависимостей в БЗ основываются на рассмотрении полного множества возможных сочетаний элементов представлений пары свойств или отношений, принципе взаимообоснования свойств и отношений, а также выделении множества семантических признаков противоречивости.

8. В базе знаний могут быть реализованы механизмы разрешения противоречий. Соответствующие стратегии и связанные с ними классы операций строятся на базе аналогии с принятием решений человеком в конфликтных ситуациях. В них используется несимметричная интерпретация конфликтующих элементов описания как старого (известного) и нового знаний. Данные механизмы моделируют процессы накопления сомнений и обновления представлений о ПрО, отражаемых в БЗ.

9. Под операцией наследования в БЗ понимается процесс расширения (доопределения) описания вещи на основе знаний о ее соседях в модели представления ПрО. Операция наследования является комплексной. На ее последнем этапе осуществляется верификация полученного описания вещи.

Выделяются следующие разновидности наследования: свойств по свойствам, отношений по свойствам, свойств по отношениям и отношений по отношениям.

Вопросы для самопроверки

1. Что такое БЗ?
2. Чем различаются замкнутая и открытая БЗ?
3. Какие составляющие входят в обобщенную структуру БЗ?
4. Почему система операций для работы со знаниями в БЗ является многоуровневой?
5. Какие операции со знаниями в обобщенной модели представления ПрО *М*4 отнесены ко второму уровню, и каким требованиям они отвечают?
6. Какие виды операции абстракции выделены в *М*4?
7. Какая операция является обратной по отношению к операции абстракции?
8. Какая операция реализует переход от вещей к отношениям?
9. Какие типы операции формализации выделены в *М*4?
10. В чем суть операции интерпретации?
11. Чем различаются операции абстракции и формализации?
12. Как классифицируются методы интеллектуальной верификации знаний, предусмотренные в *М*4?
13. Что понимается под «сомнительными» семантическими структурами в БЗ?
14. На чем основан анализ семантических зависимостей в БЗ?
15. В чем заключается операция абстракции обоснований?
16. Как в *М*4 классифицируются бинарные противоречия?
17. Чем различаются сильные и слабые противоречия?
18. Что понимается под разрешением противоречий?
19. Что является исходной информацией для разрешения противоречия? Как интерпретируются конфликтующие элементы описания?
20. Как классифицируются стратегии разрешения противоречий?
21. В чем состоит разрешение противоречий в пространстве и во времени?
22. На основе чего выбирается стратегия разрешения противоречий?
23. Что такое показатель доминирования и как он определяется?
24. Опишите итерационный алгоритм совместного применения стратегий разрешения противоречий на фиксированном уровне. Какие процессы, присущие человеческому интеллекту, моделирует этот алгоритм?
25. В чем заключается разрешение противоречий с выходом на другие уровни?
26. Как определяется механизм наследования в БЗ?
27. Какие основные стадии включает процесс наследования?
28. Что такое комплексное наследование (замещение)?
29. Что лежит в основе классификации операций наследования?

Воображение является столь же эффективным средством успешного научного анализа, как и любая другая способность разума.

У. Черчмен, Р. Акоф, Л. Арнов

6. НЕЙРОННЫЕ СЕМИОТИЧЕСКИЕ СИСТЕМЫ

Главным содержанием технологии нейронных семиотических систем является создание электронных и программных аналогов естественных нейронных сетей и использование этих аналогов для имитации функций человеческого интеллекта. Данному направлению прочат исключительные перспективы в XXI в.

В главе рассмотрены основные понятия нейротехнологий, структура работ в области нейрокибернетики, классификация, характеристики и примеры нейропакетов, архитектура и критерии оценивания универсальных нейропакетов. Описан подход к моделированию сенсорных и языковой систем человека искусственными нейронными сетями.

Содержание главы соответствует направлению исследований в области ИИ 1.4.

6.1. Общая характеристика направления

Причины действий человеческих обыкновенно бесчисленно сложнее и разнообразнее, чем мы их всегда потом объясняем, и редко определенно очерчиваем.

Ф.М. Достоевский

В России сформировались три крупные *научные школы в области нейротехнологий.*

1. Центр нейрокомпьютеров РАН (А.И. Галушкин). Широкую известность он получил благодаря изданию 28-томной серии книг «Нейрокомпьютеры и их применение», а также регулярно проводимым Всероссийским конференциям с одноименным названием.

2. Научная школа нейротехнологий МГУ (А.В. Чечкин). Известность получили разработанные в ее рамках нейромодели сенсорных и языковых систем человека, относящиеся к бионическому направлению в нейрокомпьютинге.

3. Научная школа нейротехнологий в Красноярском государственном университете (А.Н. Горбань). Ученые этой школы развивают оригинальные алгоритмы обучения нейросетей и проводят международные конференции по нейрокомпьютингу.

Потенциальными сферами применения нейротехнологий являются все плохо формализуемые ПрО, в которых классические математические модели и алгоритмы оказываются мало эффективными по сравнению с человеком, демонстрирующим успешное решение задач. К областям использования нейротехнологий, в которых уже получены значимые практические результаты, относятся: обработка изображений, реализация ассоциативной памяти, системы управления реального времени, распознавание образов и речи, системы безопасности, выявление профилей интересов пользователей Internet, системы анализа финансового рынка и др. Сегодня можно уверенно говорить о том, что нейротехнологии становятся неотъемлемым компонентом НИТ.

Одной из характерных черт нейротехнологий является *обучение нейросети на примерах.* Эта же «технология» служит основой развития ребенка, который в первые годы жизни проходит гигантский путь формирования интеллекта и языковой системы, обучаясь на примерах.

Мозг состоит из различных типов клеток. Большинство нейрофизиологов считает, что объяснить феномены работы мозга можно, изучая функционирование объединенных в единую сеть клеток, называемых *нейронами.* Мозг включает 10^{10}—10^{11} нейронов. Количество связей между ними может достигать 10^{22}. Поэтому отображающие и моделирующие возможности нейросети огромны.

Модели нейронов, структуры ИНС, доказательство их обучаемости и алгоритмы обучения оставлены за рамками изложения. Перечисленные вопросы раскрыты в специальной литературе [227, 228, 230]. В данной главе рассмотрим архитектуру и характеристики нейропакетов, а также фундаментальную проблему разработки программных моделей высшей нервной системы (ВНС) человека.

Объем информации, хранящейся в мозге человека и других млекопитающих, превышает объем генетической информации, закодированной в ДНК. Строение мозга отражает его эволюцию. Наиболее древние участки мозга, доставшиеся человеку от рыб и амфибий, ответственны за поддержание жизнедеятельности (гомеостазис) и размножение. Другие отделы мозга (рептильный комплекс) возникли несколько сот миллионов лет назад и обеспечивают ориентацию в пространстве. Третий слой — лимбическая система — сформировался около 150 млн лет назад и отвечает за эмоциональную сферу. Наконец, кора больших полушарий, возникшая несколько миллионов лет назад, обеспечивает функции речи и логического мышления.

Обычно лишь 2...3 % нейронов мозга активны. Поэтому мозг обладает огромным запасом «прочности» и «пластичности», позволяющих ему работать даже при серьезных повреждениях и приспосабливаться к значительно меняющимся внешним условиям.

Нейрофизиологи в ВНС человека явно различают три типа нейронных структур: сенсорные, внутренние и эффекторные. Первые обеспечивают поступление в мозг информации из внешнего мира, поддерживая шесть чувств: зрение, слух, вкус, обоняние, осязание и чувство равновесия. Но если первые пять сенсорных систем «вынесены наружу», то вестибулярный аппарат изолирован и скрыт от непосредственного наблюдения. Внутренние нейроны воспринимают информацию от сенсорных систем, перерабатывают ее и передают на эффекторные нейроны. Последние управляют мышцами, внутренними органами, сосудами и пр.

Качественное различие компьютерной обработки символьной и образной информации заключается в малом числе разрядов цифрового представления символа. Поэтому для символов в принципе можно комбинаторно описать любой способ их обработки. Для образов (из-за комбинаторного взрыва) это в принципе невозможно за исключением ряда типовых преобразований, поддающихся формализации и реализуемых в графических ускорителях, пакетах обработки изображений и т. д. В общем случае операции с образами формализуемы не полностью. Для их обработки применяются эвристические алгоритмы. Главный механизм обработки образов — обучение на множестве примеров (обучающем множестве).

С учетом сказанного ***парадигма нейрокомпьютинга*** формулируется следующим образом: алгоритмы, порождаемые данными в универсальном процессе обучения, специализированные для данного класса операций с образами, адаптированные под конкретные информационные задачи [229].

В отличие от памяти ЭВМ память человека адресуется по содержанию, является ассоциативной, распределенной, робастной* и активной.

Полушария мозга человека имеют разное назначение. Левое полушарие отвечает за работу с абстрактными представлениями, математические вычисления и логический вывод, речь, письмо, восприятие времени. Эти процессы связаны с пошаговой обработкой информации. Правое полушарие оперирует с образами конкретных объектов. Все виденное нами в жизни хранится в правом полушарии, а все имена — в левом. Правое полушарие способно учиться узнавать предметы при их предъявлении, а не по описаниям, как левое. У правого полушария данные и метод составляют единое целое. Оно также отвечает за воображение и интуицию. Это более древние

* Робастность — способность системы восстанавливаться при возникновении ошибок и сбоев.

Рис. 6.1. Структура работ в области нейрокибернетики

механизмы мозга по сравнению с логическим мышлением. Правое полушарие поддерживает параллельную обработку информации, оно же отвечает за творчество.

К *феноменам мозга* относят:

- кодирование (представление) информации о внешнем мире;
- кратковременное и долговременное запоминание, хранение и извлечение информации;
- ассоциативный поиск и самоорганизацию памяти;
- оперирование информацией в процессе решения мыслительных задач;
- симультанное (мгновенное) распознавание (например, узнавание человека, с которым не было встреч много лет);
- неожиданное творческое озарение (инсайт) и др.

Конструктивного научного объяснения этим феноменам до сих пор не найдено.

Структура работ в области нейрокибернетики приведена на рис. 6.1.

Первое из выделенных направлений связано с построением и применением нейросетей из искусственных нейронов. Для их программной реали-

зации создаются *нейропакеты* (НП), а для их физической реализации — *нейрокомпьютеры* (НК). Исследования в области нейрофизиологии показывают, что в мозге имеются разные типы нейронов, выполняющих весьма сложные функции. В качестве примера назовем нейроны, ответственные за определение скорости и ускорения движущегося тела, выделение признаков замкнутости и разомкнутости кривой и др. Однако в ИНС, как правило, используются модели простых нейронов, реализующих элементарные нелинейные функции (функцию знака, сигмоидальные функции и т. д.).

В [228] сформулирован вывод, согласно которому, если для какого-то класса задач требуются сложные специализированные нейроны, их проще ввести в модель сразу, нежели строить из простых. Если же такая необходимость не очевидна, то целесообразно использовать универсальную сеть из простых нейронов. Она сама «вырастит» нужные структуры в ходе обучения. Заметим, что эти соображения лежат в основе декомпозиции НП на универсальные и специализированные (см. § 6.2).

Сравнение основных характеристик традиционных компьютеров и нейрокомпьютеров приведено в табл. 6.1.

Таблица 6.1

Основные характеристики	Традиционные компьютеры	Нейрокомпьютеры
Режим функционирования	В основном последовательный	Параллельный
Описание функционирования	Заданные алгоритмы	Алгоритмы формируются на основе обучения ИНС на примерах
Характер операций	Иерархическая структура алгоритмов. Разбиение сложных задач на простые составляющие. «Жесткие» математические модели	Непосредственное манипулирование образами. «Мягкие» математические модели
Аналог	Левое полушарие	Правое полушарие

Отметим следующие важные *особенности ИНС*.

1. ИНС содержит большое число (миллионы и миллиарды) параллельно работающих простых элементов — нейронов. Благодаря такой структуре обеспечивается высокое быстродействие при решении задач, традиционно требующих значительных вычислительных ресурсов.

2. Место программирования в ИНС занимает обучение. В связи с этим ожидается появление новых специальностей:

• нейроконструктора, в задачи которого входят формирование универсальных компонентов — нейронных блоков — и конструирование из них НК (универсальных и специализированных);

• учителя ИНС.

3. Выделяют два подхода к организации обучения ИНС:

• обучение и самообучение на примерах;

• обучение в процессе игры.

Достижениями в области обучения ИНС считаются:

• возможность управления процессом обучения в режиме двойственного функционирования ИНС, при котором НП (НК) самостоятельно оценивает ошибки и сообщает об этом учителю;

• возможность восприятия и отражения на ИНС информации из внешней среды.

Под *обучением ИНС* понимается процесс нахождения экстремума некоторой функции, отображающей взаимодействие типа вход-выход. Спецификой оптимизационных задач на ИНС является большая размерность, что обусловливает целесообразность применения алгоритмов оптимизации, которые допускают параллельное выполнение аналогично естественным нейросетям. Кроме методов оптимизации при обучении ИНС используют генетические алгоритмы.

Базовыми понятиями нейротехнологий являются:

• нейрон — базовый элемент сети, единственный ее нелинейный элемент;

• синапс — элемент, обеспечивающий связь между нейронами; может обладать весом и «задержкой»;

• сумматор, рассматриваемый как компонент нейрона или как специализированный нейрон;

• обучающие примеры, представляющие наборы значений вход— предписанный выход и целевую функцию, определяющую штраф за отклонение реального выхода от предписанных значений при данных входах;

• цвета и цветовые группы, ассоциируемые с нейронами (окрашенные нейроны по разному участвуют в обучении);

• алгоритмы обучения.

В нейротехнологиях обучается не отдельный нейрон, а вся сеть в целом. В ходе ее обучения предъявляются примеры и их оценки. Для быстрого параллельного обучения в сеть вводятся элементы, оценивающие частные производные. В естественных нейросетях такие элементы также существуют.

Выделяют три базовые подхода к представлению результатов обучения нейросетью:

• коннекционизм — модифицируются веса синаптических связей, параметры нейронов не меняются;

• гетерогенные ИНС — модифицируются параметры нейронов, связи не меняются;

• комплексный подход, объединяющий первые два подхода.

Основные выводы

1. Нейронные семиотические системы основаны на моделировании функций ВНС человека. Это направление получит исключительное развитие в XXI в.

2. Одной из ключевых технологий нейронных семиотических систем является обучение ИНС на примерах.

3. Основная парадигма нейрокомпьютинга: алгоритмы, порождаемые данными в универсальном процессе обучения, специализированные для данного класса операций с образами, адаптированные под конкретные информационные задачи.

4. Работы в области нейрокибернетики включают два базовые направления:

• разработку, реализацию и использование для решения практических задач математических моделей ИНС;

• разработку и реализацию математических моделей ВНС человека.

5. В рамках первого направления создаются НП и НК. Эти работы интенсивно развиваются, обеспечивая быстрый рост сферы коммерческих приложений нейротехнологий.

6.2. Нейропакеты

Масштаб дистанции между нейрокомпьютерами и обычными ЭВМ соответствует различиям между царствами живых организмов.

А.Н. Горбань

Нейропакетом называется программная система, эмулирующая среду НК на обычном компьютере.

Классификация НП предложена в [231]. В соответствии с ней НП делятся на семь основных классов.

1. НП для разработки других НП (инструментарий построения НП).

2. Универсальные НП. Под универсальностью понимается возможность моделирования ИНС разной структуры и с разными алгоритмами обучения.

3. Специализированные НП, использующие нейроны сложной функциональности и включающие специализированные средства для:

- обработки изображений;
- распознавания образов;
- распознавания рукописных и печатных символов;
- распознавания речи;
- управления динамическими системами;
- финансового анализа и др.

4. Нейронные ЭС.

5. Пакеты генетического обучения ИНС.

6. Пакеты нечеткой логики, использующие ИНС.

7. Интегрированные пакеты, использующие ИНС.

Примеры НП первого класса:

- OWL (разработчик — Hyper Logic Corp.);
- Neuro Windows (разработчик —Ward Systems Group);
- NNet+ (разработчик — NeuroMetric Vision System);
- Neural Network Toolbox for Matlab (разработчик — Math Works);
- Neuro Office (разработчик — ЗАО «АльфаСистем»).

Коротко охарактеризуем последний НП. Он предназначен для создания программных модулей, реализующих модель ИНС с ядерной организацией. Процесс разработки состоит из четырех этапов:

1) визуальное проектирование структуры и топологии ИНС;

2) определение синаптической карты и функций активации нейронов;

3) обучение построенной ИНС;

4) тестирование обученной ИНС (в том числе оценивание скорости работы ИНС).

Нейропакет Neuro Office позволяет сформировать обученную ИНС с программным интерфейсом на базе технологии COM. Пакет включает три основных компонента:

- NeuroView+ — редактор структуры и топологии ИНС;
- NeuroEmulator — средство для определения синаптической карты и функций активации нейронов, обучения и тестирования ИНС;
- ActiveX-элемент «нейронная сеть» — компонент, встраиваемый в приложения и реализующий программный интерфейс с ИНС.

В NeuroView+ можно одновременно редактировать несколько ИНС. Размеры формируемых ИНС программно не ограничены. В файл с описанием ИНС также заносятся данные о графическом представлении сети на экране дисплея и комментарии разработчика.

Система NeuroEmulator, как и NeuroView+, позволяет одновременно работать с несколькими ИНС. Предусмотрены четыре способа инициализации синаптической карты ИНС (рис. 6.2): ручное определение синаптиче-

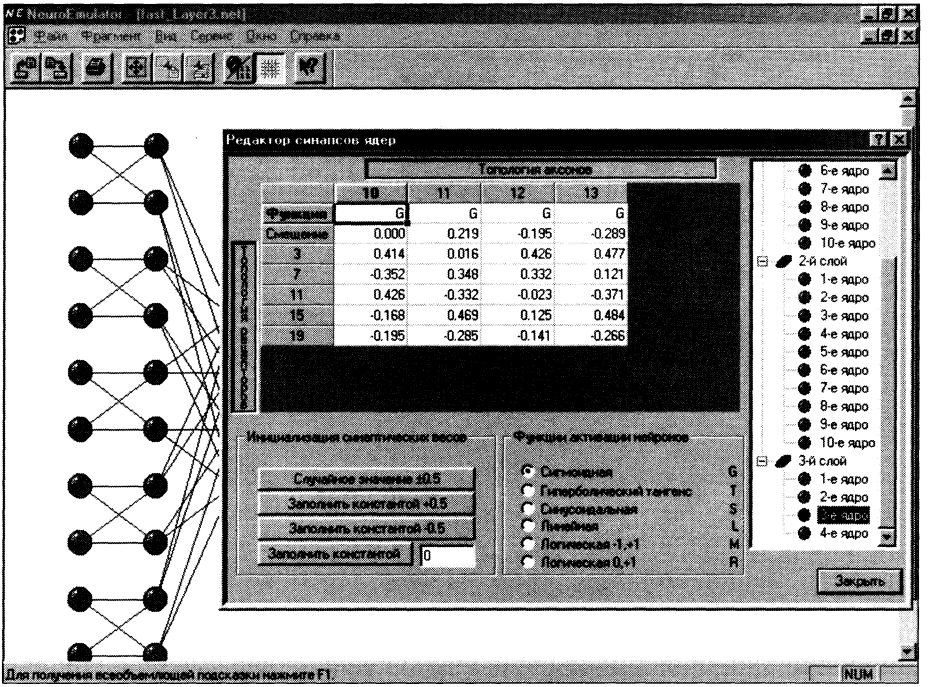


Рис. 6.2. Режим редактирования синаптической карты и функций активации нейронов в системе NeuroEmulator

ских весов; случайный выбор синаптических весов (±0,5); заполнение синаптической карты указанной константой; статическое обучение ИНС.

Обучение ИНС (рис. 6.3) основано на методе обратного распространения ошибки. В системе реализованы два режима обучения: индивидуальное обучение выбранному примеру и обучение на множестве примеров. На любом этапе обучения пользователь может просмотреть и изменить синаптическую карту ИНС. Процесс обучения иллюстрируется изменением цветов в графическом представлении ИНС.

Для тестирования обученной ИНС предназначен режим эмуляции нейрообработки на входном наборе данных, состоящем из одного или множества примеров. При тестировании на отдельном примере можно контролировать преобразование данных по слоям ИНС. Результаты тестирования представляются в табличном и графическом виде.

Примером интегрированного НП служит ИПС Excalibur RetrievalWare, рассмотренная в § 3.6.

Работа с НП включает два базовых этапа. На первом этапе осуществляется обучение ИНС на примерах. На втором — обученная ИНС применя-

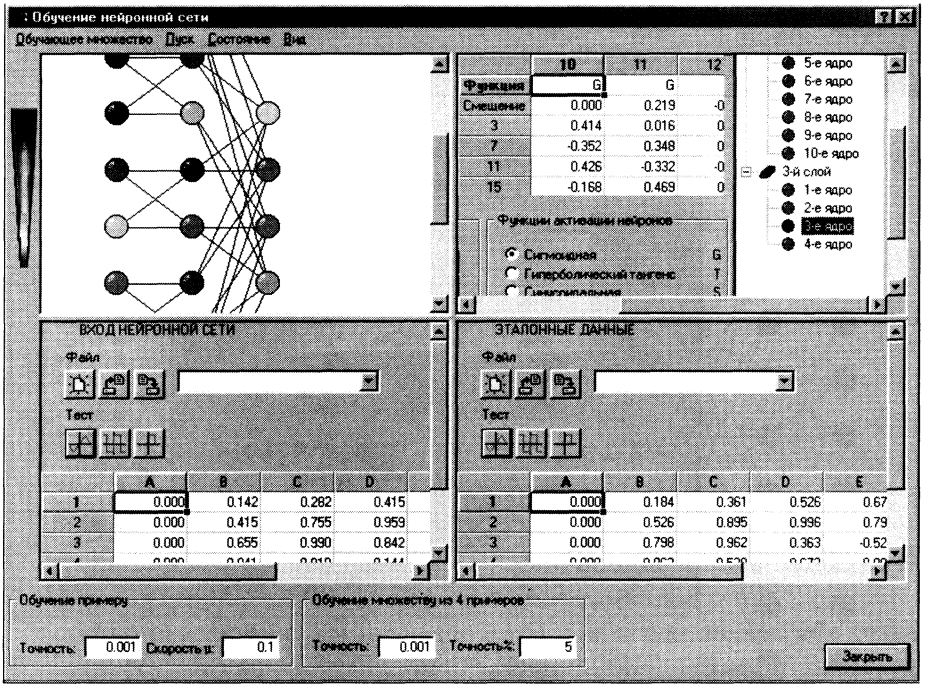


Рис. 6.3. Режим обучения нейронной сети в системе NeuroEmulator

ется для решения прикладных задач. После накопления и обобщения опыта использования программных реализаций нейросети в рамках НП может создаваться НК.

Аппаратной базой для НП служат рабочие станции или персональные ЭВМ, обладающие высокой производительностью.

Архитектура универсального НП представлена на рис. 6.4. Укрупненные структуры подсистемы формирования ИНС и подсистемы проведения экспериментов с ИНС изображены на рис. 6.5 и 6.6.

Из-за ограниченного объема пособия технологии создания НП подробно не рассматриваются. Остановимся на важном для пользователей вопросе *сравнения универсальных НП.* Результаты подобного исследования приведены в [231]. Для анализа выбраны пять наиболее мощных на начало 2000 г. зарубежных НП:

- НП1 — NeuroSolutions (разработчик – NeuroDimension, Inc.);
- НП2 — NeuralWorks Professional II/Plus (разработчик — NeuralWare, Inc.);
- НП3 — Process Advisor (разработчик — AI Ware, Inc.);
- НП4 — NeuroShell 2 (разработчик — Ward Systems Group);

Рис. 6.4. Архитектура универсального нейропакета

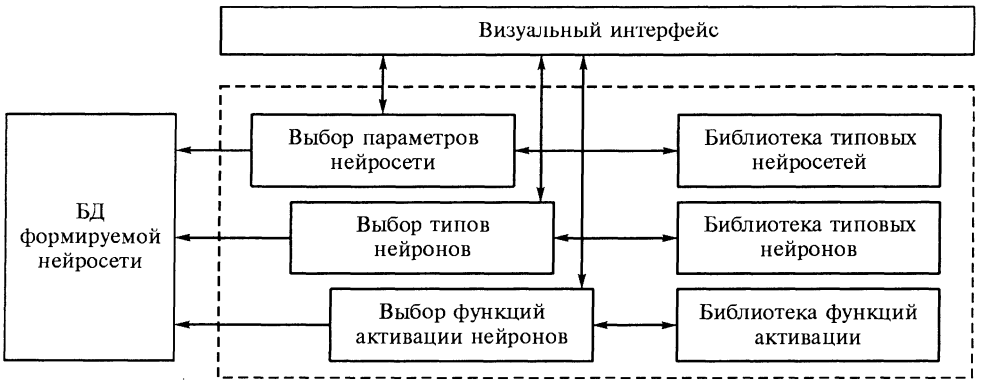


Рис. 6.5. Укрупненная структура подсистемы формирования нейросети

• НП5 — BrainMaker Pro (разработчик — California Scientific SoftWare).

При тестировании НП использовались IBM-совместимые персональные компьютеры. Все НП работали с моделью многослойной ИНС.

Главным показателем эффективности функционирования НП служит *скорость обучения ИНС*. Поскольку во всех рассматриваемых НП механизм

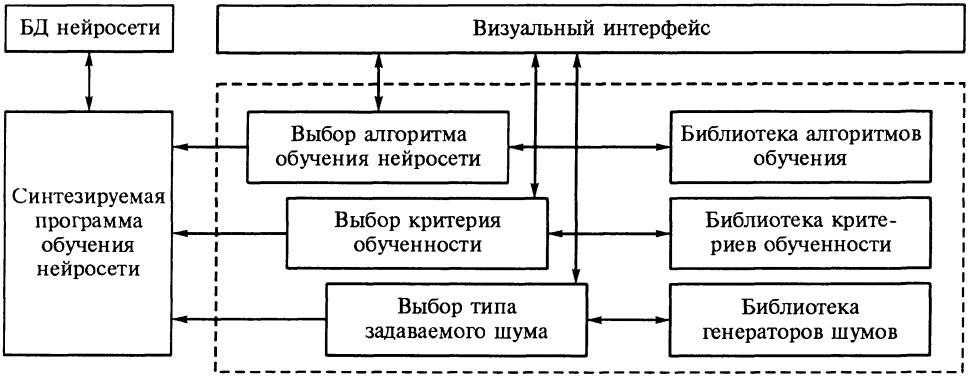


Рис. 6.6. Укрупненная структура подсистемы проведения экспериментов с нейросетью

обучения реализован на основе алгоритма обратного распространения ошибки, данный показатель у них принимает близкие значения. С учетом сказанного были выделены 11 дополнительных критериев, оцениваемых по 10-балльной шкале:

1) простота формирования и обучения ИНС при использовании интуитивно понятного графического интерфейса НП;

2) простота подготовки обучающей выборки;

3) наглядность и полнота представления информации в процессе формирования и обучения ИНС;

4) состав поддерживаемых нейронных моделей, критериев и алгоритмов обучения;

5) возможность создания собственных (т. е. нетиповых) нейронных структур;

6) возможность использования собственных критериев оптимизации;

7) возможность использования собственных алгоритмов обучения ИНС;

8) простота обмена информацией между НП и другими приложениями;

9) открытая архитектура пакета (возможность его расширения за счет внешних программных модулей);

10) наличие генератора исходного кода;

11) наличие макроязыка для ускорения работы с НП.

Названные критерии можно разделить на три класса:

- первые три критерия оценивают НП с точки зрения начинающих пользователей;
- критерии с четвертого по восьмой оценивают НП с точки зрения опытных пользователей;
- последние три критерия оценивают НП с точки зрения профессиональных разработчиков НП.

Экспертная оценка НП на основе перечисленных критериев приведена в табл. 6.2 [231].

Таблица 6.2

Критерии сравнения	Нейропакеты				
	НП1	НП2	НП3	НП4	НП5
1	9	9	8	10	6
2	9	9	7	8	7
3	10	9	7	6	4
4	8	10	5	8	6
5	10	8	5	5	5
6	8	7	0	0	0
7	10	7	3	0	0
8	10	8	5	8	5
9	10	10	3	2	0
10	10	10	0	10	0
11	10	0	0	0	0
Суммарная оценка	104	87	43	57	33

Отметим достоинства лучшего пакета НП1:

- поддержка традиционно используемых моделей — полносвязных многослойных ИНС и самоорганизующихся карт Кохонена [233];
- наличие мощного редактора визуального проектирования ИНС;
- возможность простого задания собственного алгоритма обучения и критериев обученности ИНС;
- наличие хорошо продуманных средств визуализации (НП1 позволяет визуализировать множество моделей и характеристик — структуру ИНС, процесс и результаты обучения и т. д.).

Нейропакет НП1 функционирует на базе операционных систем семейств Win9x и WinNT, поддерживает OLE2, содержит генератор программного кода на C++. Он имеет открытую архитектуру: можно реализовать надстройку над пакетом с помощью макроязыка и подключать внешние модули при проектировании и обучении ИНС.

Нейропакет НП1 поддерживает три типа нейронов:

- взвешенный сумматор (нейрон 1-го порядка);
- нейроны высших порядков с перемножением кодов;
- интегрирующие нейроны.

Для активации нейронов используются пять функций: 1) стандартная функция знака; 2) кусочно-линейная функция; 3) три сигмоидальных функции произвольного вида.

Искусственная нейтронная сеть, с которой работает НП1, может содержать прямые, перекрестные и обратные связи между нейронами. При организации связей в многослойной ИНС используются так называемые векторные связи. Они задаются матрицей весов, а не набором скалярных связей с весовыми коэффициентами. Это значительно увеличивает производительность при подготовке данных для обучения ИНС и повышает наглядность визуального представления ИНС, так как вместо множества скалярных связей отображается одна обобщенная, соответствующая векторной связи.

При формировании обучающих выборок, благодаря наличию встроенных конвертеров данных, НП1 позволяет предъявлять ИНС как графические изображения (в формате BMP), так и текстовые файлы, содержащие числовые и символьные данные, а также описания функций непрерывных аргументов (в том числе времени). Исходные данные для эксперимента могут быть представлены в виде аналитических выражений или выборок числовых значений.

На этапе обучения используются дискретные и непрерывные критерии обученности ИНС (последние важны для ИНС с интегрирующими нейронами). Помимо традиционного квадрата ошибки можно использовать критерии из функционального пространства L_p (для $p = 2$ имеем квадрат ошибки).

Нейропакет НП1 позволяет накладывать на исходные данные аддитивный белый шум, а также любой заданный пользовательским приложением тип шума.

В качестве примера специализированного НП отметим продукт Excel Neuro Package компании «НейрОК Интелсофт». Данный НП включает два компонента:

- Winnet 3.0 — программный эмулятор многослойной нейросети;
- Kohonen Map 1.0 — средство построения самоорганизующихся карт Кохонена.

Названные компоненты реализованы в виде программных надстроек (add-ins), расширяющих возможности Microsoft Excel. Их пользовательский интерфейс иллюстрирует рис. 6.7.

Компонент Winnet 3.0 представляет собой инструмент статистического анализа и прогнозирования, основанный на использовании ИНС и позволяющий выявлять скрытые зависимости в больших массивах числовой информации.

Компонент Kohonen Map 1.0 служит средством визуализации и анализа многомерных данных, отображаемых в виде наглядной двумерной цветной карты. Он предназначен для решения задач кластеризации и визуализации многомерной информации.

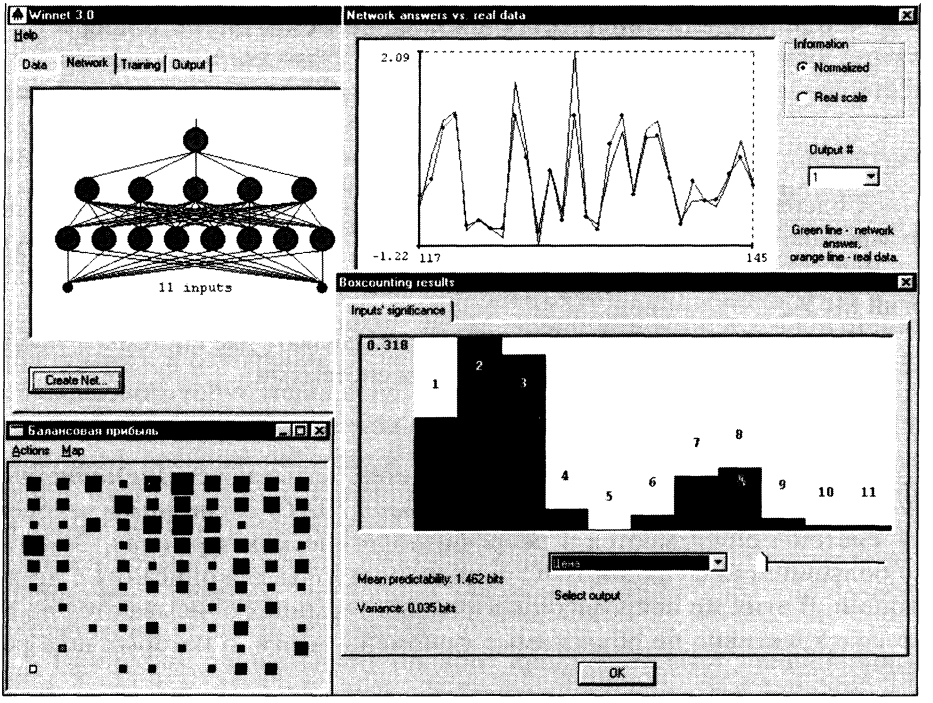


Рис. 6.7. Пакет Excel Neuro Package

Основные выводы

1. Разработка НП — наиболее быстро развивающаяся область нейротехнологий.

2. Нейропакеты подразделяют на семь основных классов.

3. Фундамент архитектуры универсального НП составляют подсистема формирования нейросети, БД нейросети и подсистема проведения экспериментов с нейросетью.

4. Существует развитая система критериев сравнения универсальных НП, отражающая интересы начинающих и опытных пользователей, а также профессиональных разработчиков НП.

6.3. Модели сенсорных и языковой систем человека

Мозг рационален, сознание — может быть, нет.

Дуглас Р. Хофштадтер

Содержание данного параграфа базируется на интересных и перспективных гипотезах, лежащих в основе подхода к моделированию ряда функций ВНС человека. Этот подход развивается научной школой нейротехнологий МГУ.

В высшей нервной системе человека выделяют две системы функций, реализуемые нейронными семиотическими системами:

- сенсорные системы;
- языковая система.

Рассматриваемые модели этих систем называются ***элементарной сенсорной (ЭСС)*** и ***элементарной языковой (ЭЯС) системами*** человека. Эти системы описываются в нейрофизиологических терминах. Заметим, что большинство функций ВНС пока не доступны для описания с других позиций. В этих же нейрофизиологических терминах описываются до сих пор конструктивно не объяснимые феномены мозга, о которых шла речь в § 6.1.

Как известно, человек обладает шестью сенсорными системами. Все они, кроме обонятельной, являются многоканальными. Например, зрительная система имеет два канала: ахроматический, обеспечивающий анализ формы, и цветовой. Для упрощения дальнейшее изложение строится на примере одноканальной обонятельной системы. Схема модельного подхода представлена на рис. 6.8.

Сформулируем *основные парадигмы моделирования сенсорных систем:*

- общие принципы моделирования отдельных сенсорных каналов нейросетями;
- иерархическая организация ИНС;
- выделение в ИНС специальных групп нейронов;
- однородность способов восприятия сенсорными системами внешнего мира.

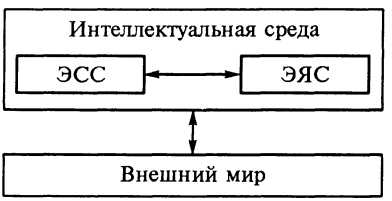


Рис. 6.8. Схема модельного подхода

Сенсоризм рассматривается как интерфейс между внешним миром и интеллектуальной средой. С помощью речи нейросети отдельных индивидуумов «объединяются в единую сеть». В результате обучение происходит быстрее, и может быть организована специализация индивидуумов.

В основе подхода к моделированию лежат следующие основные идеи.

1. Сенсоризм как часть мозга от рецепторов до эффекторных структур по отношению к внешнему миру (внешней ПрО) понимается как отображающая система (модель).

2. Внешний мир организован, не хаотичен, существует объективно, имеет многоуровневую иерархическую структуру и эволюционирует.

3. Отображающая система (сенсоризм) организована иерархически таким образом, чтобы соответствовать внешнему миру.

4. По модельным средствам сенсоризм является нейронным, т. е. объекты представляются нейронами (а не молекулярными и субклеточными структурами), а отношения — связями между нейронами, реализуемыми в возбуждающих/тормозящих нейромедиаторных взаимодействиях.

Мир невообразимо сложен, поэтому возникает вопрос, можно ли его представить внешней нейронной моделью? Известно, что информационная емкость мозга колоссальна и вполне покрывает множество объектов и связей между ними, которые человек наблюдает в течение жизни. Кроме того, в ходе исследований в области нейрофизиологии открываются новые специализированные нейроны, способные выполнять весьма сложные функции. Поэтому можно сделать два вывода.

Во-первых, модельный подход предусматривает интерпретацию структурной и функциональной организации нейронов как нейронной семиотической модели мира. При этом нейроны выступают в качестве материальных знаков объектов, а связи между нейронами — в качестве отношений между объектами.

Во-вторых, нейронная модель является не только моделью внешнего мира, но и организма, служащего носителем нейросети, и его поведения, т. е. понятие субъекта занимает центральное место в модели. В качестве внешней ПрО для нейронной системы выступают внешний мир (по отношению к субъекту) и внутренний мир субъекта.

При построении ЭСС и ЭЯС выдвигаются две нейрофизиологические гипотезы:

- ЭЯС морфологически и функционально параллельна ЭСС;
- во время работы ЭЯС и ЭСС отображаются одна в другую.

Моделируемая далее обонятельная система действует по принципу разложения запаха на простые составляющие. Система обрабатывает информацию о запахе. Главным качеством для нее является оттенок запаха.

Исследования показывают, что одноканальная сенсорная система сохраняет все принципы многоканальной. Запаховые характеристики кодируются на основе иерархической схемы. Множество запахов, потенциально свойственных объекту внешнего мира, разбиваются на группы, которые, в свою очередь, делятся далее. В таком виде в модели они поступают на за-

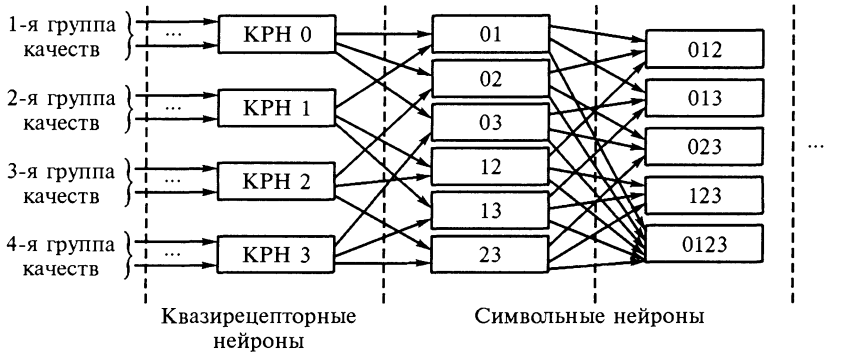


Рис. 6.9. Организация одного уровня модели обонятельной сенсорной системы:
КРН — квазирецепторные нейроны

паховые квазирецепторы. С них сигналы, соответствующие комбинациям качеств запаха, передаются на нейроны коры мозга (рис. 6.9).

В ЭСС нейроны делятся на *квазирецепторные* (КРН) и *символьные* (СН). Символьные нейроны переходят в возбужденное состояние, когда на их входе срабатывает определенная комбинация сигналов КРН (рис. 6.10).

Дальнейшая детализация модели основана на гипотезе о том, что СН в ЭСС, представляющим некоторые образы объектов внешнего мира, в ЭЯС соответствуют также СН, представляющие понятия. Модель такого типа называется ***элементарной сенсорной моделью (ЭСМ).*** В ней каждому иерархическому уровню ПрО соответствует определенный *синаптический уровень.* Простые стимулы представлены КРН, а сложные — СН. Нейроны могут проецироваться на другие нейроны, расположенные на другом уровне

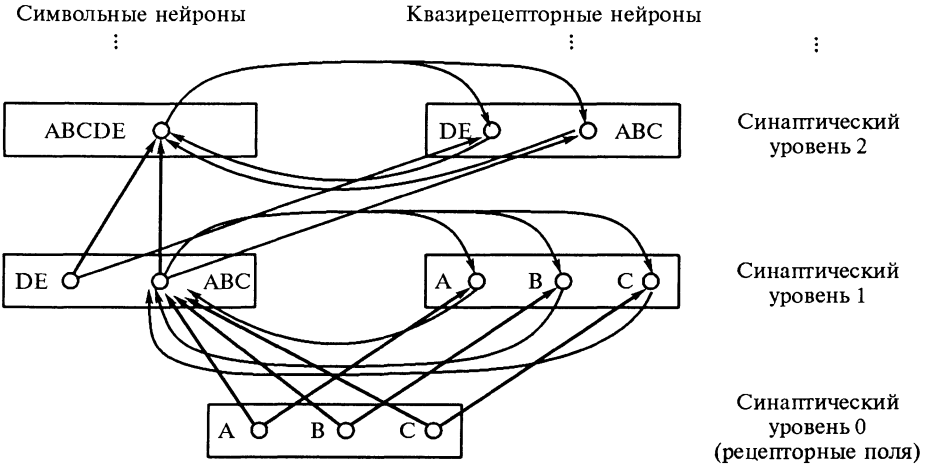


Рис. 6.10. Организация элементарной сенсорной системы

или в другой системе. *Проекции* позволяют нейронам активировать друг друга. На данном синаптическом уровне СН образуются за счет конвергенции, т. е. сочетания проекций нейронов предыдущих уровней, а КРН образуются на основе проекций нейронов предыдущего уровня по принципу «один к одному».

В элементарной сенсорной модели не отражены различные вставочные нейроны, выполняющие функции контрастирования, выделения сигнала из шума, усиления, извлечения, воспроизведения и т. д.

Можно показать, что ЭСМ адекватно представляет следующие механизмы:

- распознавание образов;
- воспроизведение образов;
- кратковременная и долговременная память;
- ассоциативная память;
- формирование и оперирование абстрактными понятиями и др.

Рассмотрим некоторые из этих механизмов.

Воспроизведение образов. Различают два подобных механизма. Первый заключается в воспроизведении прямого информационного значения СН (денотата) путем проекции СН на соответствующие КРН. Второй состоит в воспроизведении смыслового информационного содержания образа (сигнификата). Данные механизмы работают параллельно. Они активизируются, когда возникает необходимость воспроизведения образа.

Рассматриваемая модель является достаточно общей. Она иллюстрирует принцип функционирования ЭСМ, но не отражает множество важных деталей. В частности, она не раскрывает:

- способы активации механизма (когда он срабатывает, что вызывает его действие);
- механизм затухания воспроизведения, тормозящие его связи;
- продолжительность воспроизведения.

Заметим, что активный СН может активировать связанные с ним СН. Этот механизм обеспечивает воспроизведение контекста ситуации.

Механизмы памяти. Сеть СН в ЭСС выступает в качестве долговременной памяти. Эта память может функционировать отдельно от рецепторных полей, поддерживая механизмы воспроизведения образов. Актуализация соответствующих нейронов, вызываемая активностью рецепторных полей, обеспечивает опознание и извлечение информации о соответствующем фрагменте ПрО из долговременной памяти.

Формирование долговременной памяти состоит из кодирования и запоминания. Считается, что этот процесс в ЭСС может быть реализован как обучение. Механизмы кодирования и запоминания определяются генетической программой и развиваются еще до рождения человека в онтогенезе.

Высшая нервная система человека может создавать модели виртуальных, воображаемых ПрО (человек сам формирует у себя новые СН). Построение таких виртуальных ПрО лежит в основе механизмов творчества, фантазии, понимания.

Кратковременная память в рамках ЭСМ — сохранение возбужденного состояния СН какое-то время после их актуализации. Существуют механизмы сохранения повышенной возбудимости, и их можно тренировать.

Заметим, что нейроны в ЭСС по сравнению с элементарными формальными нейронами в ИНС много сложнее. Они могут быть представлены сетями из формальных нейронов.

Синаптическая организация ЭСС позволяет нейронной системе работать в режиме воспоминания, не требующем активности КРН. Эта способность создает условия для оперирования абстрактными понятиями.

Феномен вербального мышления. Существует предположение, согласно которому обучение языку, т. е. формирование *нейронов-слов,* является более быстрым процессом по сравнению с формированием СН в ЭСС. Кроме того, оно способствует формированию ЭСС. Данные закономерности объясняются тем, что ЭЯС надстраивается над слуховыми и зрительными рецепторными полями, в то время как ЭСС охватывает весь сенсоризм.

Расширим рассматриваемую модель, объединив ЭСС и ЭЯС (рис. 6.11). Рецепторному уровню в ЭЯС соответствует уровень рецепторных нейронов-слов. На следующем синаптическом уровне ЭЯС конвергирующие проекции нейронов-слов обеспечивают выборочное активирование *нейронов-понятий.* Часть нейронов-слов имеет прямые проекции на нейроны-понятия. Последние, как и СН в ЭСС, обладают проекциями на соответствующие квазирецепторные нейроны-слова. Это позволяет нейронам-понятиям воспроизводить свое денотационное содержание. Данный процесс называется вербали-

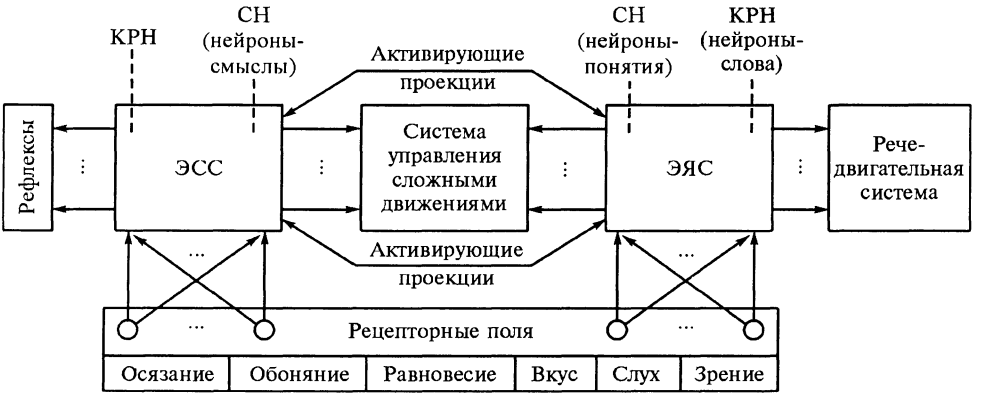


Рис. 6.11. Взаимодействие элементарной сенсорной и элементарной языковой систем

зацией (озвучиванием). В свою очередь, связи нейронов-понятий между собой позволяют воспроизводить сигнификативное информационное содержание нейронов-понятий.

Элементарные сенсорная и языковая системы имеют собственные эффекторные (двигательные) системы, управляющие мускулатурой тела и речевого аппарата.

Символьные нейроны в ЭСС и нейроны-понятия в ЭЯС взаимосвязаны активирующими проекциями. Это обеспечивает перевод образов в слова и обратно.

Процесс мышления протекает в каждой из систем (ЭСС, ЭЯС). Правостороннее и левостороннее мышление определяются доминированием ЭСС либо ЭЯС.

Осознание (как понимание) представляет собой актуализацию СН в ЭСС и (или) нейронов-понятий в ЭЯС. Решение мыслительных задач сводится к опознанию и воспроизведению полной картины по ее части (если решение знакомо) путем актуализации разных частей модели. Истинно творческим моментом в процессе мышления (протекающем как в ЭСС, так и в ЭЯС) является достраивание модели путем образования СН в ЭСС или нейронов-понятий в ЭЯС.

Основные выводы

1. Модели сенсорных и языковой систем человека базируются на экспериментальных данных нейрофизиологических исследований функций ВНС человека.

2. В высшей нервной системе выделяют сенсорные и языковую системы, описываемые нейронными семиотическими моделями.

3. Построение ЭСС и ЭЯС базируется на модельном подходе, предусматривающем интерпретацию структурной и функциональной организации нейронов как нейронной семиотической модели мира.

4. В элементарной сенсорной системе нейроны делятся на квазирецепторные и символьные. Символьные нейроны представляют образы объектов внешнего мира, в ЭЯС им соответствуют также СН, представляющие понятия.

5. В настоящее время проверяются гипотезы об адекватном воспроизведении ЭСС и ЭЯС основных механизмов ВНС человека: распознавания образов, ассоциативной памяти, формирования и оперирования абстрактными понятиями и др.

Вопросы для самопроверки

1. На какой парадигме основан нейрокомпьютинг?
2. Что обычно относят к феноменам мозга?

3. Опишите структуру работ в области нейрокибернетики.
4. В чем различие между НП и НК?
5. Что понимается под обучением ИНС? Какую роль оно играет в нейротехнологиях?
6. Какие существуют подходы к представлению результатов обучения ИНС?
7. Перечислите основные классы НП.
8. Назовите основные модули, входящие в архитектуру универсального НП.
9. Перечислите основные функции подсистемы формирования нейросети и подсистемы проведения экспериментов с нейросетью универсального НП.
10. По каким критериям сравнивают универсальные НП?
11. Какие идеи и предположения лежат в основе моделирования функций ВНС человека?
12. Что такое сенсоризм?
13. В чем состоит различие между КРН и СН?
14. Охарактеризуйте организацию ЭСС.
15. Чем различаются ЭСС и ЭЯС?
16. Опишите схему взаимодействия ЭСС и ЭЯС.
17. Чем различаются нейроны-слова и нейроны-понятия?
18. Как соотносятся ЭСС и ЭЯС с правополушарным и левополушарным мышлением?
19. Как объясняются феномены мозга на основе взаимодействия ЭСС и ЭЯС?

Человек вырастает по мере того, как растут его цели.

Ф. Шиллер

7. СИСТЕМЫ УПРАВЛЕНИЯ ЗНАНИЯМИ

Управление знаниями представляет собой интегрирующую интеллектуальную информационную технологию, которая объединяет в единый комплекс множество технологий, поддерживающих процессы формирования, накопления, хранения, распространения, обработки и использования знаний и данных в рамках организации.

В главе рассмотрены основные понятия систем управления знаниями и их технологического фундамента — хранилищ данных. Представлена классификация методов интеллектуального анализа данных, охарактеризованы методы интерактивной аналитической обработки данных и глубинного анализа данных. Описан важный класс систем управления знаниями – системы поддержки инновационной деятельности.

Содержание главы соответствует направлению исследований в области ИИ 4.10.

7.1. Общая характеристика направления

Кто владеет информацией, тот владеет миром.

У. Черчилль

Понятие *«управление знаниями»* появилось в середине 90-х годов прошлого века. Возникновение этого направления интеллектуальных информационных технологий вызвано потребностями пользователей корпоративных ИС. Особенно остро они ощущаются в крупных компаниях, БД которых содержат сотни гигабайт структурированной и неструктурированной информации, накопленной за многие годы. В современной экономике конкурентные преимущества в значительной степени определяются интеллектуальным багажом компании, т. е. знаниями, располагаемыми ее сотрудниками. Традиционные корпоративные ИС оперируют не знаниями, а данными — документами, записями в БД, выборками, отчетами и т. п. В результате знания, служащие основным компонентом потенциала компании, такими системами непосредственно не учитываются и не обрабатываются.

Управление знаниями рассматривается как совокупность процессов, управляющих созданием, распространением, обработкой и использованием знаний в рамках организации [238]. *Система управления знаниями (СУЗ)* должна обеспечивать:

- отражение изменений данных в корпоративной БД, характеризующих историю деятельности компании;
- извлечение, интеграцию и представление в явном виде знаний специалистов компании;
- представление информации, содержащейся в корпоративных БД, на семантическом уровне;
- анализ и извлечение знаний из данных в корпоративных БД;
- поиск и доступ к информации по смыслу;
- поддержку совместной работы с ИР специалистов компании;
- поддержку процессов формирования новых знаний.

Корпоративные знания условно можно разделить на три слоя:

1) формализованные знания, представленные в БЗ;

2) знания, содержащиеся в документах и БД;

3) профессиональные знания специалистов компании, не зафиксированные на материальных носителях («знания в головах» сотрудников).

В число задач СУЗ входит поддержка процессов:

- явного выражения (фиксации) знаний специалистов, т. е. перевода их с 3-го слоя на 2-й или 1-й с помощью методов приобретения знаний от экспертов [239];
- формализации и автоматизированного извлечения знаний из ИР, т. е. перевода их со 2-го слоя на 1-й.

Наличие явного представления знаний обеспечивает их сохранение при уходе из компании специалистов, упрощает обучение новых кадров, создает условия для фиксации прав интеллектуальной собственности на знания, за счет чего повышается капитализация компании.

Управление знаниями следует рассматривать как *интегрирующую технологию,* объединяющую в комплекс множество информационных технологий (как традиционных, так и интеллектуальных):

- БД, хранилищ данных и БЗ;
- управления документооборотом;
- поддержки совместной работы с ИР;
- автоматизированного извлечения знаний из текста;
- поиска в текстовой и структурированной информации (в том числе поиска по метаданным);
- автоматической классификации и кластеризации документов;
- приобретения знаний от экспертов;
- машинного перевода;

- автоматического реферирования и аннотирования;
- интеллектуального анализа данных;
- автоматического распознавания образов;
- поддержки принятия решений;
- поддержки инновационной деятельности (формирования новых знаний) и др.

Некоторые из этих технологий были рассмотрены в предыдущих главах. Предметные особенности СУЗ как корпоративной системы отражают технологии, поддерживающие процессы принятия решений и инновационную деятельность.

Следует отметить, что существующие в настоящее время продукты, относимые их разработчиками к классу СУЗ (например, Fulcrum, Documentum i4, Knowledge Station), воплощают лишь отдельные технологии из приведенного выше перечня.

Фундаментом СУЗ служат технологии хранилищ данных и БЗ на основе онтологического подхода. В последние годы на базе технологии хранилищ данных была сформирована концепция корпоративной памяти (corporate memory). Ее обобщенная структура представлена в табл. 7.1. Хранилища данных являются средой для реализации методов интеллектуального анализа данных. Эти технологии рассматриваются в следующем параграфе.

Таблица 7.1

Уровень представления информации	Вид информации		
	Документы	Данные	Знания
Онтологический	Структуры архивов	Структуры данных	Базовые онтологии
Содержательный	Отчеты, методики, инструкции	Справочники, каталоги	Правила вывода, факты
Программно реализованный	Документы (тексты, рисунки, фотографии, схемы)	БД, файлы	БЗ

Внедрение СУЗ в организациях, значительное число сотрудников которых занято обработкой информации, приносит ощутимый экономический эффект. Так, под данным компании «НейрОК Интелсофт» СУЗ позволяет ежедневно экономить в среднем 40...50 мин. рабочего времени одного сотрудника*, что эквивалентно повышению производительности труда на 8...10 %. Таким образом, общий выигрыш от использования СУЗ составит 8...10 % от соответствующего фонда заработной платы (без учета начисле-

* Данная оценка, на наш взгляд, является осторожной.

ний). Легко подсчитать, что СУЗ, стоимость которой равна месячному фонду заработной платы, окупится примерно за год. Например, для организации, имеющей 30 специалистов, занятых обработкой информации, со средней зарплатой 850 долл., обосновано приобретение СУЗ стоимостью около 25 тыс. долл.

Основные выводы

1. Управление знаниями — одна из наиболее интенсивно развивающихся областей коммерческого применения интеллектуальных информационных технологий.

2. Система управления знаниями предназначена для управления созданием, распространением, обработкой и использованием знаний в рамках организации.

3. Управление знаниями рассматривается как интегрирующая технология, объединяющая множество информационных технологий.

4. Фундаментом СУЗ служат технологии хранилищ данных и БЗ на основе онтологического подхода.

Вопросы для самопроверки

1. Что понимается под управлением знаниями?
2. Что обеспечивает СУЗ для корпорации?
3. На какие слои разделяются корпоративные знания, и какие задачи СУЗ связаны с ними?
4. Какие информационные технологии интегрирует СУЗ?
5. Что является технологическим фундаментом СУЗ?

7.2. Технологии хранилищ данных и интеллектуального анализа данных

Узнать больше можно, только меняя свою позицию.

Жак Валле

7.2.1. Основные понятия

Понятие ***хранилища данных*** было введено Б. Инмоном, определившим его как предметно-ориентированное, привязанное ко времени и неизменяемое собрание данных для поддержки принятия управляющих решений [251]. Хра-

нилище данных представляет собой репозиторий, содержащий непротиворечивые консолидированные исторические данные корпорации, отражающие ее деятельность за достаточно продолжительный период времени, а также данные о внешней среде ее функционирования [245].

Объем данных в хранилище как минимум на порядок превосходит объемы данных в оперативных БД (так называемых OLTP-системах). Бóльшей сложностью отличаются и запросы к хранилищу. Названные особенности обусловливают необходимость обеспечения:

- высокой производительности обработки запросов;
- масштабируемости используемых алгоритмов.

При загрузке в хранилище новых данных должна выполняться их верификация, включающая:

- выявление и устранение ошибок, а также нарушений ограничений целостности;
- выявление и разрешение противоречий в данных, поступающих из разных источников;
- выявление и устранение избыточности в данных и т. д.

Соответствующие вопросы были рассмотрены в § 5.5 на примере обобщенной модели представления знаний о ПрО $M4$.

В архитектурном плане хранилище данных может включать два или три уровня. В первом случае на верхнем уровне располагается обобщенная информация для руководителей всех подразделений предприятия, которым требуются средства анализа данных. Нижний уровень занимают источники данных, в том числе БД оперативной информации. В трехуровневой архитектуре над двухуровневым хранилищем организуются специализированные хранилища данных для отдельных подразделений.

Анализ данных в хранилищах базируется на технологиях ***интеллектуального анализа данных (ИАД).*** Целью ИАД является извлечение знаний из данных, т. е. обнаружение в исходных данных ранее неизвестных нетривиальных практически полезных и доступных для интерпретации знаний, необходимых для принятия решений в различных ПрО.

Возможности применения конкретного метода ИАД определяются характером извлекаемых им знаний и используемым представлением исходных данных. Наиболее распространенный тип знаний, извлекаемых с помощью технологий ИАД, — это закономерности ПрО. В зависимости от их характера ПрО можно разделить на три группы:

1) ПрО с доминированием случайных событий;

2) ПрО, в которых все события причинно обусловлены;

3) ПрО, в которых наблюдаются как причинно обусловленные, так и случайные события.

В первой группе ПрО преобладают частотные закономерности, во

второй — жесткие причинно-следственные зависимости, в третьей — причинно-следственные зависимости, допускающие исключения.

Как правило, методы ИАД оперируют с данными, представленными тремя основными способами:

- атрибутивным (объекты описываются значениями фиксированного набора атрибутов);
- структурным (объекты определяются типологически);
- полнотекстовым (исходными данными служат тексты на ЕЯ).

Методы ИАД подразделяют на три класса [243].

1. Алгебраические методы. Исходные данные в них представляются в виде алгебраических структур.

2. Статистические методы. Они используют аппарат теории вероятностей и математической статистики. Примером может служить GUHA-метод, в котором применяются статистические кванторы и статистические критерии корреляции.

3. Методы мягких вычислений. В них используются нечеткое представление данных и нейросети (см. гл. 6).

Традиционно принято считать, что исходные данные в технологиях ИАД имеют структурированное представление и являются цифровыми или символьными. Поскольку до 80 % всех данных существуют в неструктурированном виде (содержатся в текстовых документах), важность интегрированных средств, реализующих технологии ИАД и анализа текста, будет возрастать. Вопросы извлечения знаний из текста были рассмотрены в гл. 3.

Методы ИАД реализуются в технологиях:

- интерактивной аналитической обработки данных (On-Line Analytical Processing — OLAP);
- глубинного анализа данных (Data Mining — DM);
- визуализации данных.

7.2.2. Технология OLAP и многомерные модели данных

Технология OLAP ориентирована, главным образом, на обработку нерегламентированных запросов к хранилищам данных. Создание хранилищ данных вызвано тем, что анализировать данные OLTP-систем напрямую невозможно или затруднительно, так как они являются разрозненными, хранятся в форматах различных СУБД и в разных сегментах корпоративной сети. В целом можно сказать, что данные OLTP-систем не ориентированы на потребности аналитиков. Поэтому основной задачей хранилища является представление данных для анализа в одном месте в рамках простой и понятной структуры. На рис. 7.1 показаны компоненты, входящие в типичное

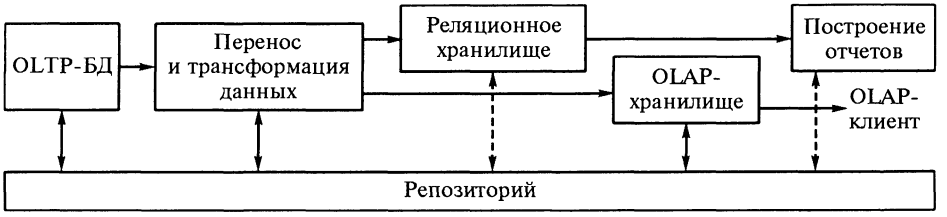


Рис. 7.1. Структура хранилища данных

хранилище данных [249]. Сплошные стрелки обозначают потоки данных, пунктирные – метаданных.

Основная цель анализа данных — качественная и количественная оценка достигнутых результатов и (или) динамики деятельности компании. Принципы OLAP, используемые для этого, были сформулированы Э. Коддом. Центральное место среди них занимает поддержка *многомерного представления данных.* В многомерной модели данных БД представляется в виде одного или нескольких *кубов данных (гиперкубов).*

Осями гиперкуба служат основные атрибуты анализируемого бизнес-процесса. Например, для бизнес-процесса «Продажи» такими атрибутами могут быть товар, регион, тип покупателя, время продажи и т. д. Независимые измерения гиперкуба представляют многомерное пространство данных. Каждому измерению соответствует атрибут, характеризующий одно из качественных свойств данных: время, территорию, категорию продукции и т. п. На пересечении осей-измерений (dimensions), т. е. в ячейке гиперкуба, содержатся данные, количественно характеризующие анализируемый процесс. Эти данные называются мерами (measures) или показателями. Для рассматриваемого примера в качестве показателей могут выступать объем продаж, остатки на складе, величина издержек и т. д.

В процессе анализа выполняются операции построения сечений (проекций) гиперкуба путем фиксации значений наборов атрибутов-координат, а также операции сжатия гиперкуба путем использования значений атрибутов-измерений более высоких уровней иерархии и соответствующего агрегирования значений ассоциируемых с ними показателей. Иерархические отношения могут быть естественным образом введены для ряда атрибутов. Например, для атрибута «время» иерархия имеет вид: годы — кварталы — месяцы, для атрибута «территория»: регионы — города — районы. Могут применяться и обратные операции детализации данных.

Заметим, что куб данных рассматривается как концептуальное, а не физическое представление. Для обеспечения удобства восприятия данных аналитиками используются операции вращения куба путем изменения порядка измерений. Для визуализации данных из гиперкуба, как пра-

а

II квартал
Виды транспортных средств

Регионы	трактора	грузовые автомобили	легковые автомобили
Урал	700	2350	7800
Поволжье	1150	4650	11500
Сибирь	920	3300	8900

б

Рис. 7.2. Гиперкуб (*а*) и одно из его сечений (*б*)

вило, применяются двумерные представления в виде таблиц, имеющих сложные иерархические заголовки строк и столбцов. Двумерное представление куба можно получить, фиксируя значения всех измерений, кроме двух. На рис. 7.2, *а* изображен гиперкуб, а на рис. 7.2, *б* — одно из его сечений.

Многомерность в OLAP-приложениях воплощается в рамках двух- или трехуровневой архитектуры. Первый уровень поддерживает многомерное представление данных, абстрагированное от их физической структуры. Он содержит средства многомерной визуализации и манипулирования данными для конечного пользователя. Второй уровень обеспечивает многомерную обработку. Он включает язык формулирования многомерных запросов (SQL для этих целей непригоден) и программный процессор, способный выполнять такие запросы. На третьем уровне архитектуры реализуется физическая организация хранения многомерных данных. В рамках него для поддержки многомерных моделей данных используются либо специальные OLAP-СУБД, либо обычные реляционные структуры.

В качестве примеров средств первого уровня можно назвать OLAP-клиенты Pivot Tables из Microsoft Excel 2000 и Pro Clarity фирмы Knosys. На

этом уровне могут применяться OLAP-сервера, например, Oracle Express Server и Microsoft OLAP Services.

Слой многомерной обработки обычно встраивается в OLAP-клиент или в OLAP-сервер, но может выделяться и в самостоятельном качестве (например, Pivot Tables Service фирмы Microsoft).

Слой физического хранения данных реализуется либо в реляционных, либо в многомерных структурах, представляемых в виде многомерных массивов. Обычно OLAP-продукты обеспечивают оба эти способа хранения, а также их комбинации:

MOLAP (Multidimensional OLAP) — и детальные данные, и агрегаты данных хранятся в многомерной БД;

ROLAP (Relational OLAP) — детальные данные хранятся в реляционной БД, агрегаты — в специально созданных служебных таблицах;

HOLAP (Hybrid OLAP) — детальные данные хранятся в реляционной БД, агрегаты — в многомерной БД.

В ROLAP используются две схемы хранения многомерных данных: звезда и снежинка. В базу данных входят таблица фактов и ряд таблиц измерений. Строка таблицы фактов представляет набор фактов, которые могут быть как атомарными, так и агрегированными, т. е. соответствующими совокупностям значений элементов измерений. Для каждой таблицы измерений ассоциируемая с ней строка таблицы фактов содержит значение внешнего ключа. В строках таблиц измерений указаны значения первичных ключей, в качестве которых выступают значения атрибутов для различных измерений.

При выполнении запросов используются операции соединения таблицы фактов и таблиц измерений. В схеме типа звезда таблицы измерений являются денормализованными и могут содержать составные первичные ключи. Это обеспечивает упрощение запросов и сокращение количества операций соединений таблиц при их выполнении, а также повышает наглядность представления данных. Расплатой за положительный эффект является избыточность данных, вызывающая рост требуемого объема памяти. Для минимизации избыточности используется схема типа снежинка. В ней таблицы измерений нормализованы за счет их декомпозиции.

В технологии хранилищ данных важную роль играет *управление метаданными* (см. рис. 7.1). Метаданные хранилищ делятся на административные, операционные и бизнес-метаданные. Административные метаданные отражают сведения, необходимые для инсталляции и эксплуатации хранилища. Они описывают OLTP-БД, служащие источниками для OLAP, схемы данных хранилища, измерения гиперкубов, физическую организацию данных, формы стандартных отчетов, полномочия пользователей, типовые запросы и др. Операционные метаданные фиксируются в процессе работы

хранилища. Они отражают информацию о текущем состоянии данных, статистике функционирования и т. д. К бизнес-метаданным относятся словарь терминов с их определениями, описания источников и владельцев данных, платежной политики и т. п.

7.2.3. Глубинный анализ данных

Технология DM предназначена для анализа структурированных данных с помощью математических моделей, основанных на статистических, вероятностных и оптимизационных методах, с целью выявления в них заранее неизвестных закономерностей, зависимостей и извлечения непредвиденной информации [244].

К числу основных *задач DM* относятся задачи классификации, кластеризации, поиска ассоциаций и корреляций, выявления типовых образцов на заданном множестве, обнаружения объектов данных, не соответствующих установленным характеристикам и поведению, исследование тенденций во временных рядах и др. Решение этих задач требует обработки больших объемов информации, содержащейся в хранилищах данных. Особенность используемых при этом алгоритмов состоит в том, что при их создании необходимо учитывать организацию источника данных, их значительный объем и большие размерности задач. Одно из важных требований к алгоритмам связано с обеспечением их масштабируемости.

В рамках DM для сегментирования данных применяются ИНС (см. гл. 6) и методы кластерного анализа, для индуктивного вывода — деревья принятия решений, для выявления в информационных массивах часто встречающихся пар объектов — статистические и ассоциативные методы.

Схема процесса ИАД на основе технологии DM изображена на рис. 7.3.

Процесс ИАД включает четыре основных этапа. На первом аналитик формулирует постановку задачи в терминах целевых переменных. На втором этапе осуществляется подготовка данных для анализа.

Обычно данные представляются в виде таблицы, строки которой (записи) соответствуют объектам или состояниям объекта, а столбцы (поля, переменные) — свойствам (признакам) объектов. Значения свойств могут выражаться числами, логическими величинами и категориальными (нечисловыми) данными.

Категориальная переменная может быть заменена набором логических переменных, количество которых равно количеству значений категориальной переменной. Между множеством значений категориальной переменной и множеством логических переменных устанавливается взаимно однозначное

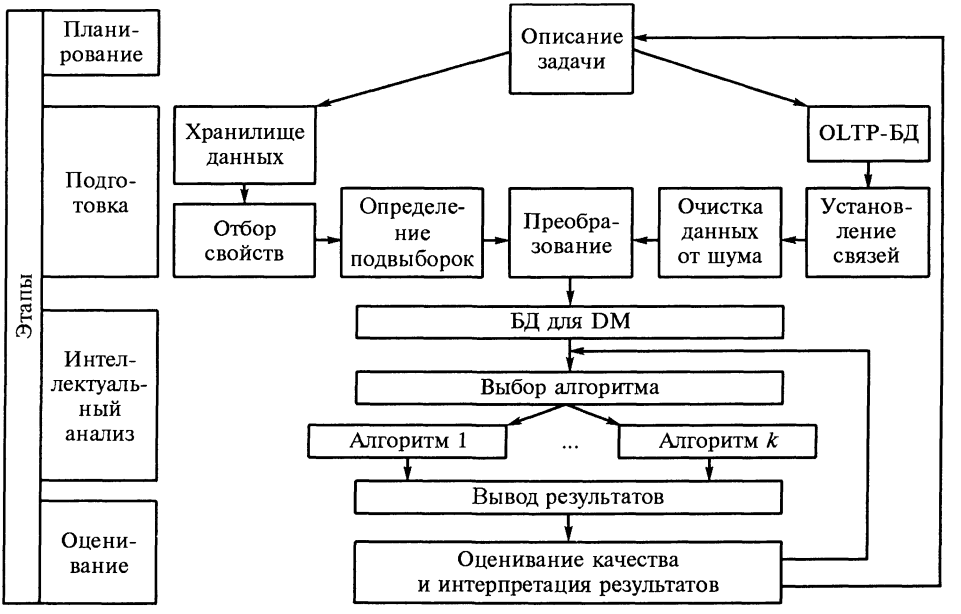


Рис. 7.3. Схема процесса ИАД на основе технологии DM

соответствие: значение «истина» (1) каждой логической переменной представляет соответствующее ей значение категориальной переменной. Например, категориальную переменную «Результат голосования», принимающую значения из множества {«за», «против», «воздержался», «не голосовал»}, можно заменить набором логических переменных z_1, z_2, z_3, z_4 (табл. 7.2).

Таблица 7.2

Значение категориальной переменной	Логические переменные			
	z_1	z_2	z_3	z_4
«за»	1	0	0	0
«против»	0	1	0	0
«воздержался»	0	0	1	0
«не голосовал»	0	0	0	1

Из множества свойств должны быть исключены избыточные и малоинформативные элементы. Малоинформативными являются свойства, имеющие одно и то же значение почти для всех записей, а также свойства, количество значений которых приближается к числу записей.

Некоторые методы DM требуют, чтобы все свойства всех объектов имели определенные значения. При невыполнении этого условия проводится

доопределение недостающих данных (например, путем присвоения им средних значений соответствующих переменных).

Еще одна задача этапа подготовки связана с выявлением и удалением записей, представляющих редкие особые ситуации либо содержащих ошибочные или очень неточные значения, т. е. способных оказать существенное негативное влияние на результаты анализа.

В рамках описываемого этапа также выполняется нормализация числовых данных. Часто для этого используется преобразование:

$$x_{ij} \to \frac{x_{ij} - \overline{x}_j}{\sigma_j}, \tag{7.1}$$

где x_{ij} — исходное значение j-го свойства i-го объекта; $\overline{x}_j$ — среднее значение j-го свойства; σ_j — дисперсия j-го свойства.

Собственно анализ данных с помощью методов DM проводится на третьем этапе.

Содержанием четвертого этапа является верификация и интерпретация полученных результатов (извлеченных знаний). При верификации применяется тестовый набор записей, выделенных из исходных данных и не подвергавшихся анализу.

В качестве примера назовем некоторые зарубежные продукты DM.

1. Intelligent Miner (разработчик — фирма IBM). Используются ИНС, методы предсказывающего моделирования, обнаружения ассоциаций, сегментации БД и др.

2. Decision Series (разработчик — Neo Vista Software). Используются ИНС, деревья и кластеры решений, ассоциативные правила.

3. Darwin, Loyalty Stream (разработчик — Thinking Machines). Используются ИНС и деревья решений.

В качестве примера российского продукта DM отметим систему Polyanalyst фирмы Megaputer*. Она позволяет выявлять многофакторные зависимости, которые представляются в виде функциональных выражений, а также формировать структурные и классификационные правила. В Polyanalyst используются метод группировки и поиска ближайшего соседа, генетические алгоритмы, ИНС, статистические и ассоциативные методы, деревья решений, регрессионные модели, методы кластерного анализа и эволюционного программирования.

Унификация и стандартизация технологий DM являются целями проекта CRISP-DM — Cross Industry Standard Process for Data Mining**. Его ре-

* http://www.megaputer.ru.

** http://www.crisp-dm.org.

зультаты реализуются в рамках CASE-системы для разработки средств DM. Она поддерживает все этапы, изображенные на рис. 7.3. Методика CRISP-DM обещает сделать DM более понятной и практичной технологией.

Основные выводы

1. Технологии хранилищ данных и ИАД вступили в пору зрелости и активного коммерческого использования.

2. Под ИАД понимаются процессы извлечения из данных ранее неизвестных нетривиальных практически полезных и доступных для интерпретации знаний.

3. Методы ИАД оперируют с данными, представленными в атрибутивном, структурном и полнотекстовом видах.

4. Выделяются три класса методов ИАД: алгебраические, статистические и методы мягких вычислений. В методах последнего класса используются нечеткое представление данных и нейросети.

5. Методы ИАД реализуются в технологиях OLAP, DM и визуализации данных.

6. При загрузке данных в хранилище должны выполняться операции их интеллектуальной верификации, предусматривающие проверку выполнения ограничений целостности, выявление и разрешение противоречий, устранение избыточности.

7. Технология OLAP базируется на многомерном представлении данных и реализуется в двух- или трехуровневой архитектуре. Для хранения данных используются или специальные OLAP-СУБД, или обычные реляционные СУБД.

8. В системах OLAP применяют MOLAP-, ROLAP- и HOLAP-способы организации хранения многомерных данных.

9. В технологии хранилищ данных важную роль играет управление административными, операционными и бизнес-метаданными.

10. В технологии глубинного анализа данных (DM) используются статистические, вероятностные и оптимизационные методы, а также ИНС для извлечения ранее неизвестных знаний из структурированных данных. С помощью DM решаются задачи классификации, кластеризации, поиска ассоциаций и корреляций, выявления типовых образцов на заданном множестве, обнаружения объектов данных, не соответствующих установленным характеристикам и поведению, исследования тенденций во временных рядах и др.

Вопросы для самопроверки

1. Что такое хранилище данных?
2. В чем состоят интеллектуальные свойства хранилищ данных?

3. Что понимается под ИАД?
4. На какие группы подразделяют ПрО с точки зрения возможностей ИАД?
5. Какие основные способы представления данных используются в методах ИАД?
6. На какие классы подразделяются методы ИАД?
7. В каких технологиях реализуются методы ИАД?
8. Что такое OLAP?
9. Какие основные компоненты входят в типичное хранилище данных?
10. Что такое многомерное представление или гиперкубы данных?
11. Как интерпретируются сечения гиперкуба данных?
12. Какие основные операции анализа и визуализации данных, представленных гиперкубом, используются в OLAP?
13. Что такое MOLAP, ROLAP и HOLAP?
14. Чем различаются схемы хранения многомерных данных типа звезда и снежинка?
15. Какие классы метаданных выделяются в технологии хранилищ данных?
16. Для чего предназначена технология глубинного анализа данных?
17. Какие задачи решаются с помощью технологии DM?
18. Охарактеризуйте основные этапы процесса глубинного анализа данных.
19. Какие модели и методы используются в рамках DM?

7.3. Системы поддержки инновационной деятельности

Каждый принимает конец своего кругозора за конец света.

А. Шопенгауэр

Системы данного класса имеют комплексный характер и реализуют множество функций управления знаниями, среди которых центральное место занимают функции, обеспечивающие методическую и информационную поддержку решения *типовых задач инновационной деятельности.* К таким задачам относятся:

- формирование научно-технической политики организации (планирование НИОКР, выбор стратегических партнеров, мониторинг технологических достижений конкурентов и др.);
- концептуальное проектирование технических систем (анализ и выявление проблем, присущих технической системе, разработка принципиальных способов их преодоления, поиск аналогов проектируемой технической системы, определение перспективных направлений совершенствования существующей технической системы и др.);
- поиск новых рынков для существующих продуктов;
- анализ новизны концепций технических решений;
- выявление новых технически реализуемых потребностей;

- анализ технологических тенденций и рынков;
- систематизация интеллектуальной собственности организации и др.

Рассмотрим подходы к реализации средств компьютерной поддержки инновационной деятельности на примере технологий, развиваемых фирмой Invention Machine Corp*. (Бостон, США). В основе этих технологий лежат три класса методов:

1) обработка текстов на ЕЯ (извлечение знаний из текста и построение проблемных БЗ, семантический поиск в тексте, автоматическое реферирование и аннотирование, автоматическая классификация документов);

2) моделирование и анализ функциональных структур технических объектов и технологических процессов;

3) методология концептуального проектирования технологий и техники, базирующаяся на теории решения изобретательских задач (ТРИЗ) [253, 254].

Использование технологий Invention Machine Corp. позволяет приобрести или повысить конкурентные преимущества за счет:

- существенного увеличения эффективности и результативности НИОКР, снижения времени и затрат на их выполнение;
- систематизации знаний и интеллектуальной собственности организации;
- значительного сокращения времени, затрачиваемого специалистами на поиск и чтение информационных источников (технической литературы, статей, отчетов, описаний объектов промышленной собственности и т. п.);
- обладания оперативной систематизированной информацией о технологических тенденциях, состоянии рынков и технологических достижениях конкурентов.

Продукты Invention Machine Corp. представлены в табл. 7.3.

Таблица 7.3

Старая линия продуктов (до 2002 г.)	Новая линия продуктов (с 2002 г.)
Knowledgist («Знайка»)	Goldfire Research
CoBrain («Корпоративный мозг»)	Goldfire Intelligence
TechOptimizer	Goldfire Innovation

Продукты старой линии в настоящее время не продаются, но фирма продолжает их поддерживать (в том числе предоставлять владельцам лицензий на право их использования обновленные версии).

Новая линии продуктов получила название *платформа Goldfire.* Она представляет собой масштабируемое комплексное корпоративное решение для поддержки инновационной деятельности. Каждый из продуктов плат-

* http://www.invention-machine.com.

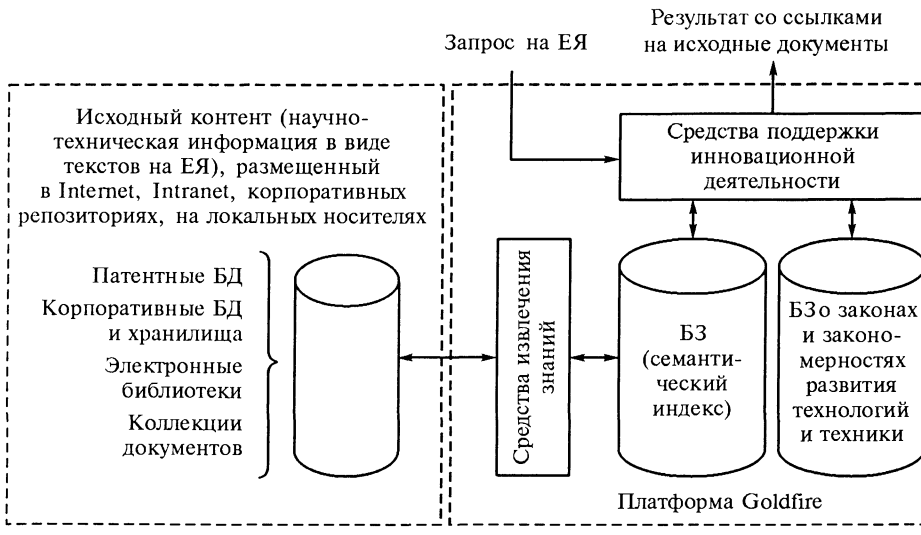


Рис. 7.4. Платформа Goldfire

формы Goldfire базируется на продукте старой линии, указанном с ним в одной строке табл. 7.3.

Обобщенная схема функционирования продуктов платформы Goldfire изображена на рис. 7.4.

Системы Goldfire Research и Knowledgist рассчитаны на широкий круг пользователей — инженеров, ученых, менеджеров, т. е. всех, кто испытывает потребности в извлечении требуемых сведений из объемного массива текстовых данных, явном представлении и систематизации знаний. Исходные электронные документы могут храниться в Internet, Intranet, корпоративных репозиториях и на локальных носителях. Goldfire Research поддерживает более 100 входных форматов, в том числе HTML, XML, DOC, RTF, PDF, TXT, XLS, PPT и др. Информация может извлекаться из репозиториев Lotus Notes, сообщений электронной почты и БД, совместимых с ODBC.

Система Goldfire Research позволяет обрабатывать контент глубинного web, размещенный на более чем 2000 сайтов правительственных, академических, исследовательских и коммерческих организаций США, представляющих 26 отраслей. Как известно, для традиционных поисковых машин глубинный web недоступен. Goldfire Research обладает информацией о механизмах доступа к БД глубинного web и автоматически генерирует запросы к ним.

Базовой моделью для анализа текста и представления знаний в БЗ служит *тройка Subject — Action — Object (SAO)*. Обработка текста выполняется по предложениям на морфологическом и синтаксическом уровнях. Первый член SAO (*S*) соответствует субъекту, второй (*A*) — действию, вы-

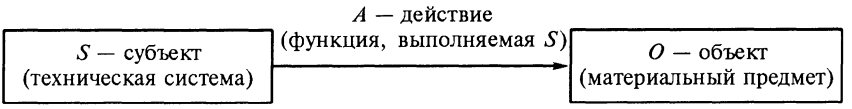


Рис. 7.5. Модель SAO

полняемому субъектом, третий (*O*) — объекту, на который направлено это действие. В предложении *S* выражает подлежащее, *A* — сказуемое, *O* — дополнение. Заметим, что в технических документах в качестве субъекта обычно выступает название (марка, идентификатор) технической системы (устройства, способа, изделия, технологии), действие представляет функцию данной системы, а объект фиксирует материальный предмет, на который воздействует *S* (рис. 7.5). Он может обозначать предмет труда, средство для обеспечения функционирования *S* (сырье, энергию и т. д.), другую техническую систему, объект живой и неживой природы, человека, надсистему *S*, часть *S* или даже *S* целиком.

При обработке текста применяется развитая система словарей, включающая словари синонимов, настраиваемые пользователем. Пример анализа текста показан на рис. 7.6.

Формируемая корпоративная БЗ, называемая *семантическим индексом,* содержит совокупность экземпляров SAO, снабженных атрибутами документов-источников и ссылками на них. Система Goldfire Research позволяет включать в БЗ *автоматически составляемые аннотации* исходных документов. В дальнейшем пользователь работает с этой БЗ.

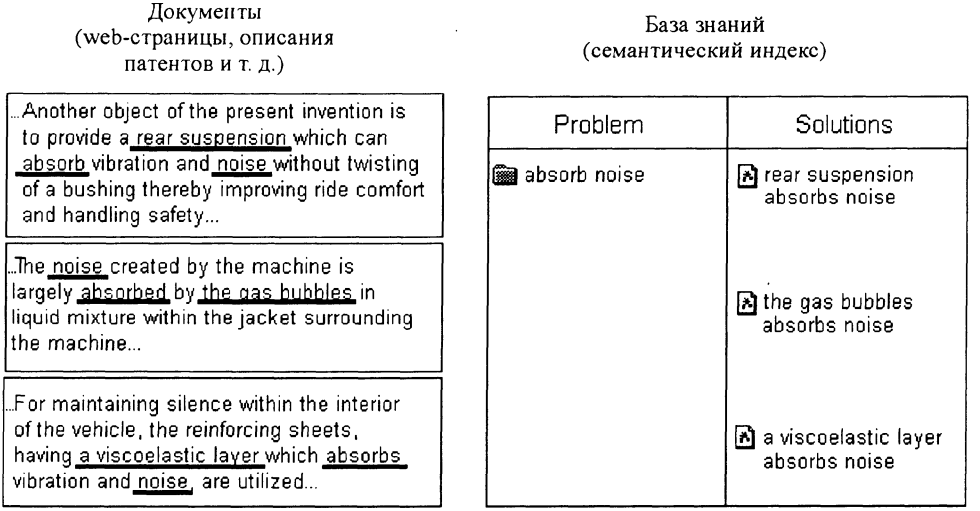


Рис. 7.6. Пример анализа текста

Решение задач инновационной деятельности на основе семантического индекса выполняется по следующей схеме:

1) составляется запрос, содержащий значения одного или двух компонентов SAO, а также дополнительные условия для выделения фрагмента БЗ (как правило, эти условия связаны с атрибутами документов-источников);

2) система осуществляет выборку из БЗ фрагмента, удовлетворяющего запросу;

3) выборка сортируется в требуемом порядке (обычно по убыванию частоты упоминания в документах заданных значений компонентов SAO);

4) пользователь анализирует выборку, сопоставляет несколько выборок, построенных для разных подмножеств документов.

Описанная схема исходит из того, что экземпляр SAO, фигурирующий в запросе на первом этапе, отражает типовую постановку задачи. Интерпретации каждой из шести возможных комбинаций фиксированных и переменных значений компонентов SAO приведены в табл. 7.4.

Таблица 7.4

Элемент модели			Интерпретация запроса	Примеры
S	*A*	*O*		
?	o	o	Какой субъект выполняет требуемое воздействие на данный объект? Найти устройства (технологии), обеспечивающие требуемое преобразование имеющегося объекта	1. Найти вещества, уничтожающие данный вид микробов. 2. Найти технологии дезагрегации данного материала
o	?	o	Каким образом данный субъект может воздействовать на данный объект? Оценить потенциальные воздействия имеющегося субъекта на данный объект либо определить роли, в которых может выступать имеющийся объект по отношению к данному субъекту	1. Оценить воздействие мобильного телефона на здоровье человека. 2. Определить, как можно использовать отходы производства в рамках данной технологии
?	?	o	Какие субъекты и каким образом могут воздействовать на данный объект? Определить возможные воздействия на данный объект либо для имеющегося объекта (продукта, вещественного или энергетического ресурса) найти новое применение (назначение, сферу сбыта)	1. Оценить, какие факторы внешней среды и каким образом могут воздействовать на рукописи в хранилище. 2. Найти области, в которых используется данный вид сырья

Окончание табл. 7.4

Элемент модели			Интерпретация запроса	Примеры
S	*A*	*O*		
o	o	?	На какой объект может заданным образом воздействовать имеющийся субъект? Найти объекты, с которыми указанным образом может взаимодействовать имеющаяся система	Определить виды микробов, уничтожаемых данным веществом
?	o	?	Какие субъекты реализуют данную функцию? Найти технологии (устройства, способы), обеспечивающие выполнение данной функции	Найти системы, обеспечивающие охлаждение чего-либо
o	?	?	Какие действия над какими объектами может выполнять данный субъект? Для имеющейся системы (устройства, технологии, продукта) найти новое применение (назначение, сферу сбыта)	Определить, как можно использовать износившиеся автомобильные покрышки

Примечание. «?» — переменное значение; «o» — фиксированное значение.

На рис. 7.7 приведена иллюстрация пользовательского интерфейса системы Knowledgist, демонстрирующая работу с БЗ. В левой части окна располагается фрагмент БЗ, содержащий выборку экземпляров SAO, сгруппированных и упорядоченных заданным образом. В правой части отображаются перечень экземпляров SAO, входящих в выделенную группу, графическое представление текущего экземпляра и ссылки на документы-источники. Кнопки «Action», «Subject» и «Object» в левой верхней зоне окна позволяют определять различные варианты группирования компонентов SAO и упорядочения фрагмента БЗ.

В системе Goldfire Research предусмотрена возможность ввода запроса в виде предложения ЕЯ. Такой запрос обрабатывается аналогично предложениям исходных документов, в результате чего он переводится в один или несколько экземпляров SAO, которые при поиске в БЗ сопоставляются с содержащимися в ней экземплярами SAO. Найденные экземпляры SAO дополняют экземпляры SAO из запроса, что дает основание говорить о том, что результаты поиска не только предоставляют информацию, которая может помочь в решении проблемы, стоящей перед пользователем, но и фактически отвечают на поставленный им вопрос.

Поскольку каждый экземпляр SAO в БЗ ссылается на документ, из которого он был извлечен, механизмы формирования БЗ, а также группирова-

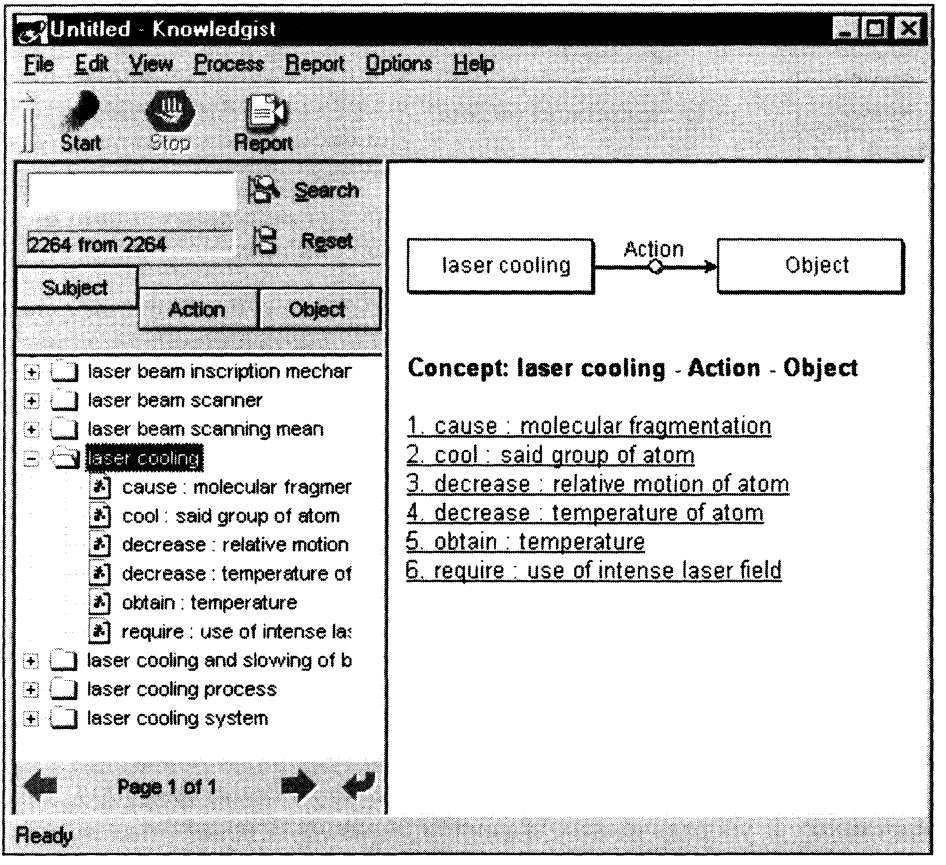


Рис. 7.7. Работа с БЗ в системе Knowledgist

ния и упорядочения экземпляров SAO в ней обеспечивают *автоматическую кластеризацию документов.*

Система Goldfire Research позволяет определить план автоматического пополнения и обновления БЗ. Средства отложенного поиска анализируют вновь выявленные знания и информируют пользователя о поступлении в БЗ интересующих его сведений.

Алгоритмы Goldfire Research и Knowledgist инвариантны к ПрО. Единственный вид настройки на нее связан с формированием и корректировкой словарей.

Система Goldfire Research реализована в рамках архитектуры клиент—сервер. Рекомендуемая конфигурация включает два сервера: сервер приложений (application server) и сервер формирования и обновления БЗ (indexing server).

Система Goldfire Intelligence базируется на Goldfire Research, расширяя ее функциональность за счет средств, обеспечивающих:

- доступ к БД патентных агентств промышленно развитых стран;
- анализ патентной библиографии с целью выявления технологических тенденций и построения профилей компаний, изобретателей и технологий;
- автоматическое реферирование описаний патентов;
- навигацию по сети, образованной гиперссылками, связывающими описания патентов;
- поиск альтернативных способов реализации функций разрабатываемых или совершенствуемых технических систем на основе БД, содержащей более 8000 анимированных описаний естественнонаучных и научно-технических эффектов, представляющих законы и закономерности природы и типовые принципы действия технологий и техники.

Система Goldfire Intelligence ориентирована на следующие основные категории пользователей:

- руководителей, определяющих научно-техническую политику организации;
- специалистов, проводящих маркетинговые исследования;
- менеджеров, занимающихся планированием НИОКР;
- ученых, изобретателей и проектировщиков технических систем;
- патентоведов, специалистов по управлению интеллектуальной собственностью и др.

База знаний, представляющая патентные источники, хранится на сайте Invention Machine Corp. Доступ к ней открыт только для владельцев лицензий на право использования продуктов фирмы. Обновление и пополнение этой БЗ осуществляется еженедельно.

Корпоративные и персональные БЗ, формируемые Goldfire Intelligence, размещаются на пользовательских компьютерах. Система выполняет поиск во всех БЗ, доступных данному пользователю, и предоставляет единый интерфейс для формулирования запросов и оперирования выделенными экземплярами SAO вне зависимости от места хранения БЗ.

Существует вариант работы с Goldfire Intelligence, не требующий установки ее серверных компонентов в рамках вычислительной среды организации-пользователя. Их функции выполняют серверные компоненты, поддерживаемые сайтом Invention Machine Corp.

Предшественником Goldfire Intelligence является *система CoBrain.* Ее первые версии были размещены в Internet в открытом доступе, что способствовало популяризации технологий и продуктов Invention Machine Corp.

Еще одна система поддержки инновационной деятельности — *Goldfire Innovation* — представляет собой интеграцию Goldfire Intelligence с па-

кетом TechOptimizer — первым продуктом Invention Machine Corp. *TechOptimizer* реализован на основе системы IM Lab, разработанной в конце 80-х годов в Минске под руководством В.М. Цурикова. Его методологическим и информационным фундаментом служит ТРИЗ, созданная Г.С. Альтшуллером и развитая его учениками и последователями.

Главные функции TechOptimizer — поиск, синтез и стимулирование формирования новых решений для задач *концептуального проектирования технологий и техники* [252, 255, 257]. Концептуальными называют начальные стадии проектирования, на которых определяется облик создаваемого объекта: функции, потребительские свойства, основные качественные характеристики и параметры. Задачи концептуального проектирования обладают высокой ценой ошибки и имеют ярко выраженный творческий характер, обусловливающий сложность их формализации.

Пакет TechOptimizer состоит из семи модулей:

1) анализа продукта;
2) анализа процесса;
3) переноса свойств;
4) эффектов;
5) принципов;
6) предсказаний;
7) Internet-помощника для поиска и анализа патентной информации.

Первые два модуля реализуют методы моделирования и анализа функциональных структур технических объектов и технологических процессов. Функциональная модель продукта (устройства) или процесса (технологии) строится с помощью специального графического редактора (рис. 7.8). Далее система выполняет анализ модели и предлагает варианты ее совершенствования за счет исключения вредных и бесполезных компонентов и перераспределения функций между оставшимися компонентами. Данная процедура называется *тримингом* (trimming — выравнивание, балансировка). Цель тримминга состоит в упрощении продукта или процесса путем перераспределения функций и исключения компонентов с сохранением полезных функций. Это удешевляет совершенствуемую систему и сокращает количество возможных проблем, связанных с ее созданием и эксплуатацией. Эффект тримминга основан на том, что:

- при удалении компонента исключаются все его вредные функции;
- в результате выравнивания функциональной нагрузки на систему ее структура упрощается, становясь более однородной;
- если одна или несколько функций одного компонента передаются другому компоненту, то первый компонент становится проще и дешевле.

Компонент исключается из функциональной структуры в четырех случаях:

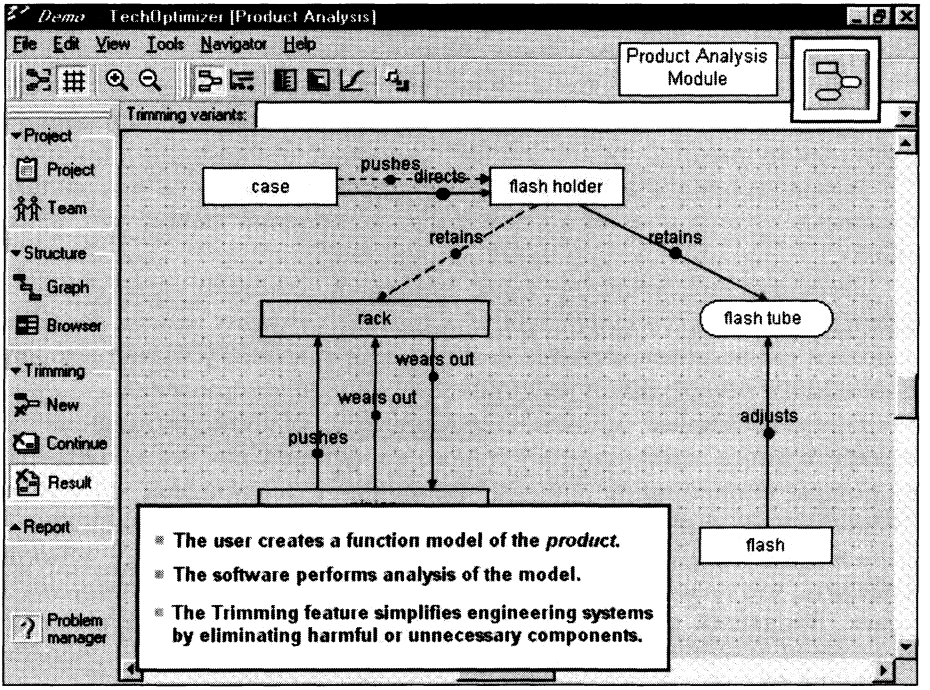


Рис. 7.8. Модуль анализа продукта

1) его функция передана другому компоненту или надсистеме;

2) его функция передана объекту, на который он воздействует (т. е. объект способен требуемым образом воздействовать сам на себя);

3) исключен компонент, на который воздействует данный компонент;

4) исключена функция данного компонента.

Для выбора компонентов, подвергаемых триммингу, используется следующий критерий:

$$C_t = F\frac{F}{P+C}, \tag{7.2}$$

где F — ранг функции, отражающий ее важность; P — ранг проблем, связанных с выполнением функции; C — стоимость (затраты на выполнение функции).

Как видно из (7.2) на значение C_t влияют две составляющие: $\frac{F}{P+C}$ и F. Первая отражает эффективность компонента: чем выше ее значение, тем лучше компонент. Вторая характеризует сложность тримминга: чем важнее функция, тем сложнее выполнить тримминг реализующего ее компонента.

Очевидно, что компоненты с минимальным значением C_i должны подвергаться триммингу в первую очередь.

Ранги функций и проблем вычисляются по формальным признакам и параметрам, представленным в модели. Значения стоимостей определяются экспертным путем.

В модулях анализа продукта и процесса используются разные функциональные модели и методы формирования значений параметров, влияющих на C_i.

Работая с TechOptimizer, можно построить несколько «траекторий» совершенствования продукта или процесса, соответствующих разным последовательностям компонентов, которые подвергаются триммингу. Пакет фиксирует все варианты и предоставляет наглядные средства для их сравнения.

Рассматриваемые модули TechOptimizer обеспечивают поддержку предварительного этапа концептуального проектирования, заключающегося в *выявлении и постановке проблем,* связанных с реализацией улучшенной функциональной структуры продукта или процесса, которая была сформирована в результате тримминга. Иными словами, данные модули помогают определить, что целесообразно изменить, но не отвечают на вопрос, как это сделать. Входящий в них менеджер проблем автоматически фиксирует и упорядочивает проблемы, ассоциируемые с разными вариантами тримминга. Для поддержки их решения служат другие модули TechOptimizer.

Модуль переноса свойств позволяет проанализировать множество объектов, являющихся функциональными аналогами проектируемого объекта, выбрать среди них наиболее перспективный объект, который может служить прототипом, и выделить свойства других объектов, реализация которых в прототипе обеспечивает наилучший результат.

Все объекты описываются общим набором параметров. Параметр задает пятерка: название, единица измерения, значение, степень важности и направление желаемого изменения (увеличение, уменьшение). Система на основе интегрального критерия выбирает наиболее перспективный аналог (прототип) и помогает выделить лучшие свойства других объектов для воплощения в нем. Поиск способов реализации свойств определяет содержание задач, решаемых на других этапах концептуального проектирования.

Модуль принципов предназначен для поддержки формулирования и разрешения технических противоречий. *Техническим противоречием* называется ситуация, при которой улучшение одной характеристики объекта вызывает ухудшение другой.

Технические противоречия в ТРИЗ специфицируются парами конфликтующих характеристик. На основе анализа большого числа изобретений были выделены и обобщены типовые способы разрешения противоречий, называемые стандартами. Совокупность подобных стандартов ТРИЗ

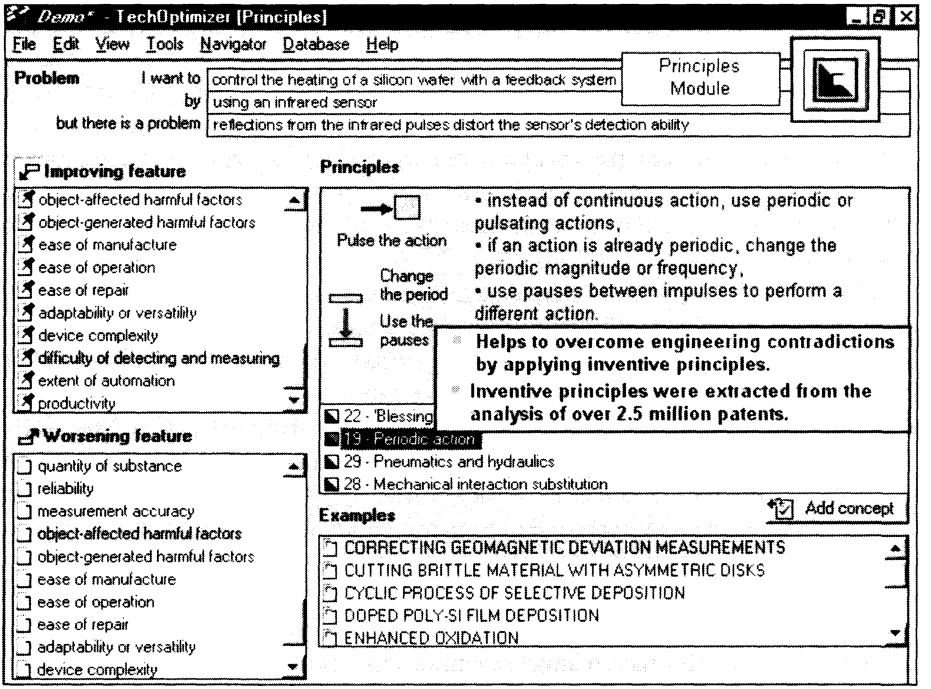


Рис. 7.9. Модуль принципов

сформулирована применительно к 39 наиболее важным техническим характеристикам. Для выбора стандартов служит квадратная матрица, в ячейках которой указаны их номера, а строки и столбцы представляют конфликтующие характеристики.

Описанная методика реализована в модуле принципов (рис. 7.9). Пользователь формулирует техническое противоречие, выбирая пару конфликтующих характеристик. В ответ система предлагает соответствующий набор стандартов. Все стандарты снабжены примерами их использования со ссылками на патенты.

Модуль эффектов включает базу функций технических систем и гипертекстовую базу естественнонаучных и научно-технических эффектов. Связи между функциями и эффектами отражают возможности реализации функций на основе эффектов.

Информация, с которой манипулирует пользователь, представлена на экране в виде иерархии: обобщенная функция (группа функций) — функция — эффект. Модуль позволяет:

- искать и выбирать эффекты, реализующие данную функцию;
- просматривать описания эффектов;

• описывать эффекты (формировать базу «пользовательских» эффектов);

• оценивать возможность установления связей между эффектами по входам-выходам.

Описание эффекта раскрывает его сущность, варианты технического воплощения и сопутствующие проблемы, примеры использования в технике и др. Описания связаны гиперссылками, включают графические иллюстрации и анимации.

Доступ к базе эффектов также обеспечивает система Goldfire Intelligence.

Модуль предсказаний предоставляет советы по решению технических проблем, основанные на эвристических приемах, которые отражают законы и закономерности развития технологий и техники. Приемы сгруппированы в две линии: Transformation Line и Measurement Line. Первая ориентирована на проблемы, связанные со взаимодействием нескольких материальных объектов, в ходе которого эти объекты изменяются или уничтожаются. Вторая линия рассчитана на задачи, требующие нахождения новых способов измерения свойств материального объекта. Приемы снабжены наглядными схемами, а также примерами использования, включающими графические иллюстрации.

Модуль позволяет оценить соответствие текущего состояния рассматриваемой системы законам и закономерностям технической эволюции, а также определить перспективные направления ее развития.

Internet-помощник предоставляет средства контекстного поиска в Internet (в том числе в патентных БД) и анализа патентной библиографии с целью выявления технологических тенденций. Система передает запросы наиболее мощным поисковым машинам Internet, принимает выдаваемые ими результаты и обрабатывает их.

Основные выводы

1. Системы поддержки инновационной деятельности имеют комплексный характер и реализуют множество функций управления знаниями. Их главное назначение — обеспечение методической и информационной поддержки решения типовых задач инновационной деятельности.

2. Один из наиболее успешных проектов в этой сфере — технологии, развиваемые фирмой Invention Machine Corp. В них используются три класса методов: обработка текстов на ЕЯ, моделирование и анализ функциональных структур технических систем, методология концептуального проектирования технологий и техники.

3. В продуктах платформы Goldfire реализованы функции извлечения знаний из текстовых документов, размещенных в Internet и корпоративных хранилищах, и формирования предметно-ориентированной БЗ — семантического индекса. Методы анализа текста и представления знаний базируют-

ся на модели SAO. Обработка текста выполняется на морфологическом и синтаксическом уровнях.

4. Семантический индекс является основой для процедур поиска, автоматической кластеризации и аннотирования документов. Система Goldfire Research обеспечивает выполнение запросов, представленных в виде предложений английского языка. Используемые алгоритмы обработки текста, построения и манипулирования семантическим индексом инварианты к ПрО. Единственный вид настройки на нее связан с формированием и корректировкой словарей.

5. Методы моделирования и анализа функциональных структур технических систем применяются в модулях анализа продукта и анализа процесса пакета TechOptimizer. Для совершенствования технической системы на функциональном уровне служит процедура тримминга. Ее суть состоит в упрощении продукта или процесса путем исключения вредных и бесполезных компонентов и перераспределения функций между оставшимися компонентами.

6. Пакет TechOptimizer предназначен для поиска, синтеза и стимулирования формирования новых решений для задач концептуального проектирования технологий и техники. Его методологическую основу составляют идеи ТРИЗ, реализованные в модулях переноса свойств, эффектов, принципов и предсказаний.

Вопросы для самопроверки

1. Назовите типовые задачи инновационной деятельности.
2. Какие классы методов используются в технологиях поддержки инновационной деятельности, развиваемых Invention Machine Corp.?
3. Охарактеризуйте задачи концептуального проектирования технологий и техники.
4. Опишите общую схему функционирования продуктов платформы Goldfire.
5. Для чего предназначена модель SAO?
6. Что такое семантический индекс, как он организован?
7. Какие типы запросов реализуются на основе модели SAO?
8. Каким образом в Goldfire организована автоматическая кластеризация документов?
9. Каковы функции системы Goldfire Intelligence? На какие категории пользователей она рассчитана?
10. Для чего предназначен пакет TechOptimizer? Что является его методологической основой?
11. Каково назначение основных модулей пакета TechOptimizer?
12. Какой этап концептуального проектирования поддерживают модули анализа продукта и процесса?
13. Для чего предназначена процедура тримминга?
14. Какие модели используются при тримминге?
15. С помощью какого критерия выбираются компоненты, подвергаемые тримингу?
16. Что такое техническое противоречие? Приведите примеры.
17. Охарактеризуйте идеи, лежащие в основе модулей принципов и эффектов.

Нам говорят «безумец» и «фантаст»,
Но, выйдя из зависимости грустной,
С годами мозг мыслителя искусный
Мыслителя искусственно создаст.

И.В. Гете, «Фауст»

ЗАКЛЮЧЕНИЕ

Хочется выразить надежду, что после изучения пособия у читателя сложились убеждения в полезности, а часто и необходимости применения интеллектуальных информационных технологий, сформировались представления об их возможностях, принципах построения и направлениях развития.

Эволюция интеллектуальных информационных технологий протекает стремительно. Некоторые из них (например, ГТ, ЭС, распознавание образов, МП и др.) уже сегодня являются привычными компонентами информационно-компьютерной среды, служащей человеку. Другие, кажущиеся пока экзотическими и далекими от практического применения, завтра прочно войдут в нашу жизнь.

Интеллектуальные информационные технологии базируются на результатах исследований, относящихся ко множеству областей науки. В первую очередь следует назвать дискретную математику, математическую логику, кибернетику, математическую лингвистику, ИИ, психологию, системотехнику и др. Кроме того, в них используются все последние достижения технологий программирования, Internet, многоагентных систем и т. д.

В качестве примера взаимного проникновения интеллектуальных информационных технологий и современных технологий программирования можно указать платформу Microsoft.NET, в создание которой корпорация вложила свыше 2 млрд. долл. и над развитием которой работают более 5 тыс. специалистов. Новые возможности она приобретает за счет воплощения идей и методов ИИ.

Так, технологии Natural Interface, отвечающие за «естественное» общение с компьютером, обеспечивают рукописный ввод, распознавание и синтез речи, а также создают условия для удобного использования самых разнообразных внешних устройств.

Технология Universal Canvas представляет основанную на XML-схемах архитектуру, превращающую Internet из платформы для «чтения» в платформу для «чтения/записи», поддерживающую такие операции с доку-

ментами, как совместное редактирование, объединение источников, комментирование, аннотирование и анализ.

Технология Information Agent позволяет ввести в Internet персонального представителя пользователя, знающего его потребности, предпочтения и привычки и позволяющего Internet-сервисам адаптироваться к ним.

Технология Smarttags является развитием технологии Intellisense. Она позволяет предугадывать и корректировать действия пользователя с помощью программ-помощников, работающих в Microsoft Office, Internet Explorer и других приложениях, а также функций автозамены, автоформатирования, автозаполнения и т. п.

Базовыми парадигмами ИИ остаются опора на знания, диалог на ЕЯ, понимание текста, логический вывод, обоснование и объяснение решений.

Ограниченный объем учебного пособия не дает возможности рассмотреть все направления, приведенные в классификации на с. 15–16. Для разработчиков прикладных ИАС наибольший интерес представляют направления четвертой группы. Информацию по ним можно получить из литературы, указанной в библиографическом списке. Технология ЭС описана в [2, 33, 124, 139]. Проблематика интеллектуальных АСУ рассмотрена в [7, 30, 36, 140, 249]. Направления интеллектуализации САПР очерчены в [122, 255, 256]. Методы, обеспечивающие интеллектуальные свойства АСНИ, изложены в [85, 130]. Подходы к созданию интеллектуальных компьютерных средств обучения представлены в [31, 132, 172]. Модели, лежащие в основе интеллектуальных роботов, описаны в [7]. Вопросы разработки и использования интеллектуальных консультирующих систем освещены в [53, 124]. Методы построения интеллектуальных интерфейсов и систем виртуальной реальности рассмотрены в [2, 29, 50, 51, 57, 63, 127].

Интеллектуальные информационные технологии становятся одной из наиболее перспективных областей для инвестиций. По данным зарубежных источников, 1 долл., вложенный в эту сферу, приносит до 50 долл. прибыли при сроках окупаемости проектов в 2–3 года. Прибыль от применения интеллектуальных информационных технологий в мире в 2003 г. превысила 5 млрд. долл.

Наука и техника XXI в. не мыслятся без широкого использования интеллектуальных информационных технологий. Благодаря созданию условий для раскрытия возможностей естественного интеллекта, они окажут глубокое влияние на все стороны жизни человека, сделают ее более комфортной и интересной.

СПИСОК ОСНОВНОЙ ЛИТЕРАТУРЫ

1. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. — СПб.: Питер, 2000. — 384 с.

2. Искусственный интеллект: В 3 кн. Кн. 1. Системы общения и экспертные системы: Справочник / Под ред. Э.В. Попова. — М.: Радио и связь, 1990. — 464 с.

3. Искусственный интеллект: В 3 кн. Кн. 2. Модели и методы: Справочник / Под ред. Д.А. Поспелова. — М.: Радио и связь, 1990. — 304 с.

4. Искусственный интеллект: В 3 кн. Кн. 3. Программные и аппаратные средства: Справочник / Под ред. В.Н. Захарова, В.Ф. Хорошевского. — М.: Радио и связь, 1990. — 368 с.

5. *Люггер Дж.Ф.* Искусственный интеллект. Стратегии и методы решения сложных проблем: Пер. с англ. — 4-е изд. — М.: Издательский дом «Вильямс», 2003. — 864 с.

6. *Нильсон Н.* Принципы искусственного интеллекта: Пер. с англ. — М.: Радио и связь, 1985. — 376 с.

7. *Поспелов Д.А.* Логико-лингвистические модели в системах управления. — М.: Энергоатомиздат, 1981. — 232 с.

8. *Поспелов Д.А.* Моделирование рассуждений. Опыт анализа мыслительных актов. — М.: Радио и связь, 1989. — 184 с.

9. Представление и использование знаний: Пер. с япон. / Под ред. Х. Уэно, М. Исидзука. — М.: Мир, 1989. — 220 с.

10. Толковый словарь по искусственному интеллекту / Авторы-составители А.Н. Аверкин, М.Г. Гаазе-Рапопорт, Д.А. Поспелов. — М.: Радио и связь, 1992. — 256 с.

11. *Тыугу Э.Х.* Концептуальное программирование. — М.: Наука, 1984. — 256 с.

СПИСОК ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ

Введение

12. *Гаазе-Рапопорт М.Г., Поспелов Д.А.* Структура исследований в области искусственного интеллекта // Толковый словарь по искусственному интеллекту. — М.: Радио и связь, 1992. — С. 5—20.

13. *Заморин А.П.* Этапы интеллектуализации ЭВМ общего назначения // Сб. «Электронная вычислительная техника». Вып. 1. — М.: Радио и связь, 1987. — С. 17—23.

14. *Поспелов Д.А.* Послесловие // Разговор с компьютером: Психолингвистический аспект проблемы / И.Н. Горелов. — М.: Наука, 1987. — С. 230—250.

15. *Уинстон П.* Искусственный интеллект: Пер. с англ. — М.: Мир, 1980. — 519 с.

§ 1.1

16. Анализ и моделирование производственных систем / Б.Г. Тамм, М.Э. Пуусепп, Р.Р. Таваст; Под общ. ред. Б.Г. Тамма. — М.: Финансы и статистика, 1987. — 191 с.

17. *Минц Г.Е.* О Е-теоремах // Записки научных семинаров ЛОМИ АН СССР. 1974. — Т. 40. — С. 110—118.

18. *Непейвода Н.Н.* Построение правильных программ // Вопросы кибернетики. 1978. — Т. 46, вып. 100—103. — С. 88—122.

19. *Непейвода Н.Н.* Соотношение между правилами естественного вывода и операторами алгоритмических языков // ДАН СССР. 1978. — Т. 239, № 4. — С. 526—529.

20. *Цейтин Г.С.* О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ АН СССР. 1968. — Т. 8. — С. 234—259.

21. *Feder J.* Plex Languages // Information Sciences. 1971. — № 3. — P. 225—241.

§ 1.2

22. Анализ и моделирование производственных систем / Б.Г. Тамм, М.Э. Пуусепп, Р.Р. Таваст; Под общ. ред. Б.Г. Тамма. — М.: Финансы и статистика, 1987. — 191 с.

23. Инструментальная система программирования ЕС ЭВМ (ПРИЗ) / М.И. Кахро, А.П. Калья, Э.Х. Тыугу. — М.: Финансы и статистика, 1981. — 158 с.

§ 2.1

24. *Абраменко А.* Принципы распознавания // Компьютер-пресс. 1997. — № 12.

25. *Загоруйко Н.Г.* Методы распознавания и их применение. — М.: Сов. радио, 1972. — 208 с.

26. *Линдсней П., Норман Д.* Переработка информации у человека: Пер. с англ. — М.: Мир, 1974. — 550 с.

27. *Шамис А.Л.* Принципы интеллектуализации автоматического распознавания изображений и их реализация в системах оптического распознавания символов // Новости искусственного интеллекта. 2000. — № 1. — С. 27—30.

§ 2.2

28. Cognitive Forms — система массового ввода структурированных документов [Электронный ресурс] / В.В. Арлазаров, В.В. Постников, Д.Л. Шоломов. — Электрон. текстовые дан. (195072 байт). — М.: Институт системного анализа РАН : Cognitive Technologies, 2002. — Режим доступа: ftp://ftp.dol.ru/pub/users/cgntv/download/sbornic/sbornic3/POSTNIK.DOC. — Содерж.: опубликовано в сборнике трудов Института системного анализа РАН «Управление информационными потоками».

§ 3.1

29. *Агеев В.Н., Узилевский Г.Я.* Человеко-компьютерное взаимодействие: концепции, процессы, модели. — М.: Мир книги, 1995. — 352 с.

30. Системный анализ в управлении: Учеб. пособие / В.С. Анфилатов, А.А. Емельянов, А.А. Кукушкин; Под ред. А.А. Емельянова. — М.: Финансы и статистика, 2002. — 386 с.

31. *Башмаков А.И., Башмаков И.А.* Разработка компьютерных учебников и обучающих систем. — М.: Информационно-издательский дом «Филинъ», 2003. — 616 с.

32. *Вуль В.А.* Электронные издания. — СПб.: БХВ-Петербург, 2003. — 560 с.

33. *Гаврилова Т.А., Червинская К.Р.* Извлечение и структурирование знаний для экспертных систем. — М.: Радио и связь, 1992. — 200 с.

34. *Гасов В.М., Цыганенко А.М.* Методы и средства подготовки электронных изданий: Учеб. пособие. — М.: Моск. гос. ун-т печати, 2001. — 735 с.

35. *Иванов В.В.* Чет и нечет: асимметрия мозга и знаковых систем. — М.: Сов. радио, 1978. — 187 с.

36. *Клименко С.В., Крохин И.В., Кущ В.М., Лагутин Ю.Л.* Электронные документы в корпоративных сетях: второе пришествие Гутенберга. — М.: Анкей-Экотрендз, 1999. — 271 с.

37. *Кречман Д.Л., Пушков А.И.* Мультимедиа своими руками. — СПб.: БХВ— Санкт-Петербург, 1999. — 528 с.

38. *Линдсней П., Норман Д.* Переработка информации у человека: Пер. с англ. — М.: Мир, 1974. — 550 с.

39. *Морозов В.П., Тихомиров В.П., Хрусталев Е.Ю.* Гипертексты в экономике. Информационная технология моделирования: Учебное пособие. — М.: Финансы и статистика, 1997. — 256 с.

40. *Нельсон Т.* Информационные системы будущего // Информационный поиск / Пер. с англ. под ред. К.Н. Трофимова. — М.: Воениздат, 1970.

41. *Поликсахин А.В., Савин А.Ю.* Гипертекст: сущность, состояние, перспективы. — М., 1993. — 128 с.

42. *Спрингер С., Денч Г.* Левый мир, правый мир. — М.: Мир, 1983. — 256 с.

43. Средства дистанционного обучения. Методика, технология, инструментарий / Авторы: С.В. Агапонов, З.О. Джалиашвили, Д.Л. Кречман, И.С. Никифоров, Е.С. Ченосова, А.В. Юрков / Под ред. З.О. Джалиашвили. — СПб.: БХВ-Петербург, 2003. — 336 с.

44. *Сэлтон Г.* Автоматическая обработка, хранение и поиск информации: Пер. с англ. / Под ред. А.И. Китова. — М.: Сов. радио, 1973. — 560 с.

45. *Bush V.* As we may think // The Atlantic Monthly. — 1945. — Vol. 176, № 1. — P. 101—108.

46. *Nelson T.H.* Managing Immense Storage // Byte. — 1988. — Vol. 13, № 1. — P. 225—238.

47. *Nielsen J.* Hypertext & Hypermedia. — Oxford: Oxford University Press, 1990. — 263 p.

48. *Van Rijsbergen C.J.* Information retrieval. — London: Butterworths, 1979.

§ 3.2

49. *Антопольский А.Б.* Лингвистическое обеспечение электронных библиотек. — М.: ФГУП Научно-технический центр «Информрегистр», 2003. — 302 с.

50. *Горелов И.Н.* Разговор с компьютером: Психолингвистический аспект проблемы. — М.: Наука, 1987. — 256 с.

51. *Дракин В.И., Попов Э.В., Преображенский А.Б.* Общение конечных пользователей с системами обработки данных. — М.: Радио и связь, 1988. — 288 с.

52. *Ершов А.П.* К методологии построения диалоговых систем, феномен деловой прозы. — Препринт / ВЦ СО АН СССР. 1978. — № 1. — 29 с.

53. *Любарский Ю.Я.* Интеллектуальные информационные системы. — М.: Наука, 1990. — 232 с.

54. *Мельчук И.А.* Опыт теории лингвистических моделей <<смысл—текст>>. — М.: Наука, 1974.

55. Поиск знаний — как основа управления знаниями [Электронный ресурс] / Компания «Весть-МетаТехнология». — Электрон. текстовые дан. (370373 байт). — М.: Компания «Весть-МетаТехнология», 2003. — Режим доступа : http://www.vest-meta.ru/tech/knowledge/Knowledge-Management-Rus.pdf.

56. Поисковые системы в сети Интернет [Электронный ресурс] / В. Тихонов. — Электрон. текстовые дан. — М. : CITFORUM.RU, 2000. — Режим доступа: http://citforum.nis.nnov.su/internet/search/searchsystems.shtml.

57. *Попов Э.В.* Общение с ЭВМ на естественном языке. — М.: Наука, 1982. — 360 с.

58. *Поспелов Д.А.* Послесловие // Разговор с компьютером: Психолингвистический аспект проблемы / И.Н. Горелов. — М.: Наука, 1987. — С. 230—250.

59. Семантические технологии НейрОК [Электронный ресурс] / Компания «НейрОК Интелсофт». — Электрон. текстовые дан. (1362702 байт). — М.: Компания «НейрОК Интелсофт», 2003. — Режим доступа: http://soft.neurok.ru/pub/wp/wp_neuroksemtech.zip.

60. *Сэлтон Г.* Автоматическая обработка, хранение и поиск информации: Пер. с англ. / Под ред. А.И. Китова. — М.: Сов. радио, 1973. — 560 с.

61. *Храмцов П.* Информационно-поисковые системы Internet // Открытые системы. 1996. — № 3.

62. *Храмцов П.* Моделирование и анализ работы информационно-поисковых систем Internet // Открытые системы. 1996. — № 6.

63. *Шенк Р.* Обработка концептуальной информации: Пер с англ. — М.: Энергия, 1980. — 360 с.

64. Search Engine Sizes [Электронный ресурс] / Danny Sullivan, Search Engine Watch. — Электрон. текстовые дан. — [Б.м.] : Search Engine Watch, 2000. — Режим доступа : http://www.searchenginewatch.com/reports/sizes.html — Англ.

§ 3.3

65. *Хан У., Мани И.* Системы автоматического реферирования // Открытые системы. 2000. — № 12.

66. *Шенк Р.* Обработка концептуальной информации: Пер с англ. — М.: Энергия, 1980. — 360 с.

67. Design and Implementation of the WordNet Lexical Database and Searching Software [Электронный ресурс] / R. Beckwith, G.A. Miller, R. Tengi. — Электрон. текстовые дан. (367763 байт). — Princeton : Princeton University, 1993. — Режим доступа : http://www.cogsci.princeton.edu/~wn/5papers.pdf. — Англ.

68. *Fellbaum C.* English Verbs as a Semantic Net // International Journal of Lexicography. 1990. — № 3(4). — P. 278—301.

69. *Gross D., Miller K.J.* Adjectives in WordNet // International Journal of Lexicography. 1990. — № 3(4). — P. 265—277.

70. Introduction to WordNet: An On-line Lexical Database / G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K.J. Miller // International Journal of Lexicography. 1990. — № 3(4). — P. 235—244.

71. *Lenat D.B.* CYC: A Large-Scale Investment in Knowledge Infrastructure // Communications of the ASM 38. 1995. — № 11.

72. *Miller G.A.* Nouns in WordNet: a Lexical Inheritance System // International Journal of Lexicography. 1990. — № 3(4). — P. 245—264.

73. OpenCyc.org [Электронный ресурс] : Formalized Common Knowledge / Cycorp, Inc. — Электрон. текстовые дан. — [Austin]: Cycorp, [2002]. — Режим доступа: http://www.opencyc.org. — Англ. — Содерж.: Общее описание проекта.

74. The Cyc Knowledge Server [Электронный ресурс] / Cycorp, Inc. — Электрон. текстовые дан. — [Austin]: Cycorp, [2002]. — Режим доступа: http://www.cyc.com/products2.html. — Англ. — Содерж.: Описание возможностей продукта.

§ 3.4

75. *Гинзбург С.* Математическая теория контекстно-свободных языков: Пер. с англ. — М.: Мир, 1970. — 328 с.

76. *Королев Э.И.* Промышленные системы машинного перевода. — М.: ВЦП, 1991. — 104 с.

77. *Кулагина О.С.* Исследования по машинному переводу. — М.: Наука, 1979. — 320 с.

78. *Кулагина О.С.* Машинный перевод: современное состояние // Семиотика и информатика. 1989. — № 29. — С. 5—33.

79. Лингвистический процессор для сложных информационных систем / Ю.Д. Апресян, И.М. Богуславский, Л.Л. Иомдин и др. — М.: Наука, 1992. — 255 с.

80. *Мельчук И.А.* Опыт теории лингвистических моделей <<смысл—текст>>. — М.: Наука, 1974.

81. *Попов Э.В.* Общение с ЭВМ на естественном языке. — М.: Наука, 1982. — 360 с.

§ 3.5

82. *Андреев А.М., Березкин Д.В., Сюзев В.В., Шабанов В.И.* Модели и методы автоматической классификации текстовых документов // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2003. — № 4. — С. 64—92.

83. *Антопольский А.Б.* Лингвистическое обеспечение электронных библиотек. — М.: ФГУП Научно-технический центр «Информрегистр», 2003. — 302 с.

84. *Башмаков А.И., Старых В.А.* Систематизация информационных ресурсов для сферы образования: классификация и метаданные. — М.: «Европейский центр по качеству», 2003. — 384 с.

85. *Дюк В., Самойленко А.* Data mining: учебный курс. — СПб.: Питер, 2001. — 368 с.

86. *Клименко С.В., Крохин И.В., Кущ В.М., Лагутин Ю.Л.* Электронные документы в корпоративных сетях: второе пришествие Гутенберга. — М.: Анкей-Экотрендз, 1999. — 271 с.

87. Поиск знаний — как основа управления знаниями [Электронный ресурс] / Компания «Весть-МетаТехнология». — Электрон. текстовые дан. (370373 байт). — М.: Компания «Весть-МетаТехнология», 2003. — Режим доступа: http://www.vest-meta.ru/tech/knowledge/Knowledge-Management-Rus.pdf.

88. *Van Rijsbergen C.J.* Information retrieval. — London: Butterworths, 1979.

§ 3.6

89. *Карташова Е.* Интеллектуальные поисковые системы Excalibur // Сети. 1997. — № 6.

90. *Клименко С.В., Крохин И.В., Кущ В.М., Лагутин Ю.Л.* Электронные документы в корпоративных сетях: второе пришествие Гутенберга. — М.: Анкей-Экотрендз, 1999. — 271 с.

91. *Когаловский М.Р.* Перспективные технологии информационных систем. — М.: ДМК Пресс; М.: Компания АйТи, 2003. — 288 с.

92. *Когаловский М.Р.* Энциклопедия технологий баз данных. — М.: Финансы и статистика, 2002. — 800 с.

93. *Кохонен Т.* Ассоциативная память. — М.: Мир, 1980.

94. Поиск знаний — как основа управления знаниями [Электронный ресурс] / Компания «Весть-МетаТехнология». — Электрон. текстовые дан. (370373 байт). — М.: Компания «Весть-МетаТехнология», 2003. — Режим доступа: http://www.vest-meta.ru/tech/knowledge/Knowledge-Management-Rus.pdf.

95. Семантические технологии НейрОК [Электронный ресурс] / Компания «НейрОК Интелсофт». — Электрон. текстовые дан. (1362702 байт). — М.: Компания «НейрОК Интелсофт», 2003. — Режим доступа: http://soft.neurok.ru/pub/wp/wp_neuroksemtech.zip.

§ 4.1

96. *Антопольский А.Б.* Лингвистическое обеспечение электронных библиотек. – М.: ФГУП Научно-технический центр «Информрегистр», 2003. — 302 с.

97. *Башмаков А.И., Старых В.А.* Систематизация информационных ресурсов для сферы образования: классификация и метаданные. — М.: «Европейский центр по качеству», 2003. — 384 с.

98. *Башмаков И.А., Харченко А.С.* Использование систем метаданных для описания информационных образовательных ресурсов // Международный форум информатизации-2003: Доклады международной конференции «Информационные средства и технологии». В 3 т. Т. 1. — М.: Янус-К, 2003. — С. 198—201.

99. ГОСТ 7.14-98 (ИСО 2709-96). Формат для обмена информацией. Структура записи.

100. ГОСТ 7.19-85. Коммуникативный формат для обмена библиографическими данными на магнитной ленте. Содержание записи.

101. ГОСТ 7.52-85. Коммуникативный формат для обмена библиографическими данными на магнитной ленте. Поисковый образ документа.

102. ГОСТ 7.70-96. Описание баз данных и машиночитаемых информационных массивов. Состав и обозначение характеристик.

103. *Когаловский М.Р.* Энциклопедия технологий баз данных. — М.: Финансы и статистика, 2002. — 800 с.

104. *Рэй Э.* Изучаем XML. — СПб.: Символ-Плюс, 2001. — 408 с.

105. Средства удаленного доступа к информации и корпоративные электронные библиотеки образовательных ресурсов: Пособие для слушателей Федеральной программы развития образования / Грибов В.Т., Левова Л.В., Ефремов С.В. и др. — М.: МЭИ, 2002. — 228 с.

106. Dublin Core Metadata Element Set: Reference Description [Электронный ресурс] / DCMI. — Version 1.1. — Электрон. текстовые дан. — [USA]: DCMI, 1999. — Режим доступа: http://dublincore.org/documents/dces. — Англ.

107. vCard. The Electronic Business Card. A versit Consortium Specification [Электронный ресурс] / versit Consortium. — Version 2.1. — Электрон. текстовые дан. — [USA]: IMC, 1996. — Режим доступа: http://www.imc.org/pdi. — Англ.

108. IETF RFC 2425:1998. A MIME Content-Type for Directory Information [Электронный ресурс] / T. Howes, M. Smith, F. Dawson. — Электрон. текстовые дан. (66281 байт). — [USA]: IETF, 1998. — Режим доступа: http://www.ietf.org/rfc/rfc2425.txt. — Англ.

109. IETF RFC 2426:1998. vCard MIME Directory Profile [Электронный ресурс] / F. Dawson, T. Howes. — Электрон. текстовые дан. (77003 байт). — [USA]: IETF, 1998. — Режим доступа: http://www.ietf.org/rfc/rfc2426.txt. — Англ.

110. CCITT (ITU) Recommendations X.500-X.521. Data Communication Networks: Directory / CCITT Blue Book, Fascicle VIII.8. — [USA]: CCITT, 1988.

111. ISO/IEC 11179-1:1999. Information technology — Specification and standardization of data elements — Part 1: Framework for the specification and standardization of data elements.

112. IETF RFC 2045:1996. Multipurpose Internet Mail Extensions (MIME). Part One: Format of Internet Message Bodies [Электронный ресурс] / N. Freed, N. Borenstein. — Электрон. текстовые дан. (74673 байт). — [USA]: IETF, 1996. — Режим доступа: http://www.ietf.org/rfc/rfc2045.txt. — Англ.

113. IETF RFC 2046:1996. Multipurpose Internet Mail Extensions (MIME). Part Two: Media Types [Электронный ресурс] / N. Freed, N. Borenstein. — Электрон. текстовые дан. (108323 байт). — [USA]: IETF, 1996. — Режим доступа: http://www.ietf.org/rfc/rfc2046.txt. — Англ.

114. IEEE 1484.12.1-2002. Learning Object Metadata standard. — New York: IEEE, 2002.

115. ISO 8879:1986. Information processing — Text and office systems — Standard Generalized Markup Language (SGML).

116. RDF/XML Syntax Specification [Электронный ресурс] / W3C. — W3C Working Draft 25.03.2002. — Электрон. текстовые дан. — [Б.м.]: W3C, 2002. — Режим доступа: http://www.w3.org/TR/2002/WD-rdf-syntax-grammar-20020325. — Англ.

§ 4.2

117. *Граймс С.* Семантическая паутина // Корпоративные системы. — 2002. — № 15(56).

118. *Грейвс М.* Проектирование баз данных на основе XML.: Пер. с англ. — М.: Издательский дом «Вильямс», 2002. — 640 с.

119. *Когаловский М.Р.* Перспективные технологии информационных систем. — М.: ДМК Пресс; М.: Компания АйТи, 2003. — 288 с.

120. *Когаловский М.Р.* Энциклопедия технологий баз данных. — М.: Финансы и статистика, 2002. — 800 с.

121. *Рэй Э.* Изучаем XML. — СПб.: Символ-Плюс, 2001. — 408 с.

§ 5.1

122. Автоматизация поискового конструирования (искусственный интеллект в машинном проектировании) / А.И. Половинкин, Н.К. Бобков, Г.Я. Буш и др.; Под ред. А.И. Половинкина. — М.: Радио и связь, 1981. — 344 с.

123. *Амамия М., Танака Ю.* Архитектура ЭВМ и искусственный интеллект: Пер. с япон. — М.: Мир, 1993. — 400 с.

124. *Башлыков А.А., Еремеев А.П.* Экспертные системы поддержки принятия решений в энергетике / Под. ред. А.Ф. Дьякова. — М.: Издательство МЭИ, 1994. — 216 с.

125. *Башмаков И.А., Крылович С.В.* Объектно-ориентированная парадигма и тенденции развития свойства активности баз знаний // Программные продукты и системы. 2000. — № 1. — С. 12—16.

126. Будущее искусственного интеллекта. — М.: Наука, 1991. — 302 с.

127. Виртуальная реальность в психологии и искусственном интеллекте. — М.: Российская Ассоциация искусственного интеллекта, 1998. — 316 с.

128. *Гаврилова Т.А., Червинская К.Р.* Извлечение и структурирование знаний для экспертных систем. — М.: Радио и связь, 1992. — 200 с.

129. *Заде Л.* Понятие лингвистической переменной и его применение к принятию приближенных решений: Пер. с англ. — М.: Мир, 1976. — 165 с.

130. *Зенкин А.А.* Когнитивная компьютерная графика / Под ред. Д.А. Поспелова. — М.: Наука, 1991. — 192 с.

131. *Кандрашина Е.Ю., Литвинцева Л.В., Поспелов Д.А.* Представление знаний о времени и пространстве в интеллектуальных системах / Под ред. Д.А. Поспелова. — М.: Наука, 1989. — 328 с.

132. Креативная педагогика: методология, теория, практика / А.И. Башмаков, И.А. Башмаков, А.И. Владимиров и др.; Под ред. Ю.Г. Круглова. — М.: МГОПУ им. М.А. Шолохова: Изд. центр «Альфа», 2002. — 240 с.

133. *Линдсней П., Норман Д.* Переработка информации у человека: Пер. с англ. — М.: Мир, 1974. — 550 с.

134. Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д.А. Поспелова. — М.: Наука, 1986. — 312 с.

135. Нечеткие множества и теория возможностей. Последние достижения: Пер. с англ. / Под ред. Р.Р. Ягера. — М.: Радио и связь, 1986. — 408 с.

136. Обработка нечеткой информации в системах принятия решений / А.Н. Борисов, А.В. Алексеев, Г.В. Меркурьева и др. — М.: Радио и связь, 1989. — 304 с.

137. *Орловский С.А.* Проблемы принятия решений при нечеткой исходной информации. — М.: Наука, 1981. — 208 с.

138. *Осуга С.* Обработка знаний: Пер. с япон. — М.: Мир, 1989. — 293 с.

139. *Попов Э.В.* Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. — М.: Наука, 1987. — 288 с.

140. Прикладные нечеткие системы: Пер. с япон. / К. Асаи, Д. Ватада, С. Иван и др.; Под ред. Т. Тэрано, К. Асаи, М. Сугэно. — М.: Мир, 1993. — 368 с.

141. Приобретение знаний: Пер. с япон. / Под ред. С. Осуги, Ю. Саэки. — М.: Мир, 1990. — 304 с.

142. Техническое творчество: теория, методология, практика. Энциклопедический словарь-справочник / Под ред. А.И. Половинкина, В.В. Попова. — М.: НПО «Информ-система», 1995. — 408 с.

143. *Уинстон П.* Искусственный интеллект: Пер. с англ. — М.: Мир, 1980. — 519 с.

144. *Цаленко М.Ш.* Моделирование семантики в базах данных. — М.: Наука, 1989. — 288 с.

145. *Шрейдер Ю.А.* Равенство, сходство, порядок. — М.: Наука, 1971. — 255 с.

§ 5.2

146. *Башмаков А.И.* Рассуждения по аналогии // Новости искусственного интеллекта. 1992. — № 2. — С. 8—35.

147. *Вагин В.Н.* Дедукция и обобщение в системах принятия решений. — М.: Наука, 1988. — 384 с.

148. *Кандрашина Е.Ю., Литвинцева Л.В., Поспелов Д.А.* Представление знаний о времени и пространстве в интеллектуальных системах / Под ред. Д.А. Поспелова. — М.: Наука, 1989. — 328 с.

149. *Кондаков Н.И.* Логический словарь. — М.: Наука, 1971. — 638 с.

150. *Кузин Л.Т.* Основы кибернетики. В 2 т. Т. 2. Основы кибернетических моделей. — М.: Энергия, 1979. — 584 с.

151. *Кузнецов И.П.* Механизмы обработки семантической информации. — М.: Наука, 1978. — 174 с.

152. Логический подход к искусственному интеллекту: от классической логики к логическому программированию: Пер. с франц. / Тейз А., Грибомон П., Луи Ж. и др. — М.: Мир, 1990. — 432 с.

153. *Лорьер Ж.Л.* Системы искусственного интеллекта: Пер с франц. — М.: Мир, 1991. — 568 с.

154. *Любарский Ю.Я.* Интеллектуальные информационные системы. — М.: Наука, 1990. — 232 с.

155. Метод моделирования поиска и умозаключения по аналогии / Башмаков А.И. — М.: МЭИ, 1993. — Деп. в НИИВО 13.12.1993, № 289—93. — 220 с.

156. *Минский М.* Фреймы для представления знаний: Пер. с англ. — М.: Энергия, 1979. — 152 с.

157. *Непейвода Н.Н.* Прикладная логика: Учеб. пособие. — Новосибирск: Изд-во Новосиб. ун-та, 2000. — 521 с.

158. Обобщенная модель представления предметной области / А.И. Башмаков. — М.: МЭИ, 1997. — Деп. в ВИНИТИ 10.06.97, № 1933-В97. — 299 с.

159. *Осуга С.* Обработка знаний: Пер. с япон. — М.: Мир, 1989. — 293 с.

160. ISO/IEC 10746-1:1998. Information technology — Open Distributed Processing — Reference model: Overview.

161. ISO/IEC 10746-2:1996. Information technology — Open Distributed Processing — Reference model: Foundations.

162. ISO/IEC 10746-3:1996. Information technology — Open Distributed Processing — Reference Model: Architecture.

163. ISO/IEC 10746-4:1998. Information technology — Open Distributed Processing — Reference Model: Architectural semantics.

164. *Kleppe A., Warmer J., Bast W.* MDA Explained: The Model Driven Architecture — Practice and Promise. — Addison-Wesley, 2003. — 192 p.

165. MDA and RM-ODP: two approaches in modern ontological engineering [Электронный ресурс] / A. Naumenko, A. Wegmann. — Электрон. текстовые дан. (49428 байт). — Lausanne : Swiss Federal Institute of Technology, 2002. — Режим доступа : http://lcawww.epfl.ch/Publications/Naumenko/TR01_047.pdf. — Англ.

166. MDA Guide [Электронный ресурс] / Edited by J. Miller and J. Mukerji, Object Management Group, Inc. — Version 1.0.1. — Электрон. текстовые дан. (328400 байт). — [USA]: OMG, 2003. — Режим доступа: http://www.omg.org/cgi-bin/apps/doc?omg/03-05-01.pdf. — Англ.

167. MDA Specifications [Электронный ресурс] / Object Management Group, Inc. — Электрон. текстовые дан. — [USA] : OMG, 2003. — Режим доступа: http://www.omg.org/mda/specs.htm. — Англ.

168. Model Driven Architecture (MDA) [Электронный ресурс] / Edited by J. Miller and J. Mukerji, Object Management Group, Inc. — Document number ormsc/2001-07-01. — Электрон. текстовые дан. (299041 байт). — [USA]: OMG, 2001. — Режим доступа: http://www.omg.org/cgi-bin/apps/doc?ormsc/01-07-01.pdf. — Англ.

§ 5.3

169. *Башмаков А.И., Башмаков И.А.* Механизмы наследования, выявления и разрешения противоречий в обобщенной модели представления предметной области. Часть I // Техническая кибернетика. 1994. — № 5. — С. 14—27.

170. *Башмаков А.И., Башмаков И.А.* Механизмы наследования, выявления и разрешения противоречий в обобщенной модели представления предметной области. Часть II // Теория и системы управления. 1995, № 3. — С. 175—189.

171. *Башмаков И.А., Рабинович П.Д.* Анализ моделей семантических сетей как математического аппарата представления знаний об учебном материале // Справочник. Инженерный журнал. 2002. — № 7. — С. 55—60.

172. *Башмаков И.А., Рабинович П.Д.* О концепции информатизации учебного процесса // Вестник МЭИ. 2003. — № 4. — С. 105—110.

173. *Кузнецов И.П.* Расширенные семантические сети для представления и обработки знаний // Системы и средства информатики: Ежегод. Вып. 4 / РАН. Институт проблем информатики. — М., 1993. — С. 70—83.

174. Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д.А. Поспелова. — М.: Наука, 1986. — 312 с.

175. Обобщенная модель представления предметной области / А.И. Башмаков. — М.: МЭИ, 1997. — Деп. в ВИНИТИ 10.06.97, № 1933-В97. — 299 с.

176. *Орловский С.А.* Проблемы принятия решений при нечеткой исходной информации. — М.: Наука, 1981. — 208 с.

177. *Осипов Г.С.* Построение моделей предметных областей. Ч. I. Неоднородные семантические сети // Известия РАН. Техническая кибернетика. 1990. — № 5. — С. 32—45.

178. *Перминов И.А.* Нечеткая объектно-ориентированная семантическая сеть // Доклады Международной конференции «Информационные средства и технологии» Международного форума информатизации МФИ-99. Т. 3. — М.: Изд-во «Станкин», 1999. — С. 37—40.

179. *Перминов И.А.* Объектно-ориентированный язык для оперирования семантическими сетями // Тезисы докладов Международной конференции «Информационные средства и технологии» Международного форума информатизации МФИ-2000. Т. 2. — М.: Изд-во «Станкин», 2000. — С. 212—215.

180. *Уемов А.И.* Вещи, свойства и отношения. — М.: Изд. АН СССР, 1963. — 184 с.

181. *Уемов А.И.* Логические основы метода моделирования. — М.: Мысль, 1971. — 312 с.

§ 5.4

182. *Бениаминов Е.М., Балдина Д.М.* Система представления знаний Ontolingua – принципы и перспективы // Научно-техническая информация. Сер. 2. Информационные процессы и системы. 1999. — № 10. — С. 26—32.

183. *Бениаминов Е.М., Машунина М.Ю.* Принципы построения открытого языка шаблонных выражений в системе представления знания // Научно-техническая информация. Сер. 2. Информационные процессы и системы. 2000. — № 7. — С. 10—17.

184. Виртуальный фонд естественнонаучных и научно-технических эффектов «Эффективная физика» / А.И. Башмаков, Н.А. Бухарова, Д.Н. Жедяевский, А.А. Поляков, В.В. Попов // Компьютерные инструменты в образовании. 2003. — № 3. — С. 3—13.

185. *Клещев А.С., Артемьева И.Л.* Необогащенная система логических соотношений. Часть 1 // Научно-техническая информация. Сер. 2. Информационные процессы и системы. 2000. — № 7. — С. 18—28.

186. Конструктор онтологий мультиагентных систем [Электронный ресурс] / В. Андреев, К. Ивкушкин, И. Минаков, Г. Ржевский, П. Скобелев. — Электрон. текстовые дан. — Самара : MagentA Corporation, Plc., [2001]. — Режим доступа : http://www.kg.ru/Publish/artic31.htm.

187. Организация эффективного поиска на основе онтологий [Электронный ресурс] / О.И. Россеева, Ю.А. Загорулько. — Электрон. текстовые дан. (105366 байт). — [Б.м.]: [Российский НИИ Искусственного Интеллекта, Институт систем информатики СО РАН], [2001]. — Режим доступа: http://www.dialog-21.ru/Archive/2001/volume2/2_49.htm.

188. Стандарт онтологического исследования IDEF5 [Электронный ресурс] / Г. Верников. — Электрон. текстовые дан. — М.: CITFORUM.RU, [1999]. — Режим доступа: http://www.citforum.ru/cfin/idef/idef5.shtml.

189. *Bateman J., Kasper R., Moore J., Whitney R.* A general organization of knowledge for natural language processing: The Penman Upper Model. — Technical Report. — Marina del Rey, California: Information Sciences Institute, 1989.

190. Community is Knowledge! / Benjamins V.R., Fensel D., et. al. // Knowledge Acquisition Workshop KAW98. — Banff, 1998.

191. *Blazquez M., Fernandez M., Garcia-Pinar J.M., Gomez-Perez A.* Building Ontologies at the Knowledge Level Using the Ontology Design Environment // Knowledge Acquisition Workshop KAW98. — Banff, 1998.

192. *Braetman J.A., Magnini B., Rinaldi F.* The Generalized Italian, German, English Upper Model // Proceedings of the ECAI'94 Workshop: Comparison of Implemented Ontologies. — Amsterdam, 1994.

193. DAML+OIL (March 2001) Reference Description [Электронный ресурс] / W3C. — W3C Note 18.12.2001. — Электрон. текстовые дан. — [Б.м.]: W3C, 2001. — Режим доступа: http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218. — Англ.

194. *Fernandez M., Gomez-Perez A., Juristo N.* METHONTOLOGY: From Ontological Art Toward Ontological Engineering // Spring Symposium Series on Ontological Engineering AAAI-97. — Stanford: Stanford University, 1997.

195. *Genesereth M.R., Fikes E.R.* Knowledge Interchange Format. Version 3.0: Reference Manual. — Stanford: Stanford University, Computer Science Department, 1992.

196. *Gruber T.R.* Towards Principles for the Design of Ontologies Used for Knowledge Sharing // International Journal of Human and Computer Studies. 1993. — № 43(5/6). — P. 907—928.

197. *Gruber T.R.* A translation approach to portable ontologies // Knowledge Acquisition. 1993. — № 5(2). — P. 199—220.

198. *Guarino N., Guaretta P.* Ontologies and Knowledge Bases. Towards a Technological Clarification // Towards Very Large Knowledge Bases. — Amsterdam: IOS Press, 1995.

199. *Heijst G. van, Schreiber A.T., Wielinga B.J.* Using Explicit Ontologies in KBS Development // International Journal of Human and Computer Studies. 1996. — № 46(2—3). — P. 183—292.

200. How to Write F-Logic Programs. A Tutorial for the Language F-Logic covers OntoBroker Version 3.62 [Электронный ресурс] / ontoprise GmbH. — Электрон. текстовые дан. (276396 байт) — Karlsruhe : ontoprise GmbH, 2002. — Режим доступа: http://www.ontoprise.de/documents/tutorial_flogic.pdf. — Англ.

201. IDEF5 Method Report / Knowledge Base System, Inc. — College Station, Texas: KBS, 1994. — 187 p.

202. ISO 10303. Industrial automation systems and integration — Product data representation and exchange.

203. Knowledge Interchange Format (KIF) [Электронный ресурс] : Draft proposed American National Standard (dpANS) : NCITS.T2/98-004 / Michael R. Genesereth. — Электрон. текстовые дан. (88164 байт). — [Stanford]: Knowledge System Laboratory, [1998]. — Режим доступа: http://logic.stanford.edu/kif/dpans.htm. — Англ.

204. KSL Ontology Server Projects [Электронный ресурс] / Knowledge System Laboratory. — Электрон. текстовые дан. — Stanford: Stanford University, [2001]. — Режим доступа: http://www.ksl.stanford.edu/software/ontolingua/ontology-server-projects.html. — Англ.

205. *Lenat D.B.* CYC: A Large-Scale Investment in Knowledge Infrastructure // Communications of the ASM 38. 1995. — № 11.

206. *Luke S., Spector L., Rager D.* Ontology-Based Knowledge Discovery on the World-Wide-Web // Workshop on the Internet-Based Information Systems AAAI-96. — Portland (Oregon), 1996.

207. Ontolingua [Электронный ресурс] : Software Description / Knowledge System Laboratory. — Электрон. текстовые дан. — Stanford: Stanford University, 2001. — Режим доступа: http://www.ksl.stanford.edu/software/ontolingua. — Англ.

208. OpenCyc.org [Электронный ресурс] : Formalized Common Knowledge / Cycorp, Inc. — Электрон. текстовые дан. — [Austin]: Cycorp, [2002]. — Режим доступа: http://www.opencyc.org. — Англ. — Содерж.: Общее описание проекта.

209. The Cyc Knowledge Server [Электронный ресурс] / Cycorp, Inc. — Электрон. текстовые дан. — [Austin]: Cycorp, [2002]. — Режим доступа: http://www.cyc.com/products2.html. — Англ. — Содерж.: Описание возможностей продукта.

210. TOVE Manual [Электронный ресурс] / Department of Industrial Engineering, University of Toronto. — Электрон. текстовые дан. — Toronto : University of Toronto, [1999]. — Режим доступа : http://www.ie.utoronto.ca/EIL/tove. — Англ.

211. *Uschold M., Gruninger M.* ONTOLOGIES: Principles, Methods and Applications // Knowledge Engineering Review. 1996. — Vol. 11, № 2.

§ 5.5

212. *Башмаков А.И., Башмаков И.А.* Механизмы наследования, выявления и разрешения противоречий в обобщенной модели представления предметной области. Часть I // Техническая кибернетика. 1994. — № 5. — С. 14—27.

213. *Башмаков А.И., Башмаков И.А.* Механизмы наследования, выявления и разрешения противоречий в обобщенной модели представления предметной области. Часть II // Теория и системы управления. 1995. — № 3. — С. 175—189.

214. *Башмаков А.И., Башмаков И.А.* Подход к обеспечению верифицируемости объектно-ориентированных баз знаний // Вестник МЭИ. 1999. — № 3. — С. 85—92.

215. *Башмаков А.И., Башмаков И.А.* Стратегии разрешения противоречий в базах знаний // Вестник МЭИ. 2001. — № 3. — С. 80—87.

216. *Башмаков А.И., Башмаков И.А.* Уровни операций интеллектуальной верификации в объектно-ориентированных базах знаний // Тезисы докладов Международной конференции «Информационные средства и технологии» Международного форума информатизации МФИ-2000. Т. 2. — М.: Изд-во «Станкин», 2000. — С. 168—171.

217. *Кондаков Н.И.* Логический словарь. — М.: Наука, 1971. — 638 с.

218. Метод моделирования поиска и умозаключения по аналогии / Башмаков А.И. — М.: МЭИ, 1993. — Деп. в НИИВО 13.12.1993, № 289-93. — 220 с.

219. Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д.А. Поспелова. — М.: Наука, 1986. — 312 с.

220. Обобщенная модель представления предметной области / А.И. Башмаков. — М.: МЭИ, 1997. — Деп. в ВИНИТИ 10.06.97, № 1933-В97. — 299 с.

221. Обработка нечеткой информации в системах принятия решений / А.Н. Борисов, А.В. Алексеев, Г.В. Меркурьева и др. — М.: Радио и связь, 1989. — 304 с.

222. *Орловский С.А.* Проблемы принятия решений при нечеткой исходной информации. — М.: Наука, 1981. — 208 с.

223. *Саймон А.Р.* Стратегические технологии баз данных: менеджмент на 2000 год: Пер. с англ. / Под ред. и с предисл. М.Р. Когаловского. — М.: Финансы и статистика, 1999. — 479 с.

224. *Цаленко М.Ш.* Моделирование семантики в базах данных. — М.: Наука, 1989. — 288 с.

§ 6.1

225. *Балдин Е.В., Шашкин Л.О.* Генетические алгоритмы: возможности и ограничения // Научно-техническая информация. Сер. 2. Информационные процессы и системы. 2000. — № 8. — С. 19—33.

226. *Галушкин А.И.* Нейрокомпьютеры. Кн. 3: Учебное пособие для вузов. — М.: ИПЖР, 2000. — 528 с.

227. *Галушкин А.И.* Теория нейронных сетей. Кн. 1: Учебное пособие для вузов. — М.: ИПРЖР, 2000. — 416 с.

228. *Горбань А.Н.* Обучение нейронных сетей. — М.: СП ПараГраф, 1990. — 159 с.

229. Компьютеры и мозг [Электронный ресурс] / А.А. Ежов, С.А. Шумский. — Электрон. текстовые дан. (416768 байт). — М.: Компания «НейрОК Интелсофт», 1999. — Режим доступа: http://www.neurok.ru/pub/archive/cb.zip.

230. *Уоссерман Ф.* Нейрокомпьютерная техника. — М.: Мир, 1992.

§ 6.2

231. *Галушкин А.И.* Нейрокомпьютеры. Кн. 3: Учебное пособие для вузов. — М.: ИПЖР, 2000. — 528 с.

232. *Комарцова Л.Г., Максимов А.В.* Нейрокомпьютеры: Учеб. пособие для вузов. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. — 320 с.

233. *Кохонен Т.* Ассоциативная память. — М.: Мир, 1980.

234. *Уоссерман Ф.* Нейрокомпьютерная техника. — М.: Мир, 1992.

§ 6.3

235. *Байдык Т.Н.* Нейронные сети и задачи искусственного интеллекта. — Киев: Наукова думка, 2001.

236. *Воронков Г.С., Чечкин А.В.* Нейронные семиотические системы как интеллектуальные среды // Труды пятой национальной конференции с международным участием «Искусственный интеллект-96». Т. 1. — Казань, 1996. — С. 26—35.

237. *Иванов В.В.* Нейролингвистика // Биологические и кибернетические аспекты речевой деятельности. — М., 1988. — С. 26—70.

§ 7.1

238. *Гаврилова Т.А.* Логико-лингвистическое управление как введение в управление знаниями // Новости искусственного интеллекта. 2002. — № 6. — С. 36—40.

239. *Гаврилова Т.А., Червинская К.Р.* Извлечение и структурирование знаний для экспертных систем. — М.: Радио и связь, 1992. — 200 с.

240. *Попов Э.В.* Корпоративные системы управления знаниями // Новости искусственного интеллекта. 2001. — № 1.

§ 7.2

241. *Галушкин А.И.* Теория нейронных сетей. Кн. 1: Учебное пособие для вузов. — М.: ИПРЖР, 2000. — 416 с.

242. *Глова В.И., Аникин И.В., Аджели М.Л.* Мягкие вычисления (SOFT COMPUTING) и их приложения: Учебное пособие / Под ред. В.И. Глова. — Казань: Изд-во Казан. гос. техн. ун-та, 2000. — 98 с.

243. *Григорьев П.А.* Методы интеллектуального анализа данных в предметных областях с частично детерминированными свойствами объектов: Автореф. дисс. ... канд. техн. наук. — М.: РГГУ, 2000. — 24 с.

244. *Дюк В., Самойленко А.* Data mining: учебный курс. — СПб.: Питер, 2001. — 368 с.

245. *Когаловский М.Р.* Перспективные технологии информационных систем. — М.: ДМК Пресс; М.: Компания АйТи, 2003. — 288 с.

246. Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д.А. Поспелова. — М.: Наука, 1986. — 312 с.

247. Нечеткие множества и теория возможностей. Последние достижения: Пер. с англ. / Под ред. Р.Р. Ягера. — М.: Радио и связь, 1986. — 408 с.

248. Прикладные нечеткие системы: Пер. с япон. / К. Асаи, Д. Ватада, С. Иван и др.; Под ред. Т. Тэрано, К. Асаи, М. Сугэно. — М.: Мир, 1993. — 368 с.

249. *Спирли Э.* Корпоративные хранилища данных. Планирование, разработка, реализация. Т. 1: Пер. с англ. — М.: Издательский дом «Вильямс», 2001.

250. *Стулов А.В.* Хранилища данных: основные архитектуры и принципы построения // Новости искусственного интеллекта. 2003. — № 2. — С. 37—41.

251. *Inman W.* Building the Data Warehouse. — New York: John Willey & Sons, 1992.

§ 7.3

252. Автоматизация поискового конструирования (искусственный интеллект в машинном проектировании) / А.И. Половинкин, Н.К. Бобков, Г.Я. Буш и др.; Под ред. А.И. Половинкина. — М.: Радио и связь, 1981. — 344 с.

253. *Альтшуллер Г.С.* Творчество как точная наука. — М.: Сов. радио, 1979.

254. *Альтшуллер Г.С.* Алгоритм изобретения. — М.: Московский рабочий, 1973. — 296 с.

255. *Андрейчиков А.В., Андрейчикова О.Н.* Компьютерная поддержка изобретательства (методы, системы, примеры применения). — М.: Машиностроение, 1998. — 476 с.

256. *Норенков И.П.* Основы автоматизированного проектирования: Учебник для вузов. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2000. — 360 с.

257. Техническое творчество: теория, методология, практика. Энциклопедический словарь-справочник / Под ред. А.И. Половинкина, В.В. Попова. — М.: НПО «Информ-система», 1995. — 408 с.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Абстракция 200–203, 205, 206, 225, 226, 228
Абстракция обоснований 209, 210
Автоматическая классификация документов 96–99, 114, 178, 179, 254, 267
Автоматическое реферирование и аннотирование 77–82, 88, 89, 100, 106, 116, 178, 192, 255, 267, 269, 273
Агент 73–75, 116, 193
Аннотация 46, 55, 77, 123

База данных (БД) 43, 56, 57, 64, 70, 73–75, 101, 104, 107, 109, 124, 129, 130, 141, 171, 176, 190, 199, 253, 254, 257, 259, 261, 263, 264, 268
База знаний (БЗ) 17–19, 64, 69, 70, 110, 137–139, 144–146, 149, 150, 155, 156, 173–176, 180, 190–192, 198–200, 206, 207, 216, 254, 255, 267–273
- замкнутая 198
- открытая 149, 198

Верификация знаний 138, 144, 146, 200, 206, 226, 264
Вершина связи 158, 163
Взаимообоснование 164, 168, 207
Взаимопереход 164, 167, 200, 204
Воспроизведение образов 249
Временная метка 168, 169, 210, 211, 219, 220, 227, 228

Гиперграфика 45
Гиперкуб 259–261
Гипермедиа 45
Гиперссылка 44–47, 49, 52, 62, 63, 74, 106, 111, 129, 132
Гипертекст (ГТ) 41, 42, 61–65, 105
Гипертекстовая информационная технология (ГИТ) 43, 55, 64

Дескриптор 51, 55–57, 70, 74, 98
Дублинское ядро 123–128

Естественный язык (ЕЯ) 18, 55, 56, 67–69, 79, 81, 82, 90, 93, 98, 108–110, 113, 142, 157, 158, 166, 184, 191, 267, 271

Замещение 228
Знания 18–20, 135–146
- априорные 137
- декларативные 138, 143, 144, 150, 153
- корпоративные 185, 254, 255
- лингвистические 93
- метазнания 19, 138, 199
- накапливаемые 138
- объектные 136
- процедурные 19, 138, 143, 144, 153
- эвристики 139, 140, 151
- экстралингвистические (знания о предметной области) 19, 93–95, 136, 157, 174, 179, 199

Значимость 169, 170

Избыточность 213, 216
Извлечение программы из доказательства 26
Индекс 56, 70, 72, 73, 75, 104, 107, 115
- дескриптора 55
- семантический 114, 269, 270
Индексирование 55, 56, 70, 72, 74–76, 79, 107, 177
Интеллектуальный анализ данных (ИАД) 257, 258, 262
Интерпретация 205, 225, 227
Интуиционистская логика 25, 27
Информативность 169, 170
Информационная полнота, коэффициент полноты 58–60, 176
Информационно-поисковая система (ИПС) 55–60, 64, 70, 116, 171, 176, 190, 194
- библиографическая 55
- гипертекстовая (ГИПС) 56, 64, 65
- документальная 55
- фактографическая 56
Информационно-справочная статья (ИСС) 44–47, 49, 51, 53
Информационный поиск 55, 57–60, 70, 71, 97, 108, 115, 176
- ассоциативный 115
- атрибутивный (по набору признаков) 57, 115
- индексный (двоичный) 109
- использующий статистические методы 110
- лексический 115
- многоязычный 109
- нечеткий 108, 111, 112
- по ключевым словам (дескрипторам) 55–57, 115, 171, 176
- по метаданным 57
- полнотекстовый 56, 57, 75, 112
- семантический (смысловой) 104, 106, 108–114, 177, 178, 267
Информационный ресурс (ИР) 71, 97, 120–122, 129, 132, 179, 254
Информационный шум, коэффициент шума 58–60, 129, 176
Искусственная нейронная сеть (ИНС) 108, 110, 115, 232, 235–244, 246
Искусственный интеллект (ИИ) 8, 17, 20, 64, 77, 135, 198
Исчисление высказываний 148
Исчисление предикатов первого порядка 148, 159, 181

Каталог ресурсов Internet 71, 72
Квалификатор 124, 125, 128, 179
Классификатор (в OCR-технологиях)
- признаковый 36
- структурный 36
- шаблонный 35
Классификатор 70, 111, 115, 116, 121, 126
Ключевое слово 51, 55–57, 70, 74, 98
Конкретизация 203, 225, 227
Концептуализация 23, 24, 173, 174, 188
Концептуальное проектирование технологий и техники 266
Концептуальные свойства знаний 140
Коэффициент точности информационного поиска 59
Куб данных 259–261

Лингвистическое обеспечение (ЛО) 82, 87, 94, 104
Логическое программирование 149

Машина вывода (МВ) 183, 188, 190, 191, 193
Машинный перевод (МП) 90, 254
Метаданные 57, 60, 120, 121, 123, 128, 129, 131, 179, 259, 261, 262
Метод составления выдержек 79
Методы интеллектуальной верификации знаний 206
Механизмы памяти 249, 250
Модель
- вычислительная (ВМ) 22–24
- гипертекста условно-типовая 47–49
- гипертекста формализованная 44–46
- документа 129–131

- линейных весовых коэффициентов 80
- метаданных 122
- онтологии 181–183
- элементарная сенсорная (ЭСМ) 248
- SAO 268, 269

Модель знаний 146, 147, 156, 164
- комплексная 155
- логическая 147–149, 171
- объектно-ориентированная 154, 155
- продукционная 150, 151
- сетевая 153, 157, 175, 181
- специальная 155
- фреймовая 151, 152

Монотонность 149, 198
Морфологический анализ 56, 66, 67, 70, 87, 93, 112, 268
Морфологический синтез 87, 93

Наследование в базе знаний 225
Нейрокибернетика 234
Нейрокомпьютер (НК) 235–237
Нейрон 232, 233, 235, 236, 238, 243, 247, 248
- квазирецепторный (КРН) 248–251
- нейрон-понятие 250, 251
- нейрон-слово 250, 251
- символьный (СН) 248–251

Нейропакет (НП) 235, 237, 238
Нейротехнология 231, 232, 236
Не-факторы 139
Нечеткое множество 139, 162, 163, 167, 168, 170, 220, 221
Новая информационная технология (НИТ) 17, 143

Обобщенный комплексный показатель эффективности информационного поиска 59
Обучение нейросети 232, 233, 236, 239, 242, 244
Ограниченный естественный язык (ОЕЯ) 21, 23, 62, 64, 67
Онтология 153, 173–175
- задач 183
- общих знаний (верхнего уровня) 179, 183, 191
- предметная 183, 192
- расширенная 183
- web-онтология 175

Пакет решения инженерных задач (ПРИЗ) 28
Парадигма нейрокомпьютинга 233
Платформа XML 129, 130
Плекс-грамматика 24
Плекс-элемент 24
Подходы к синтезу программ 22
Поисковая машина (поисковая система) 72–75, 129, 176, 183
Поисковое предписание 55
Поисковый запрос 55, 56, 58, 98, 109, 115–117, 176, 177, 270
Поисковый образ 55
Понимание текста на естественном языке 67–69, 79
Представление знаний 145, 157, 170, 174, 268
Приложение XML 130
Продукция (правило вывода) 25–27, 65, 139, 148, 150, 151, 159, 160, 255
Пролог 27, 151, 160, 162, 163
Противоречие 210, 212–216
- качества 215
- количества 212, 215
- сильное 210, 212, 215
- слабое 210, 212, 215
- техническое 276
- условное 213

Разрешение противоречий 206, 216, 276
- в пространстве 219, 220
- во времени 219, 220
- на фиксированном уровне 216, 217, 223, 225
- с выходом на другие уровни 225

Расчетно-логическая задача 21
Релевантность 55–58, 69, 75, 114–117, 152, 183
Реферат 55, 56, 77, 79, 88, 89, 106, 123

Рубрикатор 70, 111, 115, 116, 121, 126

Семантическая сеть 23, 49, 60, 68, 82, 101, 104, 105, 108–110, 112, 153, 157, 170
- неоднородная (НСС) 160, 161
- нечеткая объектно-ориентированная (НОСС) 162, 163
- расширенная (РСС) 157–160
Семантическая шкала 141
Семантический web (web-2) 128, 129
Семантический анализ 63, 68, 94, 103
Семантический синтез 94
Сенсоризм 246, 247
Сет 42
Синапс 236
Синаптическая карта 238, 239
Синаптический уровень 248
Синсет 83
Синтаксический анализ 63, 68, 81, 82, 93, 94, 268
Синтаксический синтез 93, 94
Система
- интеллектуальная 17, 18
- интеллектуальная автоматизированная (ИАС) 19, 20, 86, 87, 99, 140, 144, 145, 188
- информационная (ИС) 49, 64, 65, 97, 121, 170, 171, 253
- метаданных 121, 123, 124
- оптического чтения текстов (OCR-система) 32, 33, 35, 36, 110
- правил структурного синтеза программ (structural synthesis rules – SSR) 27
- семиотическая 149, 150
- управления базами данных (СУБД) 57, 107, 115, 130, 260
- управления базами знаний (СУБЗ) 198
- управления знаниями (СУЗ) 118, 254–256
- экспертная (ЭС) 28, 63–65, 139, 161, 162, 177, 238
- элементарная сенсорная (ЭСС) 246
- элементарная языковая (ЭЯС) 246
Система машинного перевода 90–93
- И-система 94
- П-система 93
- Т-система 93
Скрытый web (глубинный web) 129, 268
Слот 144, 151, 152
Смысловое соответствие 55–58, 106
Смысловой вес 104–106
Структурно-пятенный эталон 36
Схема метаданных 121

Тезаурус 42, 43, 47–49, 60, 78, 82, 83, 85, 86, 108, 126, 174
Тезаурусные отношения, семантические типы отношений 48, 68, 83, 85, 141, 153, 157, 168, 176, 183, 194–196
Теорема существования решения задачи 22, 24–26
Теория решения изобретательских задач (ТРИЗ) 267, 274, 276
Технология концептуального программмирования (ТКП) 21–23, 28
Типовые задачи инновационной деятельности 266
Типы предикации 166, 167, 169
Триада вещь–свойство–отношение 164
Тримминг 274–276

Указатель 48, 49, 71
Универсальный транслятор описаний теорий (УТОПИСТ) 28
Управление знаниями 179, 253, 254

Феномен вербального мышления 250
Феномены мозга 234
Формализация 204–206, 225, 226, 228
Формальная система (ФС) 147
- закрытая 149
- расширенная 149, 150
Формат метаданных 121
Фрейм 151, 152, 144, 199
Функциональное отношение 24
Функция активации нейрона 238, 243

Хранилище данных 255–259, 261

Шаблон документа 38
Шейп 34

Элементарный фрагмент (ЭФ) 158

Adaptive Pattern Recognition Processing (APRP) 107–112

Computer-Aided Acquisition and Life-cycle Support (CALS) 180
Cross Industry Standard Process for Data Mining (CRISP-DM) 264, 265

Darpa Agent Markup Language (DAML) 188
Data Mining (DM, глубинный анализ данных) 262–264
Document Type Definition (DTD) 130, 131

eXtensible Markup Language (XML) 57, 122, 127–133, 179

Frame Logic (F-logic) 188

GILS 123, 124, 127

HTML-страница (web-страница) 43, 44, 71, 127, 192, 193
Hybrid OLAP (HOLAP) 261
Hypertext Markup Language (HTML) 43, 44, 52, 122, 127, 129, 132, 133, 193
Hypertext Transport Protocol (HTTP) 43

IDEF5 184, 186, 187

Knowledge Interchange Format (KIF) 176, 193

Learning Object Metadata (LOM) 124, 128

Model Driven Architecture (MDA) 154, 155
Model of Open Distributed Processing (ODP) 154, 155
Multidimensional OLAP (MOLAP) 261

On-Line Analytical Processing (OLAP) 258–261
On-Line Transaction Processing (OLTP) 257, 258, 261, 263
Ontolingua 193, 194
Ontology Interchange Language (OIL) 188
Optical Character Recognition (OCR) 32, 110

RDF Schema 122, 131
Relational OLAP (ROLAP) 261
Resource Description Framework (RDF) 122, 123, 128, 131, 179

Simple Object Application Protocol (SOAP) 129, 131
Structured Query Language (SQL) 60, 70, 107, 115, 260

Uniform Resource Identifier (URI) 122, 129, 132, 179
Uniform Resource Locator (URL) 43–45, 129, 132

Web-сайт 43, 71, 75
WordNet 82–86
World Wide Web (WWW) 43, 70, 128, 129, 193
World Wide Web Consortium (W3C) 122, 128, 130

XML Protocol (XMLP) 129, 131
XML Schema 130, 131

Учебное издание

Информатика в техническом университете

Башмаков Александр Игоревич
Башмаков Игорь Александрович

ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Редактор *Н.Е. Овчеренко*
Художники *О.В. Левашова, Н.Г. Столярова, С.С. Водчиц*
Корректор *М.А. Василевская*
Компьютерная верстка *С.Ч. Соколовского*

Оригинал-макет подготовлен в Издательстве МГТУ им. Н.Э. Баумана

Санитарно-эпидемиологическое заключение
№ 77.99.02.953.Д 005683.09.04 от 13.09.2004 г.

Подписано в печать 30.11.04. Формат 70×100/16. Печать офсетная.
Бумага офсетная. Гарнитура «Таймс». Печ. л. 19. Усл. печ. л. 24,7. Уч.-изд. л. 24,13.
Тираж 2000 экз. Заказ 23

Издательство МГТУ им. Н.Э. Баумана.
105005, Москва, 2-я Бауманская, 5.

Отпечатано с оригинал-макета в ГУП ППП «Типография «Наука».
121099, Москва, Шубинский пер., 6.