

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»

**ЛЕКЦИИ ПО КУРСУ
«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ»**

Направление подготовки 01.04.02 Прикладная математика и информатика

Магистерская программа Математическое и программное обеспечение вычислительных машин и компьютерных сетей

Автор: П.Р. Варшавский

Москва

2017

НИУ «МЭИ»

Оглавление

ВВЕДЕНИЕ

ЛЕКЦИЯ 1. ТЕХНОЛОГИИ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ОБРАЗОВ

ЛЕКЦИЯ 2. ТЕХНОЛОГИЯ КОНЦЕПТУАЛЬНОГО ПРОГРАММИРОВАНИЯ

ЛЕКЦИЯ 3. ГИПЕРТЕКСТОВАЯ ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ (ГИТ)

ЛЕКЦИЯ 4. АВТОМАТИЧЕСКОЕ РЕФЕРИРОВАНИЕ И АННОТИРОВАНИЕ. СИСТЕМЫ МАШИННОГО ПЕРЕВОДА

ЛЕКЦИЯ 5. МОДЕЛИРОВАНИЕ ЗНАНИЙ О ПРЕДМЕТНОЙ ОБЛАСТИ

ЛЕКЦИЯ 6. ОНТОЛОГИЧЕСКИЙ ПОДХОД И МЕТАДААННЫЕ

ЛЕКЦИЯ 7. ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ (ИНС)

ЛЕКЦИЯ 8. ИНС. БАЗОВЫЕ ПОНЯТИЯ НЕЙРОТЕХНОЛОГИЙ

ЛЕКЦИЯ 9. ТЕХНОЛОГИЯ БАЗ ЗНАНИЙ (БЗ) И СИСТЕМЫ УПРАВЛЕНИЯ ЗНАНИЯМИ (СУЗ)

Введение

В наше время новые информационные технологии (НИТ), а в них – методы и средства искусственного интеллекта (ИИ), играют особую роль. Чтобы получить представления об основных технологиях ИИ, необходимо изучить, как его важнейшие концепции воплощаются в программных решениях.

Задачи курса:

1. Формирование представления о классах и структуре программного обеспечения интеллектуальных систем (ПО ИС) и об инвариантном ядре таких систем;
2. Представление о методах, математическом аппарате и инструментальных средствах разработки ПО ИС;
3. Основы технологического подхода к разработке ПО ИС.

Аспекты технологического подхода:

1. Обеспечение концептуального единства всех частей программного проекта.
2. Обеспечение интеграции и координации деятельности исполнителей.
3. Совмещение разработки документации с реализацией проекта.
4. Повышение производительности труда программистов.
5. Повышение надежности и качества ПО.
6. Снижение стоимости разработки ПО.
7. Повышение границы сложности создаваемого ПО.

ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ ИС

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 и адаптации.

В настоящее время существует устойчивая тенденция интеллектуализации компьютеров и их 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 Д.А. Пospelовым:

Система считается интеллектуальной, если в ней реализованы следующие три базовые функции:

- 1. Функция представления и обработки знаний.**
- 2. Функция рассуждения.**
- 3. Функция общения.**

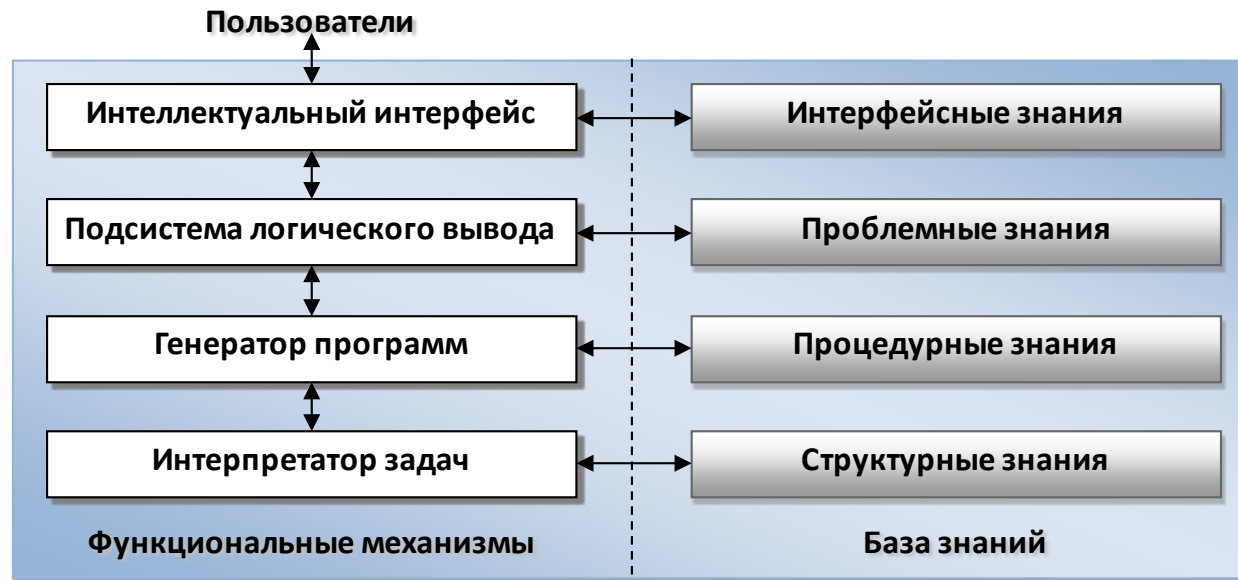


Рис. 1.1. Функциональная модель ИС

В рамках этой функциональной модели (рис. 1.1):

- **интеллектуальный интерфейс** обеспечивает общение с внешней средой и преобразование информации из внешнего во внутреннее представление и обратно;
- **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 ИС при ее функционировании.

СТРУКТУРА ИССЛЕДОВАНИЙ В ОБЛАСТИ ИИ

Попытки определения структуры исследований в области ИИ предпринимались неоднократно. В соответствии с одной из наиболее известных точек зрения по этому вопросу исследования в области ИИ делятся на два базовых направления:

- **бионическое (нейробионическое);**
- **программно-прагматическое.**

В программно-прагматическом направлении выделяют три подхода:

- **локальный (заданный);**
- **системный (основанный на знаниях);**
- **использующий метапроцедуры программирования** для составления интеллектуальных программ по описаниям задач на естественном языке.

СТРУКТУРА ИССЛЕДОВАНИЙ В ОБЛАСТИ ИИ

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
- 1.5. Программы, моделирующие поведение
- 1.6. Программы для анализа и синтеза музыкальных произведений

2. Работа со знаниями

- 2.1. Методы и средства представления знаний
- 2.2. Методы и средства извлечения знаний из различных источников
- 2.3. Методы обработки знаний

3. Интеллектуальное программирование

- 3.1. Языки для интеллектуальных систем
- 3.2. Автоматический синтез программ
- 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

ЛЕКЦИЯ 1. ТЕХНОЛОГИИ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ОБРАЗОВ

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.

Особенности предметной области (ПрО), существенные с точки зрения *OCR*-систем:

- шрифтовое и размерное разнообразие символов;
- искажения в изображениях символов;
- 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.

Автоматическое чтение печатных и рукописных текстов является частным случаем автоматического визуального восприятия сложных изображений. Для решения этой задачи необходимо интеллектуальное распознавание («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.

В современных *OCR*-системах используется технология распознавания (рис. 1.2), свойственная человеку. У человека распознавание образа является многоступенчатым.

Выделяются **три принципа**, на которых основаны все *OCR*-системы (*IPA*):

1. **Принцип целостности образа (*integrity*);**
2. **Принцип целенаправленности (*purposefulness*);**
3. **Принцип адаптивности (*adaptability*).**



Рис. 1.2. Общая схема распознавания текста

Графический образ символа на выходе сканера имеет вид **шейпа** (рис. 1.3), представляющего собой матрицу из точек, которую можно редактировать поэлементно.

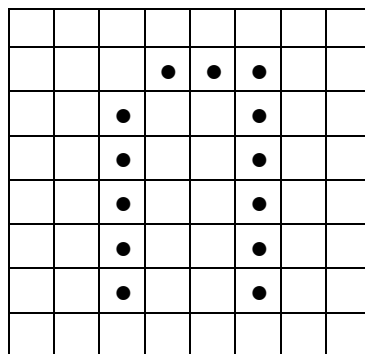


Рис. 1.3. Шейп символа

На рисунке 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, но 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.

ПРИМЕРЫ ПРОГРАММНОЙ РЕАЛИЗАЦИИ OCR-СИСТЕМ

К ведущим российским OCR-системам можно отнести:

- Линейку продуктов *FineReader* фирмы *ABBYY* (<http://www.abbyy.ru>), позволяющих распознавать как печатные, так и рукописные многоязычные тексты;
- Программные продукты фирмы *Cognitive Technologies* (<http://www.cognitive.ru>) OCR-систему *CuneiForm* (<http://www.cuneiform.ru>) и *Cognitive Forms*, предназначенную для массового ввода структурированных документов (например, бухгалтерских и налоговых форм отчетности, платежных документов и т.д.).

Работа системы типа *Fine Reader* включает два крупных этапа:

1. Анализ и предварительная обработка графических изображений;
2. Распознавание отдельных символов.

Системы распознавания реализуются как классификатор. В системах распознавания, построенных на технологиях *ABBYY*, применяются следующие основные **типы классификаторов**:

- растровые (шаблонные);
- признаковые;
- структурные.

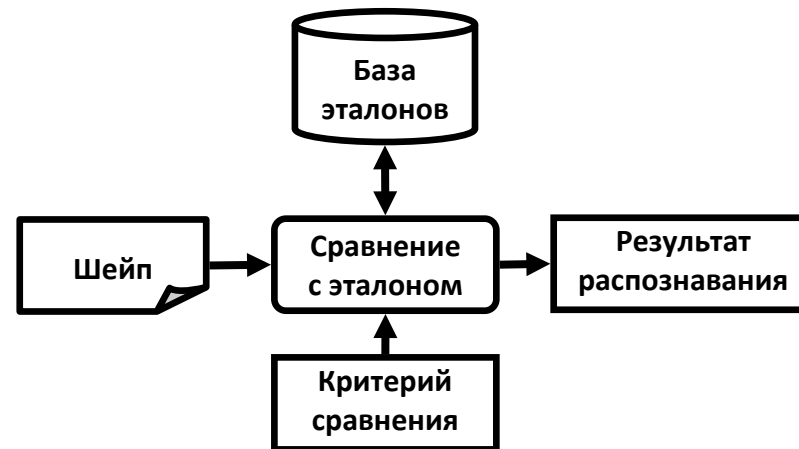


Рис. 1.4. Схема растрового классификатора

Принцип действия растрового классификатора (рис. 1.4) основан на прямом сравнении изображения символа с эталоном. 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 «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 виде графа. Такой способ обеспечивает инвариантность относительно типов и размеров шрифтов. Недостатками являются трудность распознавания дефектных символов и медленная работа.

В **ABBYY FineReader** применяется так называемый **структурно-пятенный эталон** (рис. 1.5) и его фонтанное (от англ. *font* – шрифт) представление, которое имеет вид набора пятен с попарными отношениями между ними. Данная разработка фирмы **ABBYY** первоначально использовалась для распознавания рукописного текста, а затем была успешно применена и для обработки печатных символов. При этом обеспечиваются все достоинства шаблонного и структурного классификаторов. Также данное представление нечувствительно к различным начертаниям и дефектам символов.

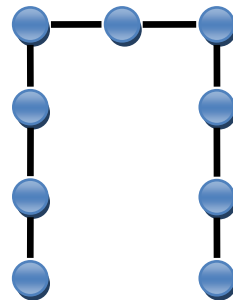


Рис. 1.5. Структурно-пятенный эталон

В современных *OCR*-системах обычно используются все три типа классификаторов, но основным является структурный. Два других для ускорения и повышения качества распознавания (рис. 1.6).

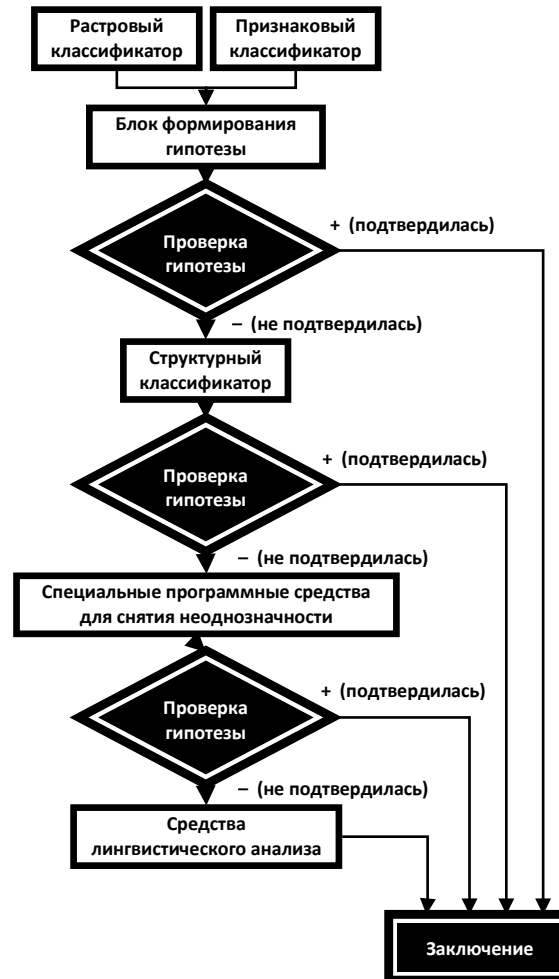


Рис. 1.6. Укрупненная схема работы системы Fine Reader

Особенности распознавания рукописных текстов:

- использование структурно-пятенного эталона с учетом особенностей траектории движения пишущего инструмента;
- основным механизмом является выдвижение и подтверждение гипотез;
- использование методов оптимизации при управлении перебором вариантов.

OCR-система *Cognitive Forms* представляет собой программный комплекс для массового ввода документов, имеющих стандартизованные формы.

Технология ввода документов в стандартизованных формах включает две стадии: **подготовительную, 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.

Основная стадия состоит из шести этапов (рис. 1.7).

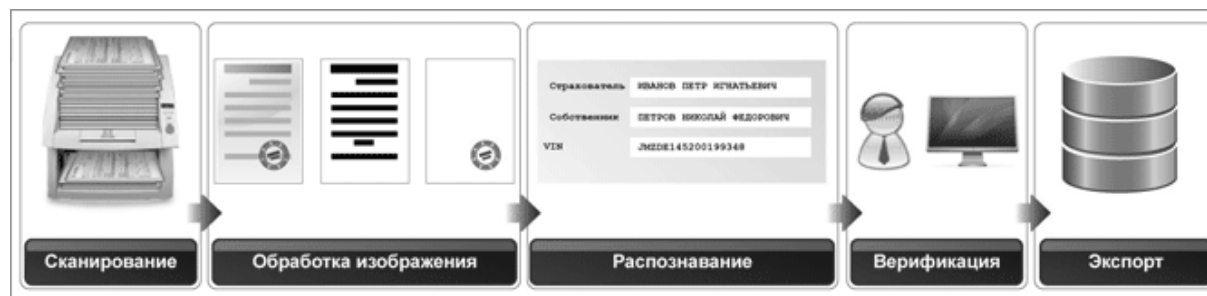


Рис. 1.7. Этапы основной стадии ввода документов в стандартизованных формах

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 типы классификаторов используются в OCR-системах? Какие достоинства и недостатки присущи классификаторам каждого типа?
7. Что такое структурно-пятенный эталон?
8. В чем заключаются особенности распознавания рукописных текстов?

ЛЕКЦИЯ 2. ТЕХНОЛОГИЯ КОНЦЕПТУАЛЬНОГО ПРОГРАММИРОВАНИЯ

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.

ТКП предназначена для **синтеза программ решения задач по их описанию на ограниченном естественном языке (ОЕЯ) при некоторых ограничениях.**

Эти ограничения требуют:

- 1) **точного указания ПрО, к которой относится решаемая задача;**
- 2) **фиксации класса решаемых задач.**

Последние получили название **вычислительных** или **расчетно-логических задач.** В общем случае их описание на ОЕЯ имеет вид:

$$\text{Зная } M, \text{ вычислить } (y_1, \dots, y_n) \text{ по } (x_1, \dots, x_m) \quad (2.1)$$

В выражении (1.1) M идентифицирует ПрО (например, тригонометрию, кинематику и т.д.).

Кортеж (x_1, \dots, x_m) содержит идентификаторы переменных с известными значениями, а кортеж (y_1, \dots, y_n) – идентификаторы переменных, значения которых требуется определить.

Существенным ограничением ТКП является предположение, что в компьютере имеется модель ПрО (МПрО), с которой можно манипулировать.

В ТКП для представления МПрО используются **семантические сети специального вида**, называемые **вычислительными моделями (ВМ)**.

Известны четыре **подхода к синтезу программ**:

- 1) **дедуктивный**;
- 2) **индуктивный**;
- 3) **трансформационный**;
- 4) **утилитарный**.

В ТКП используются **первые два подхода**. Основная идея ТКП состоит в следующем:

Пусть существует постановка задачи в виде (2.1).

Необходимо:

- 1) **перейти от (2.1) к теореме существования решения данной задачи**;
- 2) **построить доказательство теоремы существования**;
- 3) **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**.

Концептуализацией называется процесс перехода от описания ПрО на ОЕЯ к точной спецификации этого описания на некотором формальном языке, ориентированном на компьютерное представление.

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

$$S=(O, R) = (\{o_i \mid i = 1,2,\dots,k\}, \{r_i \mid j = 1,2,\dots,l\}) \quad (2.2)$$

где O – множество объектов ПрО ($|O|=k$); R – множество отношений между объектами ПрО ($|R|=l$).

ВМ для заданной ПрО определяется как кортеж:

$$(\{p_i\}, \{f_j\}, \{u_k\}) \quad (2.3)$$

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

Функциональное отношение f_j задается тройкой **плекс-элементов**.

$$f_j = (X_j, F_j, Y_j) \quad (2.4)$$

где $X_j = (x_{j1}, \dots, x_{jnj})$ — набор входных переменных для f_j ; F_j — ссылка на процедуру, реализующую вычисление $Y_j = F_j(X_j)$; $Y_j = (y_{j1}, \dots, y_{jnj})$ — набор выходных переменных для f_j .

Управляющие структуры u_k реализуют отображения X_j и Y_j в множество разрешенных типов данных, а также они позволяют приписывать переменным известные и вычисленные значения.

Графически концептуализация ПрО в рамках ВМ изображается графом G :

$$G = (V, U) = (\{x_i\} \cup \{y_j\} \cup \{F_l\}, \{u_k\}) \quad (2.5)$$

Процесс доказательства теоремы существования решения задачи (2.1) отображается на **графе** как **«волновой процесс»**, начинающийся в вершинах (x_1, \dots, x_m) и заканчивающийся, когда «волна» достигнет всех вершин (y_1, \dots, y_n) . При волновой интерпретации можно детализировать постановку задачи (2.1) и выделить четыре класса задач:

1. **Задачи на доказательство.**
2. **Задачи на вычисление значений переменных.**
3. **Задачи на прогнозирование.**
4. **Задачи планирования эксперимента.**

Рассмотрим *теорему существования решения задачи* в постановке (2.1). Обозначим $P(x)$ предикат входных условий, а $R(x, y)$ — предикат выходных условий; $x = (x_1, \dots, x_m)$, $y = (y_1, \dots, y_n)$.

Запишем теорему существования в виде

$$\forall x (P(x) \Rightarrow \exists y R(x, y)) \quad (2.6)$$

Будем рассматривать только конструктивные логические теории. Н.Н. Непейвода доказал, что различные определения реализуемости эквивалентны. Он же показал, что существует реализуемость, при которой формулам вида $\exists y R(y)$ будет соответствовать либо программа вычисления y , либо само значение y . Тогда **любой доказуемой формуле будет соответствовать программа.**

Предполагается, что реализации всех аксиом заданы априорно. Для каждого правила вывода

$$\text{П } \frac{A_1, \dots, A_k}{A} \quad (\text{или просто } \frac{A_1, \dots, A_k}{A})$$

заданы правила построения реализации выводимой по этому правилу формулы A по реализациям формул A_1, \dots, 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.

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.

Каждый шаг доказательства преобразуется во фрагмент программы отдельно от других шагов. Построенные таким образом программы являются хорошо структурированными. Существуют два способа извлечения программы из доказательства: При первом реализации формул используются непосредственно, поэтому программой является реализация теоремы существования решения. Второй способ извлечения программы заключается в составлении ее оператор за оператором из шагов доказательства теоремы существования (линейный вывод). В этом случае программа состоит из операторов присваивания и операторов вызова процедур.

Рассмотренная система правил вывода не содержала правил для индукции, поэтому в программах не было циклов, но применяя разные схемы индукции, можно получить разные схемы циклов.

ИНСТРУМЕНТАРИЙ КОНЦЕПТУАЛЬНОГО ПРОГРАММИРОВАНИЯ

ТКП 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. На рис. 2.1 представлена укрупненная схема решения задачи в ПРИЗ.

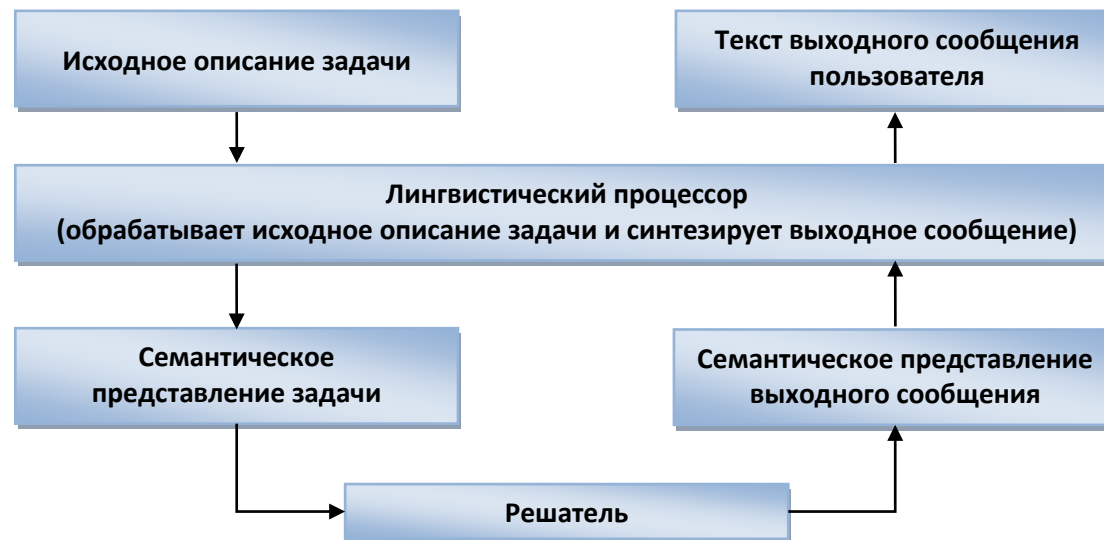


Рис. 2.1. Схема решения задачи в ПРИЗ

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. Запрос на решение задачи содержит перечень искомым параметров объектов.

Вопросы для самопроверки

1. Каково назначение ТКП?
2. Назовите подходы к синтезу программ.
3. В чем состоит основная идея ТКП?
4. Дайте определение понятия «концептуализация».
5. Что понимается под вычислительными моделями и как они описываются?
6. Сформулируйте теорему существования решения задачи в ТКП.
7. Какие способы извлечения программы решения задачи из доказательства теоремы его существования Вы знаете?
8. Назовите программные реализации ТКП.

ЛЕКЦИЯ 3. ГИПЕРТЕКСТОВАЯ ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ (ГИТ)

Текст является универсальным средством представления, накопления и передачи знаний в человеческом обществе.

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.

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

ОСНОВНЫЕ ПОНЯТИЯ ГИТ

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, ГТ можно читать, **двигаясь по разным траекториям**, образованным связанными сетями.
3. Активируемые переходы выбираются читателем (пользователем). Имена (типы) связей облегчают решение задачи выбора перехода.

ГТ документ может быть как **электронным**, так и **бумажным**. Однако в полной мере функциональность ГТ реализуется лишь в электронных гипертекстовых документах.

В ГТ документе может быть представлено несколько **уровней детализации материала**. Такие документы моделируются деревьями или сетями.

Таким образом, ГТ как информационная модель интегрирует положительные стороны энциклопедий, монографий и тезаурусов.

От **энциклопедий** ГТ наследует:

- детальное представление понятий;
- быстрый просмотр материала;
- алфавитный поиск.

От **монографий**:

- представление материала с разной степенью глубины и детальности;
- поиск 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.

Проблемы и задачи, связанные с ГИТ:

- модели ГТ (формализованная и условно-типовая);
- инструментальные средства для создания ГТ;
- гипертекстовые информационно-поисковые системы (ГИПС);
- методы извлечения знаний для гипертекстовых систем;
- автоматизация построения ГТ;
- место ГИТ среди технологий ИИ.

Области применения ГИТ:

- 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.

ФОРМАЛИЗОВАННАЯ МОДЕЛЬ ГИПЕРТЕКСТА (ФМГТ)

В основе моделей ГТ лежит понятие **информационно-справочной статьи (ИСС)**, выступающей в качестве информационной единицы ГТ. В конкретных технологиях **ИСС называют** по-разному: **страница, статья, тема.**

Элементом ИСС могут быть присвоены метки, уникальные в рамках ИСС. Кроме того, эти элементы могут наделяться интерактивным поведением. Такие элементы называются **гиперссылками**, которые могут быть локальными и глобальными.

С точки зрения программной реализации формализованная модель ГТ состоит из двух слоев. Первый слой представляет отображаемое на экране содержимое документа, а адреса переходов хранятся во втором, скрытом слое модели.

$$\text{ФМГТ} = (x_0, x_1, \dots, x_{11}), \quad (3.1)$$

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 (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.

В 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 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 .

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.

УСЛОВНО-ТИПОВАЯ МОДЕЛЬ ГИПЕРТЕКСТА (УТМГТ)

Один из недостатков ФМГТ связан с отсутствием в ней возможности явного определения типов гиперссылок. В УТМГТ 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.

- 1) **Тезаурус** — упорядоченный перечень терминов, в котором отражены семантические отношения между ними.
- 2) **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 в УТМГТ 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 лингвистике выделено около **200 семантических типов отношений**. Наиболее часто употребляются **10 типов**, используемых в УТМГТ и приведенных в таблице 3.1.

Таблица 3.1.

Часто употребляемые типы семантических отношений

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

Графовой интерпретацией УТМГТ является **семантическая сеть**. В рамках УТМГТ ИСС включает имя, заголовок, собственно текст (содержимое) и список ссылок на ИСС, связанные с данной ИСС различными типами отношений. Такой список ссылок образует **локальный справочный аппарат ИСС**.

Он может быть организован тремя способами: **в виде списка; внедрение ссылок в текст; комбинированным.**

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ДЛЯ СОЗДАНИЯ ГИПЕРТЕКСТА

Существует большое число инструментальных средств для создания ГТ и различных форматов, включая *PDF (Portable Document Format)*, *RTF (Rich Text Format)*, *DOC (Document Word)* и *WinHelp (Windows Help)*, *CHM (Compiled HTML)*, а также целое семейство языков гипертекстовой разметки, самыми популярными из которых можно считать *HTML (Hypertext Markup Language)* и *XML (eXtensible Markup Language)*. 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.

Создание гипертекстового справочника по программному продукту состоит из следующих этапов:

1. Определение структуры справочника и его разделов;
2. Подготовка текста и графических иллюстраций справочника;
3. Создание файла проекта справочника;
4. Компиляция исходных файлов тем (*topics*, ИСС), 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 трудно формализуемым и сложным. В рамках него специфицируются:

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

WinHelp и **HTML Help** представляют собой стандартные технологии построения и работы с гипертекстовыми справочниками для платформы *Windows*.

Гипертекст в формате WinHelp реализуется в виде файла с расширением *HLP* (*help*-файла). *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*. *CHM*-файл формируется на основе файлов в формате *HTML* с помощью специального компилятора. Для вызова справочника из приложения служит функция *HTML Help API HtmlHelp()*.

Гипертекст может быть разработан с помощью различных инструментальных средств. Наиболее популярными из них являются:

- **HTML Help Workshop** фирмы **Microsoft**;
- **HelpScribble**;
- **KeyTools** фирмы **KeyWorks Software**;
- Система **AnetHelp Tool** российской фирмы **Anet Soft**;
- **RoboHelp**;
- Подключаемый модуль **Mif2GO**;
- Система **Help & Manual**.

ГИПЕРТЕКСТОВЫЕ ИНФОРМАЦИОННО-ПОИСКОВЫЕ СИСТЕМЫ

ГИТ 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).

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)**.

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 данного поиска является **полнотекстовый поиск**, реализуемый, например, в поисковых машинах *Internet*.

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.

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.

Существуют различные методы информационного поиска в ИПС (рис. 3.1).

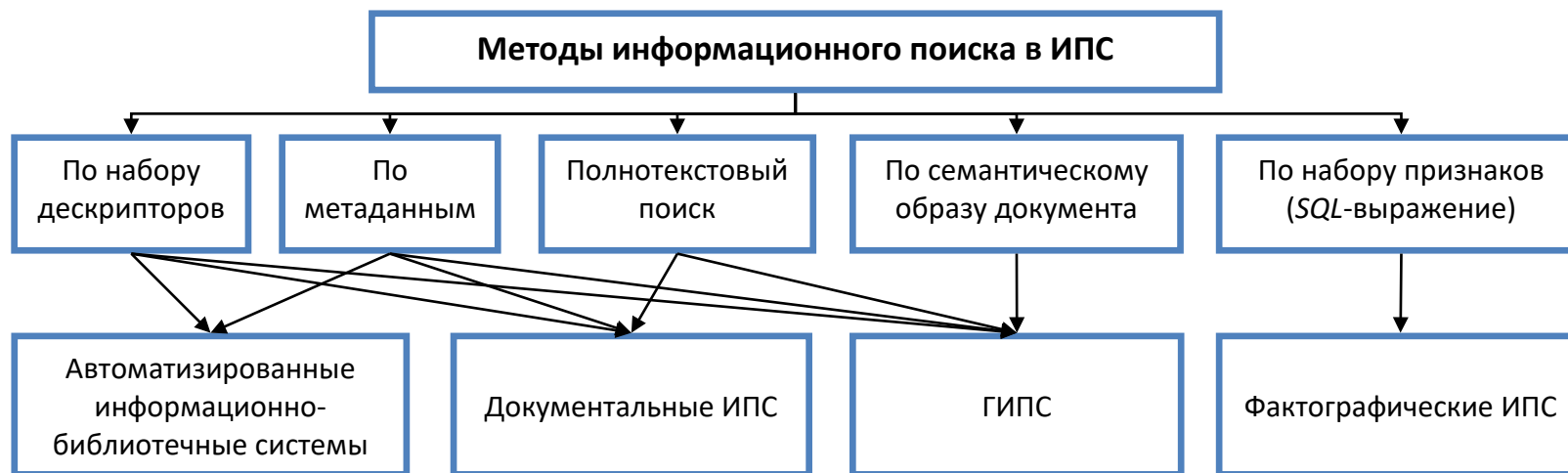


Рис. 3.1. Классификация методов информационного поиска в ИПС

Введем следующие обозначения: D — множество документов в информационном хранилище, $d_i \in D$ — i -й документ, $D_j \subseteq D$ — подмножество документов.

Зададим на D оценку смысловой близости пары документов $r(d_i, d_j) \geq 0$.

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: $S=(S_1, S_2, \dots, S_k)$, $k > 0$. Чем больше значение оценки, 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 $(D_j \subseteq D) \mid r(z, d_i \in D_j) \rightarrow \min$. Если $D_j = \emptyset$, 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 $(D_j \subseteq D) \mid r(z, d_i \in D_j) \leq \Delta$, где Δ — оценка наибольшего допустимого расхождения смыслов запроса и искомым документам;
3. Найти $(D_j \subseteq D) \mid S_f(d_i \in D_j) \rightarrow \max$. Результатом поиска служит подмножество документов, которым приписана наибольшая оценка важности. Обобщением этого варианта является векторный поиск, учитывающий оценки нескольких свойств;
4. Комбинированный поиск: Найти $(D_j \subseteq D) \mid r(z, d_i \in D_j) \leq \Delta \ \& \ S_k(d_i \in D_j) \rightarrow \max$.

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

Эффективность информационного поиска документов, обеспечиваемая ИПС, оценивается по двум показателям:

k_{Π} – коэффициент информационной полноты;

$k_{\text{ш}}$ – коэффициент информационного шума.

Коэффициенты k_{Π} и $k_{\text{ш}}$ принимают значения в интервале от 0 до 1. В некоторых источниках эти коэффициенты выражают в процентах.

Определим коэффициенты полноты и шума:

$$k_{\Pi} = \lim_{k \rightarrow m} \frac{1}{k} \sum_{i=1}^k \frac{|D_i \cap D_i^0|}{|D_i|}, \quad (3.2)$$

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

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

Эффективность информационного поиска E_1 выражается через коэффициенты $k_{\text{ш}}$ и k_{Π} , что позволяет рассматривать ее в качестве **интегрального показателя эффективности информационного поиска ИПС.**

В литературе в функции $E_1(k_{ш}, k_{п})$ вместо $k_{ш}$ принято использовать обратный ему показатель — **коэффициент точности $k_{т}$** .

$$k_{т} = 1 - k_{ш} \quad (3.4)$$

Таким образом, запишем данную функцию в виде:

$$E_1 = \frac{2k_{т}k_{п}}{k_{т} + k_{п}}. \quad (3.5)$$

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

$$E_{\beta} = \frac{(\beta^2 + 1)k_{т}k_{п}}{\beta^2k_{т} + k_{п}}, \quad (3.6)$$

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.

КОМПЬЮТЕРНЫЕ МЕТОДЫ ПОИСКА В ТЕКСТЕ

Методы поиска в тексте, используемые человеком:

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

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);
- 4) семантические методы поиска, использующие подходы ИИ.

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*) и др.

Каталог ресурсов *Internet* — постоянно обновляемая и пополняемая система ссылок на ресурсы, распределенные по иерархической структуре категорий. Каталоги облегчают поиск за счет упорядоченности ссылок на ресурсы. Все интеллектуальные функции остаются за человеком.

Подборки ссылок на информационные ресурсы *Internet* 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.

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

Поисковые машины (или поисковые системы) позволяют находить ресурсы *Internet* непосредственно по их текстовому содержанию.

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* — www.searchenginecolossus.com) зарегистрировано свыше 3500 систем из 312 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:

- *Google* (<http://www.google.com.ru>);
- *AltaVista* (<http://www.altavista.com>);
- *Yahoo!* (<http://www.yahoo.com>);
- *Excite* (<http://www.excite.com>);
- *HotBot* (<http://www.hotbot.com>);
- *Lycos* (<http://www.lycos.com>);
- *AOL* (<http://search.aol.com>);
- *MSN* (<http://search.msn.com>);
- *Yandex* (<http://www.yandex.ru>);
- *Rambler* (<http://www.rambler.ru>);
- *Anopm* (<http://www.aport.ru>);
- *Rundex* (<http://www.rundex.ru>).

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

Главными компонентами типовой поисковой машины (рис. 3.2) являются:

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

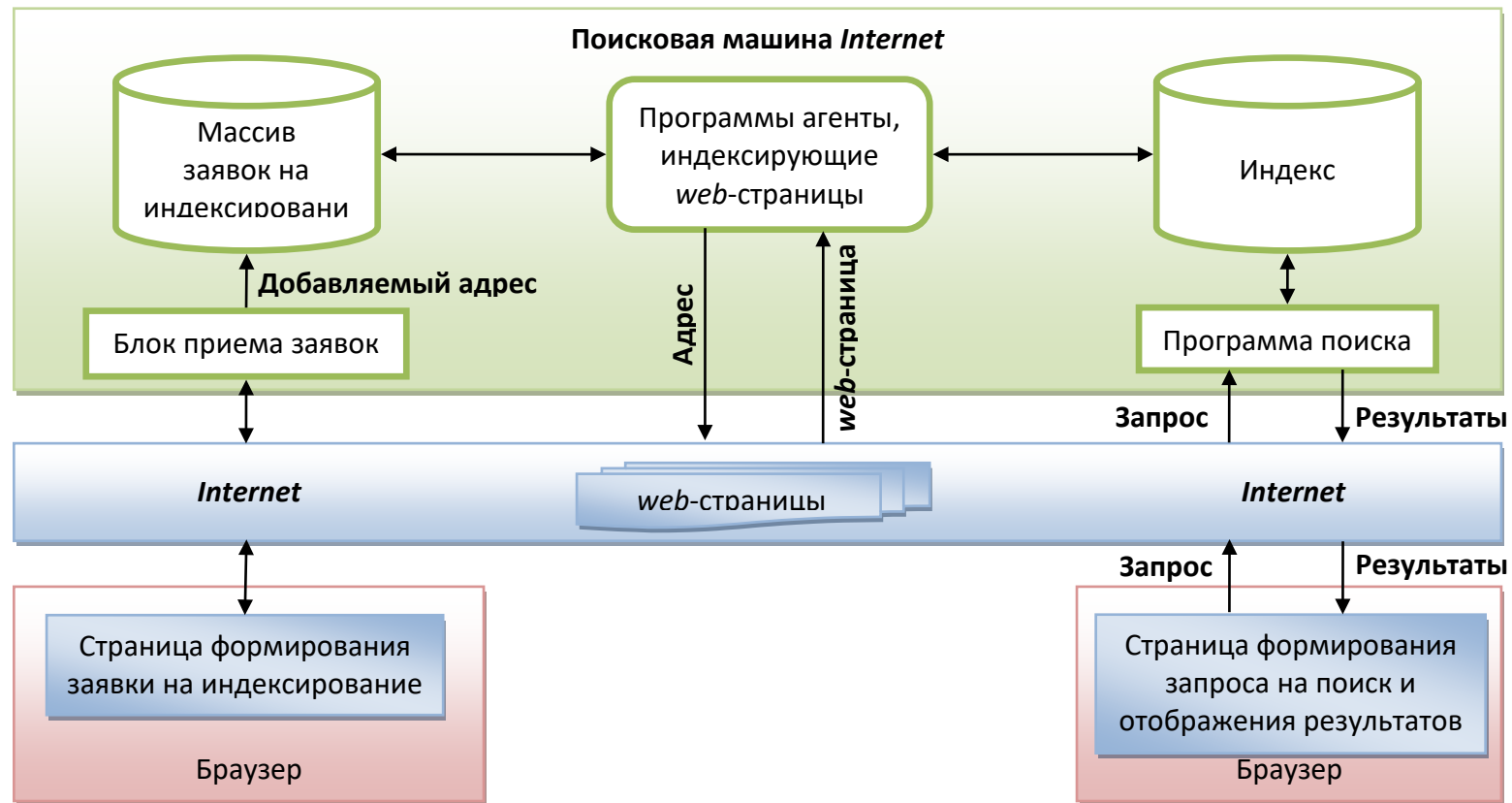


Рис. 3.2. Упрощенная структура типовой поисковой машины

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.

Одной из проблем является реализация алгоритма перемещения (навигации) по сети. Предпочтительным для индексирования web-ресурсов принят метод, который осуществляет сначала навигацию вширь, а затем вглубь (это подтверждается статистикой работы поисковых машин).

Разновидностями агентов являются:

- **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.

МЕТОДЫ ИЗВЛЕЧЕНИЯ ЗНАНИЙ ДЛЯ ПОСТРОЕНИЯ ГИПЕРТЕКСТА

Существуют два класса **источников знаний**:

- 1. Эксперты** (специалисты в ПрО, для которой формируется ГТ);
- 2. 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.
- 1.1.2. Запись и анализ лекций.
- 1.1.3. Запись и анализ вербальных отчетов.

1.2. Активные методы.

- 1.2.1. Работа с группой экспертов.
 - 1.2.1.1. Метод «мозгового штурма».
 - 1.2.1.2. Метод «круглого стола».
 - 1.2.1.3. Ролевые игры.
- 1.2.2. Индивидуальная работа с экспертом.
 - 1.2.2.1. Анкетирование.
 - 1.2.2.2. Интервьюирование.
 - 1.2.2.3. Свободный диалог.
 - 1.2.2.4. Исследовательская игра с одним экспертом.

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.

АВТОМАТИЗАЦИЯ ПОСТРОЕНИЯ ГИПЕРТЕКСТА

Ручное формирование ГТ на основе объемного текстового материала весьма трудоемкий процесс.

Для упрощения формирования ГТ 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»).

МЕСТО ГИТ СРЕДИ ТЕХНОЛОГИЙ ИИ

Основоположником гипертекстового подхода принято считать **Ванневара Буша**. 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**, под руководством которого была создана первая гипертекстовая система **Xanadu**. Первые коммерческие гипертекстовые 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. Гипертекстовое представление информации соответствует ассоциативному характеру мышления человека, способствует осознанию целей читателя, обеспечивает высокую степень свободы его мышления.

ГИТ базируется на основных парадигмах ИИ:

- **использовании БЗ;**
- **логическом выводе;**
- **общении с пользователем на ОЕЯ.**

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.

Вопросы для самопроверки

1. Какие идеи лежат в основе ГТ?
2. Назовите основные области применения ГИТ?
3. Как ГИТ используется в Internet?
4. Охарактеризуйте формализованную модель ГТ.
5. Опишите условно-типовую модель ГТ.
6. Дайте определение понятия «тезаурус».
7. Какой компонент условно-типовой модели ГТ представляет семантические отношения ИСС?
8. Назовите основные классы ИПС.
9. Что такое поисковый образ, поисковое предписание, дескриптор, индексирование, индекс?
10. Для чего предназначен критерий релевантности?
11. Какие существуют методы информационного поиска в ИПС?
12. В чем состоят особенности полнотекстового поиска по сравнению с традиционным поиском по дескрипторам?
13. Назовите показатели эффективности информационного поиска документов в ИПС. Поясните их смысл.
14. Сравните основные характеристики ИПС разных классов.

ЛЕКЦИЯ 4. АВТОМАТИЧЕСКОЕ РЕФЕРИРОВАНИЕ И АННОТИРОВАНИЕ. СИСТЕМЫ МАШИННОГО ПЕРЕВОДА

Рефератом называют:

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

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.

Автоматическое 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** (*Natural Language Processing, NLP*-системы). И в этом качестве оно относится к фундаментальным технологиям ИИ.

Основные тенденции для данной области:

- аннотированные каталоги перерастают в гипертекстовые;
- на всех крупных сайтах *Internet* предусматривают оглавления (*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*; 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. **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* (функция автоматического реферирования);
- ОРФО 5.0 (компания «Информатик»), включающую функцию автоматического аннотирования;
- «Либретто» (компания «МедиаЛингва»);
- Программный пакет «МедиаЛингва Аннотатор *SDK 1.0*»;
- Поисковую систему «Следопыт» со средствами автоматического реферирования и аннотирования;
- *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. Источниками информации для рефератов и аннотаций могут служить не только тексты, но и видеозаписи, разнообразные табличные документы и т.д.

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;
2. информационные;
3. критические (обзоры).

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. Первый из них не предполагает опоры на знания, связанные с текстом на ЕЯ. В системах такого типа применяется универсальная база правил, не зависящая от ПрО и языка текста. Вторым подходом 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** (рис. 4.1) 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 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 $V(U) = \alpha_1 k_1 + \alpha_2 k_2 + \alpha_3 k_3 + \alpha_4 k_4$. По коэффициентам важности выполняется отбор блоков в реферат.

Для вычисления каждого весового коэффициента используется своя группа правил. Для k_1 они учитывают расположение блока. Для k_2 правила учитывают результаты автоматической индексации документа. Для k_3 учитывается наличие в блоке таких ключевых фраз и выражений, как «в заключение...», «согласно результатам анализа...», «отличный от...», «малозначущий...» и т.п. Для k_4 правила учитывают вхождение термина в заголовки, колонтитулы, первый параграф текста, пользовательский профиль запроса и т.п.

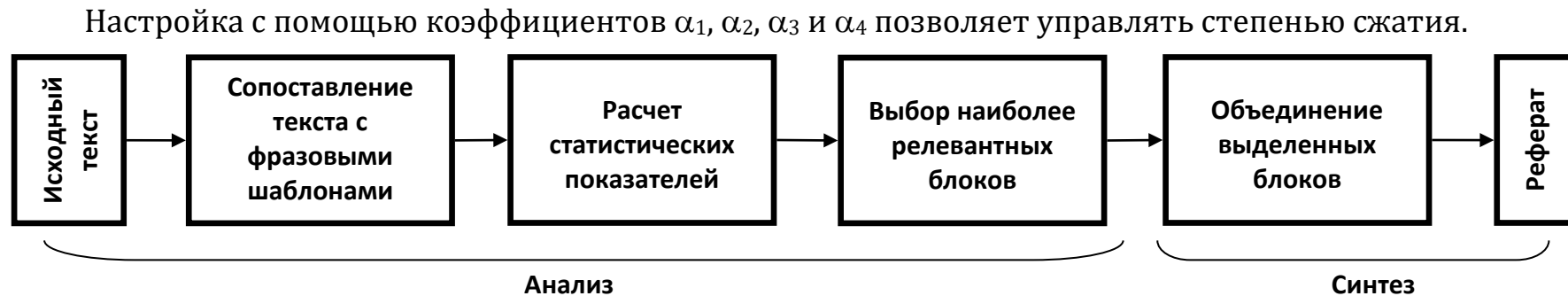


Рис. 4.1. Обобщенная архитектура системы автоматического реферирования первого типа

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.

Человеку, уловившему общий смысл информации, легче выделить главное и кратко изложить содержание. Это и обуславливает создание **реферирующих систем второго типа** (рис. 4.2). 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.

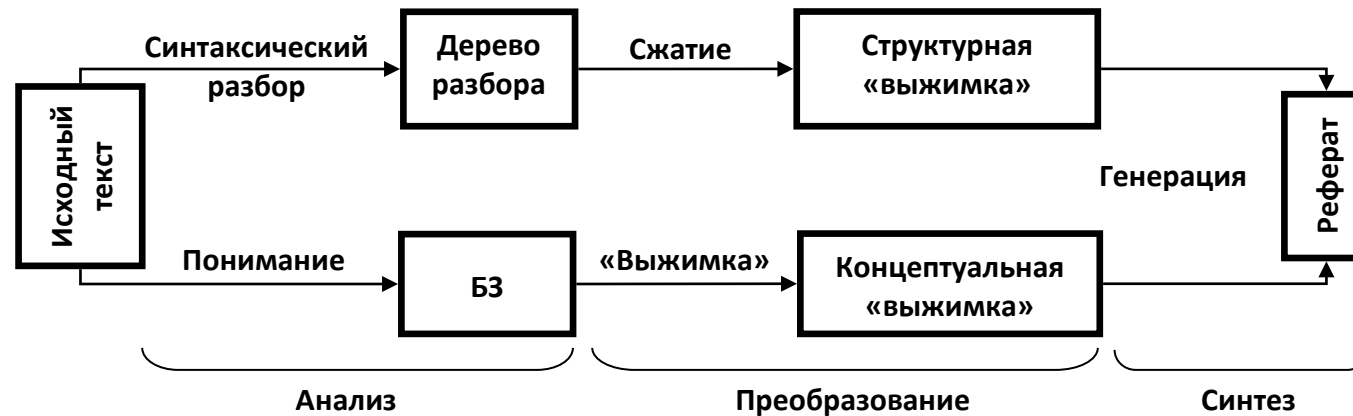


Рис. 4.2. Основные подходы к формированию реферата в системах с опорой на знания

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

Для функционирования подобных систем необходимы:

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

Отметим следующие задачи, связанные с компьютерным реферированием:

1. Создание одноязычных рефератов из источников на разных языках.
2. Построение рефератов по гибридным источникам, включающим как текстовые, так и числовые данные в разных формах (таблицы, диаграммы, графики и т.д.).
3. Создание рефератов на основе массивов документов. Например, построение единого реферата по сборнику тезисов докладов научной конференции. Одна из областей применения подобных средств — формирование новостных сообщений по газетным источникам.
4. Растущий объем мультимедийной информации обуславливает 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.

СИСТЕМЫ МАШИННОГО ПЕРЕВОДА

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 ИС:

- все возрастающая потребность в переводах в науке, литературе, дипломатии, экономике и других областях деятельности, обуславливаемая 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 **двухязычные** и **многоязычные системы МП**.

Язык исходного текста называется **входным**, а **язык перевода** (формируемого текста) — **выходным**.

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.**

Такие системы 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.).

Системы МП бывают **автоматическими** и **автоматизированными**.

Автоматизированные системы МП реализуют **три схемы работы:**

- с постредктированием;
- с предредктированием;
- с пред- и постредктированием.

Выполняя перевод, человек уясняет смысл очередного фрагмента текста (фразы, абзаца) и выражает его на выходном языке, стараясь обеспечить структурную и смысловую близость к оригиналу.

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:

- 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 этапе выполняется морфологический анализ входной фразы.**
- На втором этапе выполняется перевод морфологического представления входной фразы в морфологическое представление выходной фразы.**
- На третьем этапе выполняется морфологический синтез.**

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 этапе осуществляется морфологический анализ входной фразы (аналогично П-системам).**

- **На втором этапе** по его результатам выполняется **синтаксический анализ**.
- **На третьем этапе** выполняется **переход от входного к выходному языку**.

Выделяются три уровня преобразования:

- **поверхностно-синтаксический;**
 - **глубинно-синтаксический;**
 - **синтактико-семантический.**
- **На четвертом этапе** проводится **синтаксический синтез**.
 - **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** (рис. 5.1). Его результатом служат семантические представления входной и выходной фраз, эквивалентные с точностью до лексики.

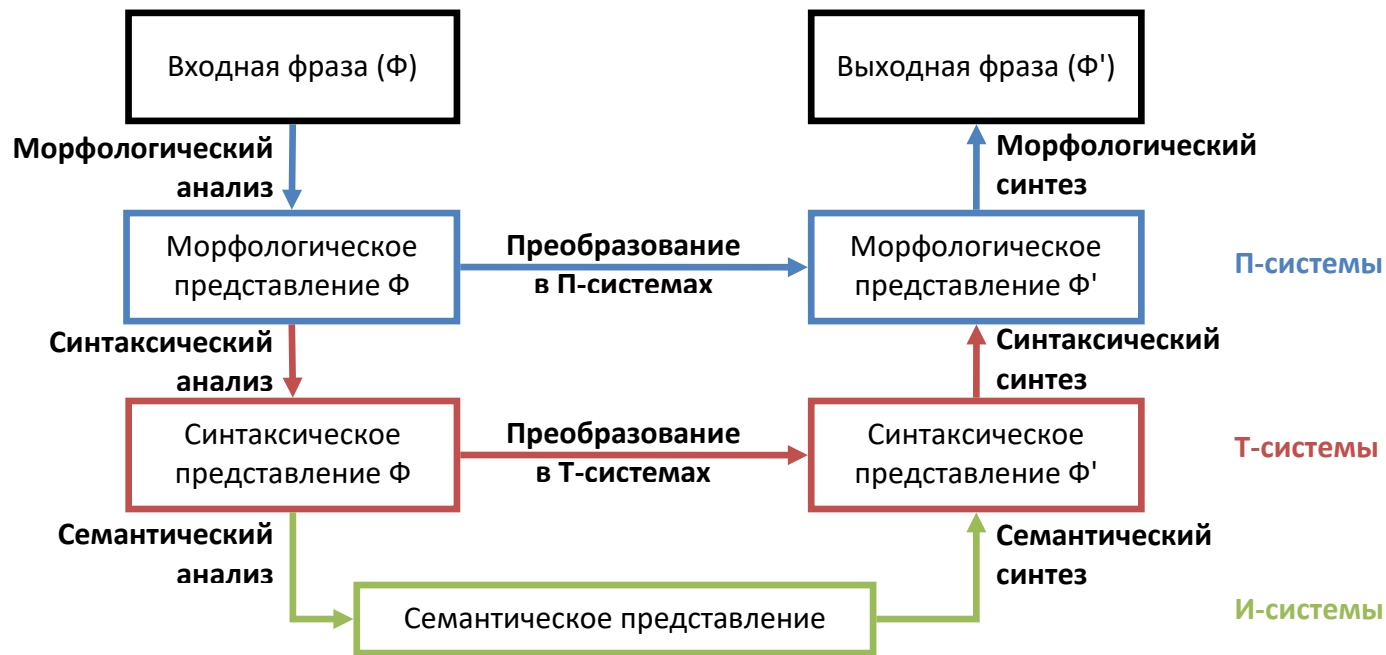


Рис. 4.3. Отношения между этапами функционирования трех поколений систем МП

Таким образом, системы МП представляют собой сложные программные комплексы с разными видами обеспечений.

К лингвистическому обеспечению (ЛО) систем МП относятся:

- словари слов и словосочетаний с соответствующими признаками;
- морфологические таблицы суффиксов и окончаний;
- базы грамматических правил и др.

К числу наиболее интересных проектов по разработке ЛО относится **WordNet** — открытая справочная лексическая система, представляющая тезаурус английского языка. Данный проект выполняется с начала 90 годов в лаборатории когнитологии Принстонского университета под руководством профессора Дж. А. Миллера.

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).

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 .

Web-интерфейс для работы с сетевой версией тезауруса доступен по адресу <http://www.cogsci.princeton.edu/cgi-bin/webwn>. **WordNet** является бесплатным, свободно распространяемым продуктом и может использоваться как в исходном, так и модифицированном виде в коммерческих приложениях.

С проектом *WordNet* связан ряд проектов, направленных на расширение модели и программных средств *WordNet*, интеграцией компонентов *WordNet* в ИС, созданием интерфейсов для доступа к информационной базе *WordNet* из приложений, основанных на различных технологиях и программных платформах, построением тезаурусов типа *WordNet* других ЕЯ (<http://www.globalwordnet.org>).

Интерактивный графический интерфейс для взаимодействия с тезаурусом *WordNet* реализован в системе *Visual Thesaurus* (www.visualthesaurus.com), разработанной фирмой *Plumb Design* (www.plumbdesign.com). Система формирует двухмерное или трехмерное представление графа тезауруса.

Математическое обеспечение систем МП включает:

- модели для представления лингвистической информации;
- алгоритмы их преобразования;
- правила логического вывода для уточнения обрабатываемого текста на основе экстралингвистических знаний.

К программному обеспечению систем МП относятся:

- программы выполнения перевода;
- ведения словарей;
- формирования базы правил и т.д.

Информационное обеспечение (ИО) систем МП 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;
- *On-line* переводчик *Google* (<http://translate.google.com/>);
- *Socrat* - система, позволяющая сканировать документы, переводить их содержимое и проверять орфографию;
- *Polyglossum* - многоязычная система МП с широким набором предметных словарей;
- *Prompt* - многоязычная система МП, содержащая множество словарей по разным Про;
- *WebTranSite* - система для перевода *web*-страниц;
- *Lingvo* - компьютерный англо-русский и русско-английский словарь.

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 требования предъявляются к реферату?
5. Перечислите виды рефератов.
6. Каковы основные идеи метода составления выдержек?
7. Охарактеризуйте модель линейных весовых коэффициентов. Каковы ее достоинства и недостатки?
8. Какие подходы реализуются в системах автоматического реферирования, основанных на знаниях?
9. Как классифицируются системы МП?
10. Какие схемы обработки текста используются при автоматизированном МП?

11. Чем различаются П-, Т- и И-системы МП?
12. Что такое экстралингвистические знания, и как они используются в системах МП?
13. Почему МП относят к технологиям ИИ?
14. Каковы перспективы систем МП?
15. Что такое синсет?
16. Охарактеризуйте математическую модель тезауруса *WordNet*.

ЛЕКЦИЯ 5. МОДЕЛИРОВАНИЕ ЗНАНИЙ О ПРЕДМЕТНОЙ ОБЛАСТИ

В основе исследований в области ИИ лежит подход, связанный со **знаниями**. Понятие «**знание**» относится к интуитивно определяемым.

В БСЭ дается следующее его толкование:

«Знание — проверенный практикой результат познания действительности, верное её отражение в сознании человека. Знания бывают житейскими, донаучными, художественными, научными (теоретическими и эмпирическими)».

Разновидности знаний:

- декларативные;
- прагматические;
- процедурные;
- эвристические;
- экспертные;
- знания о ПрО.

Выделяют две характеристики знаний: **объектность** и **личностность**.

Трактовки знаний могут быть объединены в четыре группы:

- психологическую;
- интеллектуальную;
- формально-логическую;
- информационно-технологическую.

КЛАССИФИКАЦИИ ЗНАНИЙ

В зависимости от источника знания можно разделить на две категории:

- **априорные знания** (определяются и закладываются в БЗ до начала функционирования ИС);
- **накапливаемые знания** (формируются в процессе использования БЗ):
 - **экспертные знания;**
 - **наблюдаемые знания;**
 - **выводимые знания.**

По характеру использования при решении задач в определенной Про выделяют следующие виды знаний:

- **декларативные** (факты);
- **процедурные;**
- **метазнания** («знания о свойствах знаний»), которые содержат общие сведения о принципах использования знаний.

К уровню метазнаний также относят стратегии управления выбором и применением процедурных знаний.

В зависимости от степени достоверности выделяют следующие категории знаний:

- **знания, имеющие определенную достоверность;**
- **знания с нечеткой степенью достоверности.**

В зависимости от глубины выделяют следующие виды знаний: **знания-копии; знания-знакомства; умения; навыки.**

В основе деления знаний в зависимости от степени их достоверности лежат так называемые «не-факторы», присущие знаниям:

- неполнота информации о рассматриваемом фрагменте ПрО;
- неточность количественных оценок;
- размытость качественных оценок;
- неоднозначность ряда правил вывода новых знаний;
- несогласованность некоторых положений в БЗ;
- противоречивость.

Один из способов учета подобных не-факторов при формализации знаний состоит в использовании аппарата **теории нечетких множеств** (рис. 7.1).

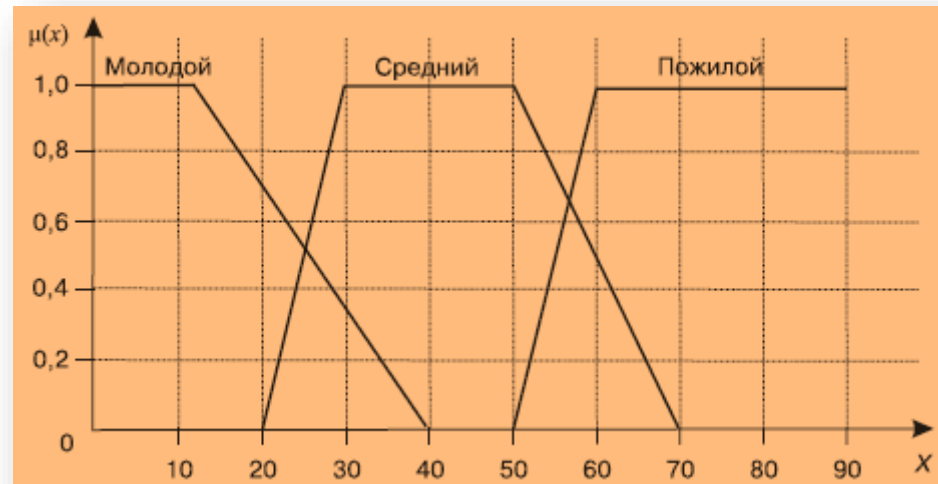


Рис. 5.1. Функции принадлежности нечетких множеств

К классу процедурных знаний с нечеткой степенью достоверности относятся **эвристики, описывающие приемы решения задач, базирующиеся на опыте экспертов в данной Про.**

С точки зрения меры возможной формализации различают **три группы эвристических методов:**

- **полностью формализованные — алгоритмы;**
- **неформализованные на данном уровне развития науки — эврисмы;**
- **частично формализованные, частично неформализованные — эвроритмы.**

Концептуальные свойства знаний:

- 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.**

МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ (МПЗ) ДЛЯ ИС

Среди МПЗ можно выделить следующие основные модели:

- **логические;**
- **продукционные (основанные на правилах);**
- **фреймовые;**
- **сетевые;**
- **объектно-ориентированные;**
- **специальные;**
- **комплексные.**

ЛОГИЧЕСКИЕ МОДЕЛИ ЗНАНИЙ

В **логических моделях** знания представляются в виде совокупности правильно построенных формул какой-либо **формальной системы (ФС)**.

Простейшей логической моделью является **исчисление высказываний**. Развитие логики высказываний нашло отражение в **исчислении предикатов первого порядка**. Представление знаний в рамках логики предикатов служит основой **логического программирования**.

Формальная система (ФС) задается четверкой (T, P, A, R) , 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 истинными; R — конечное множество правил вывода, позволяющих получать из одних синтаксически правильных выражений другие.

Положительные черты логических моделей знаний:

- высокий уровень формализации;
- согласованность знаний как единого целого;
- единые средства описания как знаний о ПрО, так и способов решения задач в этой ПрО.

Недостатки логических моделей:

- представление знаний в таких моделях ненаглядно;
- ограничения исчисления предикатов первого порядка не допускают квантификации предикатов и использования их в качестве переменных;
- 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**.

Модели знаний с открытыми БЗ и немонотонными механизмами выводов основываются на понятии **расширенной ФС (семиотической системы)**, задаваемой кортежем $(T, P, A, R, mT, mP, mA, mR)$, где T, P, A, R – составляющие замкнутой ФС; mT – правила изменения базовых элементов ФС; mP – правила изменения синтаксиса ФС; mA – правила изменения аксиом ФС; mR – правила изменения правил вывода ФС.

Семиотическая система может содержать **противоречивые и несогласованные** сведения, так как они соотносятся с разными ПрО.

ПРОДУКЦИОННАЯ МОДЕЛЬ

Центральным звеном **продукционной модели** является **множество продукций** или **правил вывода**.

Каждая такая продукция в общем виде может быть представлена выражением: $(W_i, U_i, P_i, A_i \rightarrow B_i, C_i)$, где W_i — сфера применения i -й продукции; U_i — предусловие i -й продукции; P_i — условие i -й продукции, определяемое факторами, непосредственно не входящими в A_i ; $A_i \rightarrow B_i$ — ядро i -й продукции, соответствующее правилу «если..., то...»; C_i — постусловие i -й продукции.

Системы, основанные на продукционной модели, состоят из **трех типовых компонентов**:

- **базы правил (продукций);**
- **базы фактов, содержащей декларативные знания о ПрО;**
- **интерпретатора продукций.**

Существуют **два типа механизмов вывода** в продукционных системах: **прямой** и **обратный вывод**.

Положительные стороны:

- ясность и наглядность интерпретации отдельных правил;
- простота механизмов вывода 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.

ФРЕЙМОВАЯ МОДЕЛЬ

Фундаментом **фреймовой модели знаний** служит понятие **фрейма** (теория фреймов М. Минского) — структуры данных, представляющей некоторый концептуальный объект или типовую ситуацию (рис. 7.2).

Фрейм – форма описания знаний, очерчивающая рамки рассматриваемого фрагмента ПрО.

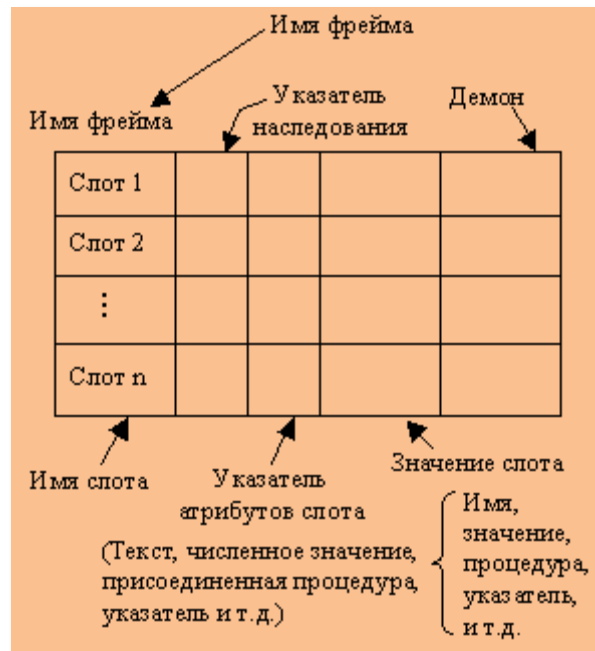


Рис. 5.2. Структура фрейма

Организация вывода во фреймовой системе базируется на **обмене сообщениями между фреймами, активации и выполнении присоединенных процедур** (рис. 7.3).

Реализация фреймовой модели знаний базируется на **языках линии LISP, FRL, KRL**.

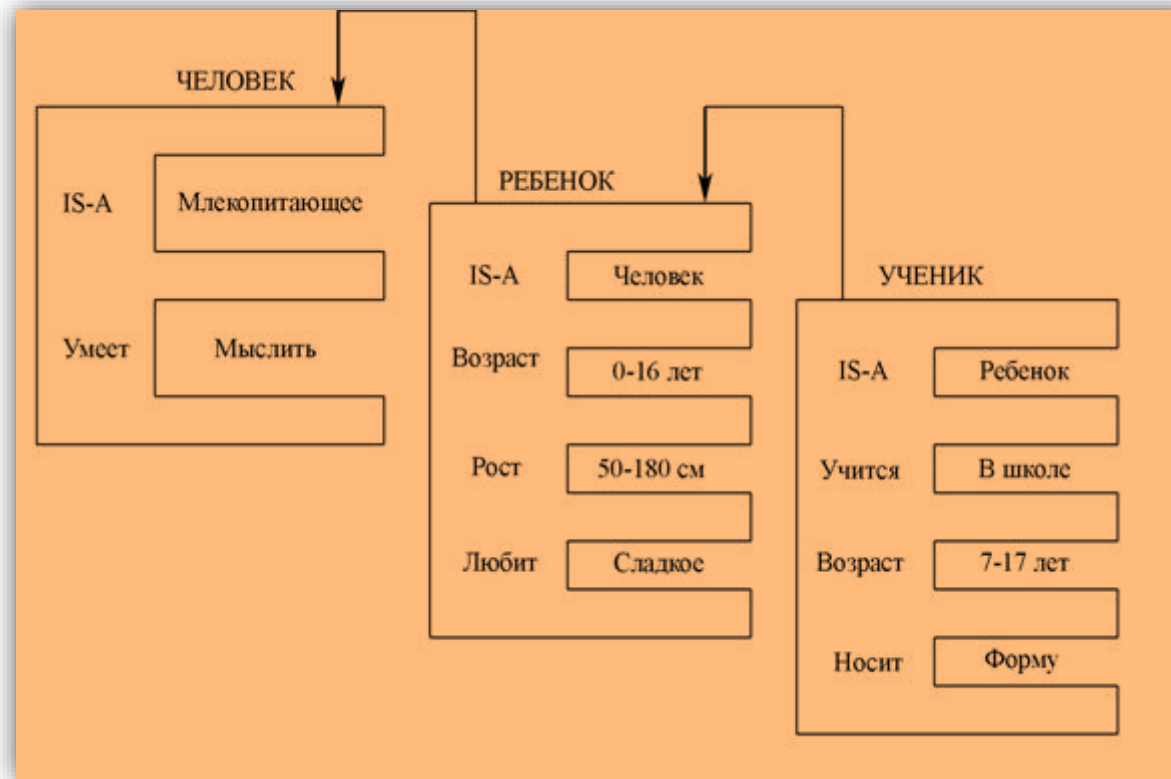


Рис. 5.3. Пример фрейма с описанием «УЧЕНИКА»

Положительными чертами в целом являются:

- наглядность;
- гибкость;
- однородность;
- высокая степень структуризации знаний;
- соответствие принципам представления знаний человеком в долговременной памяти;
- интеграция декларативных и процедурных знаний.

Недостатки фреймовой модели:

- сложность управления выводом;
- низкая эффективность его процедур.

СЕТЕВЫЕ МОДЕЛИ

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 (СС)**, вершины которой соответствуют объектам (понятиям), а дуги — отношениям между понятиями (рис. 7.4).

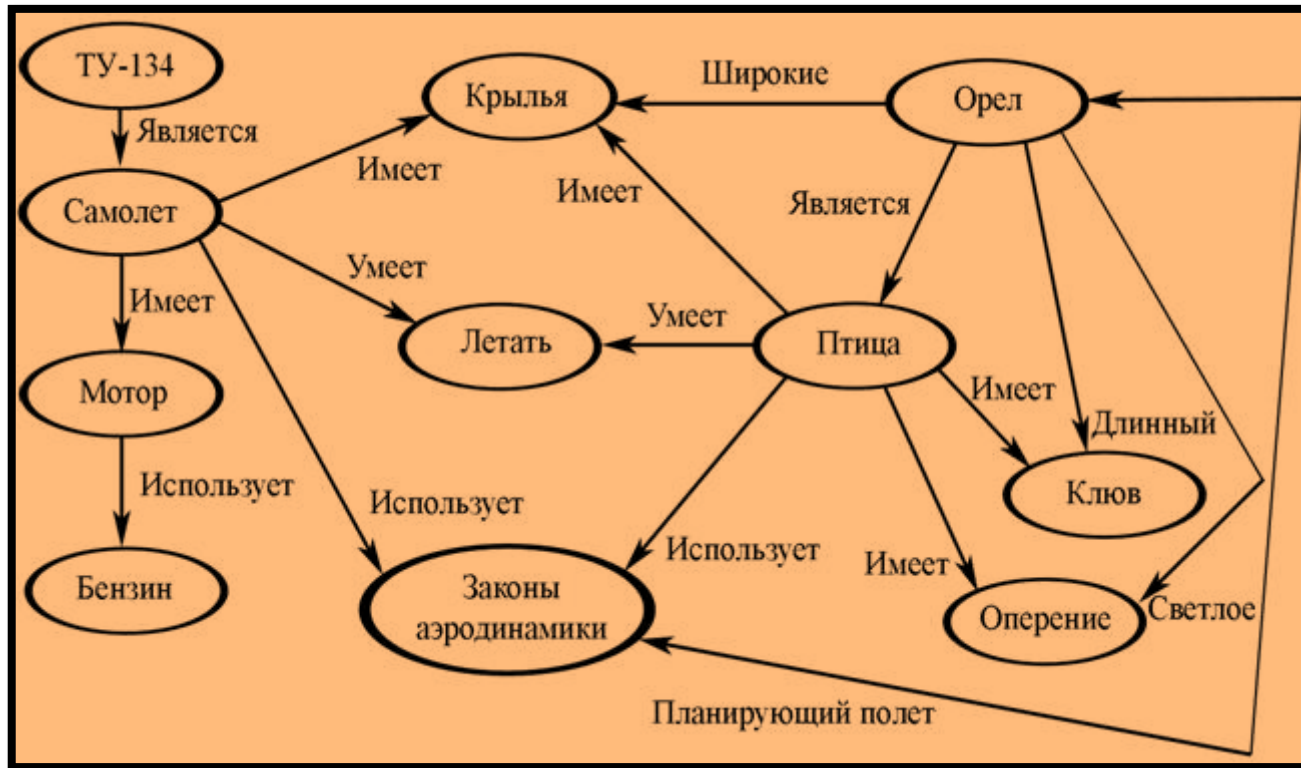


Рис. 5.4. Фрагмент СС

В общем случае под СС понимается структура: $S=(O, R) = (\{o_i \mid i= 1,2,\dots,k\}, \{r_i \mid j= 1,2,\dots,l\})$, где O – множество объектов ПрО ($|O|=k$); R – множество отношений между объектами ПрО ($|R|=l$); o_i – i -й объект ПрО; r_i – j -е отношение между объектами ПрО.

Типизация семантических сетей обуславливается 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**.

Очевидные достоинства сетевой модели:

- высокая общность;
- наглядность отображения системы знаний о ПрО;
- легкость понимания подобного представления.

Недостатки сетевой модели:

- в семантической сети имеет место смешение групп знаний, что усложняет интерпретацию знаний;
- трудность унификации процедур вывода и механизмов управления выводами на сети.

Наиболее известные отечественные **модели СС**:

М1 — расширенные СС И.П. Кузнецова;

М2 — неоднородные СС Г.С. Осипова;

М3 — нечеткие СС И.А. Перминова;

М4 — обобщенная модель представления знаний о ПрО А.И. Башмакова.

МОДЕЛЬ М1 — РАСШИРЕННЫЕ СЕМАНТИЧЕСКИЕ СЕТИ (РСС)

Для устранения неоднородности обычных СС в модели РСС введены **вершины**, соответствующие именам отношений, и **вершины связи**, выполняющие роль «развязывающих» элементов (**разрывают 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: $D_0 (D_1, D_2, \dots, D_k / D_{k+1})$, где D_0 — имя отношения; D_1, D_2, \dots, D_k — объекты, участвующие в отношении; D_{k+1} — вершина связи, обозначающая всю совокупность объектов, участвующих в отношении; эта вершина также называется c -вершиной ЭФ; $D_0, D_1, D_2, \dots, D_{k+1} \in D$, $k > 0$.

РСС рассматривается как конечное множество ЭФ: $RCC = \{ЭФ_i\}$, $i = 1, \dots, n$.

Формально описание РСС имеет вид:

- 1) если $\{D_0, D_1, \dots, D_k, D_{k+1}\} \subseteq D$, $k > 0$, то $D_0 (D_1, \dots, D_k / D_{k+1}) = T_0$;
- 2) каждый T_k есть РСС;
- 3) если T_1 и T_2 - РСС, то композиции $T_1 \bullet T_2$ и $T_2 \bullet T_1$ - тоже РСС, при этом $T_1 \bullet T_2 = T_2 \bullet T_1$.

Имена отношений играют роль объектов и могут вступать в отношения, что определяет высокую однородность модели. Вершина связи ЭФ может входить в другие ЭФ, но не в роли c -вершины.

В D выделяется три непересекающихся подмножества: $D = G \cup X \cup E$, 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: $G = R \cup A \cup \{t, f\}$.

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 операции над ними. Для этого проводится расширение множества R .

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: $Tn1 \rightarrow Tn2$** .

Для представления продукций в множество E включены две специальные вершины: $E=\{P, S\}$, где P — специальная вершина, соответствующая **отношению причина-следствие**, а S — специальная вершина, соответствующая **отношению часть-целое**.

В свою очередь, продукции могут входить как в левые, так и в правые части других продукций, что позволяет создавать метауровни РСС. На базе модели РСС построен **язык продукционного программирования Декл**.

МОДЕЛЬ M2 — НЕОДНОРОДНЫЕ СЕМАНТИЧЕСКИЕ СЕТИ

Неоднородная СС (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, τ, Δ, F, R) , где D — семейство произвольных множеств D_1, \dots, D_n ; $\tau = \{k_1, k_2, \dots, k_e\}$ — 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 k_i представляет собой упорядоченное подмножество индексов из множества $\{1, 2, \dots, n\}$ ($n = |D|$). Для каждого типа $k_i = (n_1, \dots, n_{k_i})$ строится декартово произведение множеств из D :

$$D_{k_i} = D_{n_1} \times \dots \times D_{n_{k_i}} \quad (n_{k_i} \leq n).$$

Тип k_i интерпретируется как тип декартова произведения. Для каждого D_{k_i} определяется совокупность его подмножеств Δ_{k_i} . Всякое $d \in \Delta_{k_i}$ называется **событием типа k_i** . Объединение всех Δ_{k_i} образует множество Δ : $\Delta = \bigcup_{k_i \in \tau} \Delta_{k_i}$.

$$k_i \in \tau$$

Функции из F , соответствующие типу k_i , отображают D_{k_i} в одно из множеств семейства D .

На множестве событий задано семейство **отношений совместности событий**: $R = \{R_1, R_2, R_3, R_4, R_5, R_6, R_1^0, R_2^0, R_3^0, R_4^0, R_5^0, R_6^0\}$, причем R_i^0 обратны для R_i : $R_i^0 = (R_i)^{-1}$, $i = 1, \dots, 6$.

При создании модели НСС набор отношений совместности был сформирован, исходя из потребностей обработки данных в медицинских ЭС и 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.

МОДЕЛЬ МЗ — НЕЧЕТКИЕ СЕМАНТИЧЕСКИЕ СЕТИ

Предусмотренный в М1 механизм обработки РСС имеет два существенных недостатка:

- слабая структурированность формируемых программ;
- сложность организации процесса вычисления.

Для их преодоления на базе М1 разработана модель ОО СС (ОСС). В её основе лежит объединение РСС с нечетким ПРОЛОГОм с использованием ООП.

В сеть добавлены явные средства обработки нечеткой информации, **вершины-переменные, вершины-классы, вершины-объекты** и **вершины-экземпляры**.

Аргументы предикатов интерпретируются как вершины СС, а факты представляются ее фрагментами. Правила привязаны к классам.

Синтаксис ПРОЛОГа расширен для обработки сети. Предикат \leftrightarrow ЭФ, при этом правила ПРОЛОГа являются частным случаем **сетевых продуктов (СП)** в РСС.

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 $h_j \in [0; 0,5[$ ЭФ_j. При $h_j = 0,5$ ЭФ_j отсутствует в ОСС.

Такой нечеткий ЭФ служит элементарной структурной единицей **нечеткой ОСС (НОСС)**.

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).
4. Вершины-связи (ассоциируются с вершинами D_{k+1} ЭФ), используемые для ссылок на ЭФ.
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.

Выполнение программы базируется на методе нечеткой резолюции. Результат вывода характеризуется: степенью истинности заключения; степенью доверия к резольвенте; смешанной истинностью заключения.

Из-за недостатков известной стратегии нечеткого вывода в НОСС использована линейная стратегия вывода с бэктрекингом.

Для описания данных и программного кода, реализующих сеть, разработан язык *FSNL (Fuzzy Semantic Network Language)* — ООЯ без строгой типизации, предназначенный для оперирования НОСС.

МОДЕЛЬ М4 - ОБОБЩЕННАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ О ПРО

При разработке М4 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**, т.е. выбор фундаментальной категории для формирования других категорий.

Триада: **«вещь-свойство-отношение»**.

Среди основных проявлений взаимосвязи составляющих триады необходимо отметить:

- **взаимобоснование** свойств и отношений в вещах;
- **взаимопереход** вещей, свойств и отношений.

Принципы взаимобоснования и взаимоперехода обуславливают **высокую степень однородности** модели представления ПрО, основанной на указанной триаде.

М4 включает три базовых уровня:

- информационных структур;
- операций;
- стратегии управления операциями.

На уровне информационных структур модели выделяется **множество объектов $\{O_i\}$** .

Объект задается тройкой: $O_i = (A_i, P_i, R_i)$, где A_i - вещь, соответствующая объекту O_i ; P_i - свойство, соответствующее объекту O_i ; R_i — отношение, соответствующее объекту O_i .

Модель **М4** способна отражать **многоаспектную нечеткость** представления ПрО.

Типизация нечетких свойств и нечетких отношений зависит от проблемной ориентации БЗ, в которой реализуется **М4**, и может быть различной.

В самом общем виде нечеткое свойство или отношение X можно задать парой: $(I(X), V(X))$.

Здесь **$I(X)$** — ссылка на объект, соответствующий свойству или отношению X ; **$V(X)$** — обоснование свойства или отношения X .

Обобщенный характер модели **М4** состоит в том, что она **определяет методологию построения моделей представления знаний о ПрО**, реализуемых в прикладных ИС.

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:

- четырехуровневая модель **MDA (Model Driven Architecture)** консорциума **Object Management Group (OMG)**;
- трехуровневая модель **ODP (Model of Open Distributed Processing)**, зафиксированная в стандарте **ISO/ITU**.

Стандарт **ODP** определяет пять возможных видов представлений:

- корпоративное (*enterprise*);
- информационное (*information*);
- вычислительное (*computational*);
- разработчиков (*engineering*);
- технологическое (*technology*).

Сравнение данных подходов показывает, что они позволяют моделировать одну и ту же ПрО с разных точек зрения. В **MDA** модели описывают ПрО непосредственно в отличие от **ODP**.

Класс специальных моделей знаний объединяет модели, отражающие особенности представления знаний и решения задач в отдельных, относительно узких ПрО.

Смешанные или комплексные модели, интегрирующие преимущества рассмотренных выше базовых моделей представления ПрО.

Ключевые требования 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 моделей знаний не удовлетворяет всем девяти требованиям.

К представлению знаний как направлению ИИ традиционно относят задачи **верификации знаний, пополнения БЗ** за счет логического вывода, **обобщения** и **классификации знаний** (систематизация знаний).

Вопросы для самопроверки

1. Охарактеризуйте понятие «знания».
2. Какие виды знаний принято выделять?
3. Что относят к концептуальным свойствам знаний? Поясните каждое свойство.
4. Какие существуют классы моделей знаний?

5. Дайте характеристику логических моделей знаний.
6. Что служит центральным звеном продукционной модели знаний?
7. Охарактеризуйте фреймовую модель знаний.
8. Охарактеризуйте сетевые модели знаний.
9. Сформулируйте основные требования к моделям знаний.

ЛЕКЦИЯ 6. ОНТОЛОГИЧЕСКИЙ ПОДХОД И МЕТАДААННЫЕ

ПОНЯТИЕ ОНТОЛОГИИ

Онтологии имеют непосредственное отношение к построению БЗ и частично к реализации интеллектуального интерфейса.

Они 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.

Онтология — это формально представленные на базе концептуализации знания о ПрО.

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 подобного процесса, т.е. описание множества понятий (концептов) ПрО, знаний о них и связях (отношениях) между ними.

Самым распространенным на данный момент является определение, согласно которому **онтология есть точная (выраженная формальными средствами) спецификация концептуализации.**

Онтология является не абсолютной (единственной) спецификацией концептуализации ПрО, а зависит от целей ее создания. **Независимо от вида онтологии** она должна включать **словарь терминов** и некоторые **спецификации их значений**. При таком подходе **онтология** похожа на **тезаурус**.

Интерпретации понятия «онтология»:

- 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.**

В неформальной трактовке онтология представляет собой описание некоторой 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.

ОСНОВНЫЕ ЗАДАЧИ, РЕШАЕМЫЕ С ПОМОЩЬЮ ОНТОЛОГИЙ

1. Создание и использование БЗ.
2. Организация эффективного поиска в БД, информационных каталогах, БЗ.
3. Создание систем, реализующих механизмы рассуждений (ЭС, системы управления, интеллектуальные роботы).
4. Организация поиска по смыслу в текстовой информации.
5. Семантический поиск в *Internet*.
5. Представление смысла в метаданных об ИР.
6. Построение и использование баз общих знаний для различных ИС.
7. Обеспечение общей терминологии для множества специалистов и совместно используемых приложений.
8. Многократное применение БЗ и информационных массивов, представляющих сведения о технических системах на различных стадиях их жизненного цикла.

МОДЕЛЬ ОНТОЛОГИИ

Модель онтологии O задает тройка $O = (X, R, \Phi)$, где X — конечное непустое множество концептов PrO , которую представляет онтология; R — конечное множество отношений между концептами; Φ — конечное множество функций интерпретации, заданных на концептах и (или) отношениях.

Эта модель является разновидностью сетевой модели знаний. В начале развития онтологического подхода для представления онтологии использовались языки ИППП и другие языки математической логики. Сейчас применяются эквивалентные по семантике языки нелогического типа, специально предназначенные для описания онтологии.

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**: $O_{расш} = (O_m, \{(O_p, O_z)\}, MB)$, где O_m — онтология верхнего уровня (метаонтология); O_p — предметная онтология; O_z — онтология задач ПрО; MB — модель машины вывода.

Метаонтология O_m оперирует, не зависящими от ПрО концептами и отношениями (время, объект, свойство и т.д.). Она изменяется весьма незначительно и может считаться статической. Поэтому вывод на ее уровне достаточно эффективен.

Предметная онтология O_p содержит понятия, описывающие конкретную ПрО, семантически значимые для нее отношения, а также декларативные и процедурные интерпретации этих понятий и отношений.

В **онтологии задач O_z** в качестве понятий выступают типы решаемых задач, а отношения, как правило, специфицируют декомпозицию задач на подзадачи.

Машина вывода начинает работу при активации понятий или отношений, описывающих исходную ситуацию (задачу). Вывод на СС организуется как волновой процесс, использующий свойства отношений, выходящих из узлов, задающих исходную ситуацию.

Критерием останова процесса является достижение целевой ситуации, либо превышение длительности времени, отведенного для решения задачи.

МЕТОДИКИ ПОСТРОЕНИЯ ОНТОЛОГИИ И ТРЕБОВАНИЯ К СРЕДСТВАМ ИХ СПЕЦИФИКАЦИИ

Существует много предложений по методикам разработки онтологии.

Стандарт онтологического исследования *IDEF5* подготовленный фирмой *Knowledge Base Systems, Inc.* в качестве проекта национального стандарта США (1994).

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.**

Для поддержки процесса построения онтологии в *IDEF5* определены специальные онтологические языки:

- **схематический язык (*Schematic Language — SL*);**
- **язык доработок и уточнений (*Elaboration Language — EL*).**

Язык *SL* является наглядным графическим языком диаграммного типа, предназначенным для формирования начального представления онтологии, а также дополнения существующих онтологий.

Язык *EL* – структурированный текстовый язык, позволяющий детализировать элементы онтологии (структурировать элементы концептуализации).

В стандарте *IDEF5* предусмотрены четыре 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*);
- **диаграмма естественной или видовой классификации** (*Natural Kind Classification — NKC*).

В *DS* определяющие свойства класса являются необходимым и достаточным признаком принадлежности объекта этому классу. С помощью диаграмм *DS*, как правило, классифицируются логические объекты.

В *NKC*, наоборот, свойства класса не 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 визуализировать и изучать связи между различными классами объектов системы, а также представлять зависимости между взаимосвязями классов.

Диаграмма состояния объекта (*Object State Schematic*) позволяет описать процесс изменения состояния объекта (смена состояния или смена класса).

Стандарт *IDEF5* отражает методологию, с помощью которой можно наглядно и эффективно разрабатывать онтологии, но данный стандарт охватывает не все этапы создания онтологии.

Существует еще одна методология построения онтологии. Для ее поддержки предназначена специальная **инструментальная среда проектирования онтологии (*Ontology Design Environment – ODE*)**.

Она включает **подсистемы управления проектом и поддержки разработки**. Первая подсистема 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.**

Наиболее сложной задачей является концептуализация. От успешности ее выполнения зависит эффективность всей разработки.

Концептуализация включает два этапа:

- 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.

В качестве примера программного решения для создания ИС на основе онтологического подхода можно привести мощную среду разработки **OntoStudio** фирмы **Ontoprise GmbH** (<http://www.ontoprise.de>), которая поддерживает следующие форматы и языки онтологического проектирования: *OWL, RDF (S), RIF, ObjectLogic*.

НАИБОЛЕЕ ИЗВЕСТНЫЕ ОНТОЛОГИЧЕСКИЕ ПРОЕКТЫ

В настоящее время разрабатываются онтологии всех трех уровней: верхнего уровня, предметные и онтологии задач. Наиболее активно создаются предметные онтологии.

1) Среди проектов **онтологии верхнего уровня** наиболее существенные результаты достигнуты в рамках **проекта Сус** корпорации **Cycorp** (Техас, США). Данный проект направлен на формирование информационно-логической основы для создания и функционирования систем, реализующих механизм рассуждений.

Онтология **Сус** базируется на ядре, включающем миллион утверждений, введенных в БЗ вручную. Версия **Сус** с открытым исходным кодом (**проект OpenCyc**) на сегодняшний день содержит описания **6000 концептов** и **свыше 60000 аксиом**, отражающих непротиворечивые знания о мире.

2) Другим примером онтологии верхнего уровня является **модель GUM (Generalized Upper Model)** (<http://www.darmstadt.gmd.de/publish/komet/genum/newUM.html>).

Цель проекта GUM — поддержка обработки ЕЯ (английского, немецкого и итальянского). Уровень абстракции при представлении онтологии *GUM* лежит между лексическими и концептуальными знаниями. Это позволяет упростить создание на ее базе ЕЯ интерфейса. Модель *GUM* содержит отдельные таксономии понятий и связей.

3) **Онтологии предметного уровня** разрабатываются в рамках проекта *TOVE (Toronto Virtual Enterprise)*, целями которого являются: создание средств представления общей терминологии для ПрО; 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 онтологии операций, состояний и времени, организации, ресурсов, продуктов, сервиса, производства, цены, количества.

4) **Проект KACTUS** выполняется в рамках программы Европейского Союза *ESPRIT*. Его цель заключается в создании методологии многократного применения знаний о технических системах на протяжении их ЖЦ. Другими словами, данная методология позволяет использовать одни и те же БЗ при проектировании, оценке, эксплуатации, сопровождении, репроектировании и обучении. В *KACTUS* сделана попытка объединить эти онтологии с существующими стандартами (*STEP*) на основе языка *CML (Conceptual Modeling Language)*.

5) **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.

Выделяются три направления исследований:

- **онтологический инжиниринг;**
- **автоматическое аннотирование *web*-страниц;**
- **построение и выполнение запросов к *web*-страницам и вывод результатов с использованием онтологий.**

6) В рамках проекта *SHOE (Simple HTML Ontology Extensions)* 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. Основным компонентом *SHOE* является онтология, представляющая сведения о некоторой ПрО.

7) Система *Ontolingua*, разработанная в *Knowledge System Laboratory (KSL)* отделения информатики Стенфордского университета, представляет собой инструментальное средство для совместного формирования, просмотра, редактирования и использования онтологии в среде *WWW*. Язык *Ontolingua* реализует принципы ООП и является расширением языка *KIF (Knowledge Interchange Format)*.

На сервере *Ontolingua* представлен ряд онтологических библиотек по различным областям знаний. С помощью системы *Ontolingua KSL* выполняется проект по формированию широкомасштабного репозитория выразительных и многократно используемых знаний.

СИСТЕМЫ И МОДЕЛИ МЕТАДААННЫХ

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.

Метаданные (*metadata*) — это информация о документе, понимаемая ЭВМ, т.е. обладающая свойством внутренней интерпретируемости.

Экземпляр метаданных для информационного ресурса (ИР) выступает в качестве описания этого ИР. 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.**

Метаданные могут характеризовать **сущности**, относящиеся не только к **виртуальному (информационному) пространству**, но и к **реальному миру (персоналии, организации, события).**

Система метаданных выступает в качестве центрального звена любой ИС. Метаданные могут быть как частью ИР, так и храниться отдельно от него.

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

- **инфологический**, фиксируемый **схемой метаданных**;
- **даталогический**, фиксируемый **форматом метаданных.**

К числу основных требований к системе метаданных относятся:

- универсальность в рамках установленного понимания ИР как объекта систематизации;
- структурированность и формализованность метаданных, необходимые для их автоматической обработки;
- достаточная выразительность для обеспечения 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, распространителях ИР и отношений между ИР.

Метаданные об ИР формируются и используются в различных системах и сервисах (электронных библиотеках; *web*-сайтах; хранилищах ИР и пр.).

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.**

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*, но сама модель 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*** (рис. 6.1). Базовыми категориями такого описания являются ИР (субъект), свойство (предикат) и значение (объект).

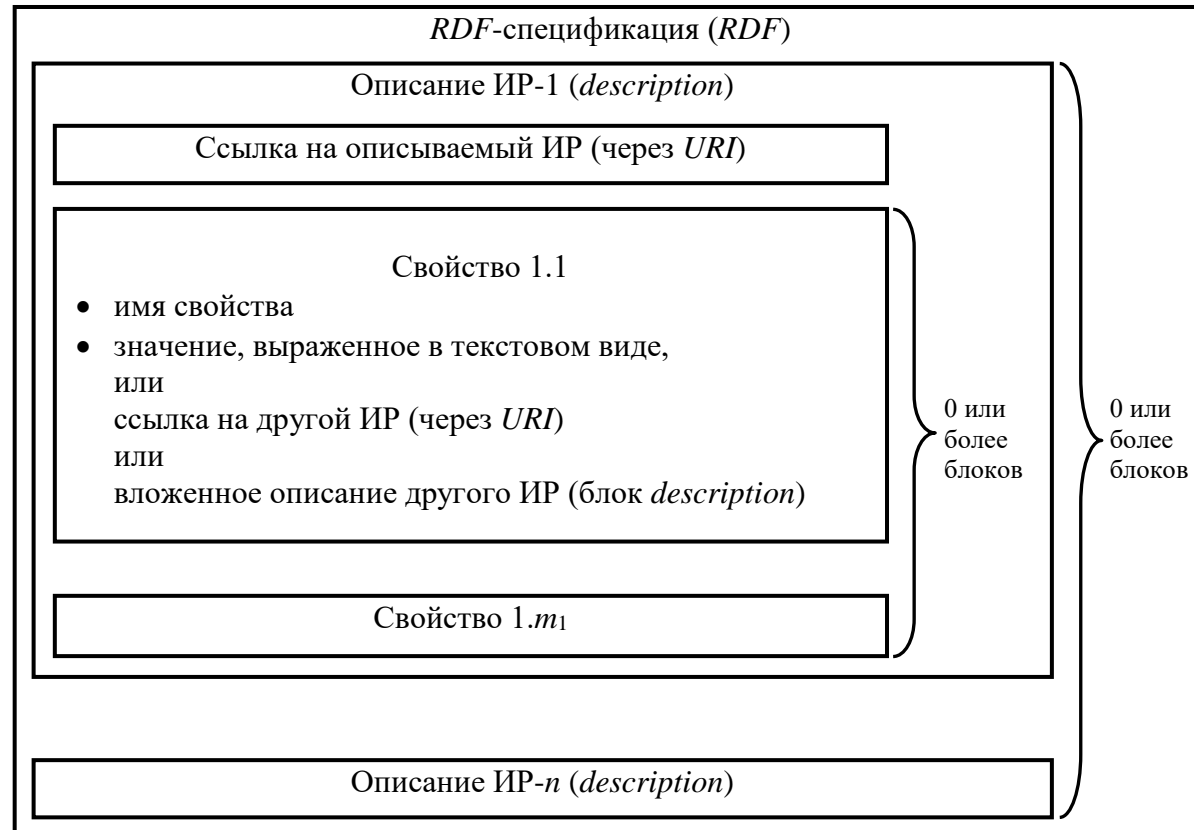


Рис. 6.1. Упрощенная структура *RDF*-спецификации

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.

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 и т.д.);
- **идентифицирующие**, служащие для однозначного представления описываемых объектов.

К настоящему времени в мире создано множество систем метаданных, обладающих разным статусом (международные, национальные и отраслевые стандарты, корпоративные спецификации, спецификации международных консорциумов и др).

Системы метаданных:

- «Дублинское ядро» (инвариантный к ПрО набор наиболее общих полей описания ИР, введенный для обеспечения глобальной интероперабельности приложений, работающих с метаданными);
- *MARC* — предназначена для описания библиотечных ресурсов (как на бумажных, так и на электронных носителях);
- *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*;
- формат описания БД и машиночитаемых информационных массивов.

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:

- простое «Дублинское ядро» (*Simple Dublin Core*);
- «Дублинское ядро» с квалификаторами (*Qualified Dublin Core*).

Первый уровень содержит 15 элементов данных, образующих три группы (табл. 6.1):

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

Таблица 6.1.

Состав простого «Дублинского ядра»

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

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)
- ***Rights Holder*** (правообладатель).

Кроме того, для повышения детальности и выразительности описаний на этом уровне вводятся и используются **квалификаторы**, уточняющие семантику элементов данных и специфицирующие источники и способы представления их значений.

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

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

- 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.
- 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.

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».

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*.

СЕМАНТИЧЕСКИЙ WEB И ПЛАТФОРМА XML

Недостатки и ограничения технологий *Internet* первого поколения (*web 1.0*) привели к разработке консорциумом *W3C* концепции «семантической паутины» (*Semantic Web* или *web 2.0*). 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;
- 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 1.0* с сохранением их достоинств.

К числу основных положительных черт *web 1.0* можно отнести:

- 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.

Около 70 % ИР *Internet* явно не представлены в *web 1.0*, 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*.

Основой *web 2.0* служит расширяемый язык разметки **XML** (**платформа XML** – перечень взаимосвязанных и согласованных стандартов и спецификаций, имеющих общее функциональное назначение и опубликованных на сайте *W3C* <http://www.w3.org>), а единицей доступа к ИР *web 2.0* является *XML*-документ.

Все стандарты и спецификации платформы *XML* синтаксически едины: компоненты платформы, расширяющие функциональность *XML*, используют синтаксис этого языка, т.е. являются приложениями *XML*. Кроме того, платформа *XML* обеспечивает совместимость *web 2.0* с технологиями *web 1.0*.

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.**

XML позволяет представлять как слабоструктурированные данные (документы без модели), так и структурированные данные (документы, ссылающиеся на модели).

Наличие модели позволяет **автоматически верифицировать XML-документ.**

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.

Концепция **Web 3.0** (проекта *WIP* Евросоюза) призвана привести в соответствие архитектуру всемирной Сети растущей потребности к созданию пользователями сетей произвольной архитектуры.

Концепция *Web 3.0* предполагает создание надежного, гибкого, оптимизируемого и при этом «дружественного» по отношению к пользователям набора технологий и стандартов, которые позволили бы любому пользователю, где бы он ни находился, идентифицировать любое находящееся поблизости от него устройство и создать сеть с ним.

Web 3.0 — новая мощная технология создания веб-приложений, разработанная первоначально для максимального удобства и комфорта пользователей, и как следствие, значительно упрощающая разработку самих приложений в отличие от *Web 1.0* и *Web 2.0*.

Вопросы для самопроверки

1. Что понимается под концептуализацией?
2. Охарактеризуйте различные интерпретации понятия «онтология».
3. Какие основные классы задач решаются с использованием онтологии?
4. Как представляется модель онтологии?
5. Что такое модель расширенной онтологии? Охарактеризуйте ее компоненты.
6. Какие этапы построения онтологии предусмотрены стандартом *IDEF5*?
7. Для чего предназначены языки *SL* и *EL*?
8. Какие типы диаграмм предусмотрены в *IDEF5*?
9. Какие этапы построения онтологии поддерживаются инструментальной средой *ODE*?
10. Каково назначение онтологии верхнего уровня? Приведете примеры таких онтологий.
11. Каково назначение онтологии предметного уровня? Приведете примеры таких онтологий.
12. Охарактеризуйте понятие «метаданные».
13. Что понимается под системой метаданных?

14. Где и для чего используются метаданные?
15. Каковы основные требования к системе метаданных?
16. Дайте характеристику модели *RDF*.
17. Назовите основные виды метаданных.
18. Перечислите наиболее известные системы метаданных.
19. Перечислите элементы системы метаданных «Дублинское ядро».
20. Где могут храниться метаданные?
21. Какие достоинства, недостатки и ограничения присущи технологиям *Internet* первого поколения?
22. Что такое скрытый *web*? Что ограничивает доступ к его IP?
23. На что направлена концепция *Semantic Web*? Каковы ее основные компоненты?
24. Какие языки предназначены для описания моделей *XML*-документов?

ЛЕКЦИЯ 7. ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ (ИНС)

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 России сформировались три крупные **научные школы в области нейротехнологий**.

1. **Центр нейрокомпьютеров РАН (А.И. Галушкин).**
2. **Научная школа нейротехнологий МГУ (А.В. Чечкин).**
3. **Научная школа нейротехнологий в Красноярском государственном университете (А.Н. Горбань).**

Сферами применения нейротехнологий являются все плохо формализуемые ПрО, где классические математические модели и алгоритмы мало эффективны по сравнению с человеком (обработка изображений, реализация ассоциативной памяти, системы управления реального времени, распознавание образов и речи и др.).

Одной из характерных черт нейротехнологий является **обучение нейросети на примерах**.

Шесть феноменов мозга:

1. Кодирование (представление) информации о внешнем мире;
2. Кратковременное и долговременное запоминание, хранение и извлечение информации;
3. Ассоциативный поиск и самоорганизацию памяти;
4. Оперирование информацией в процессе решения мыслительных задач;
5. Симультанное (мгновенное) распознавание;
6. Неожиданное творческое озарение (инсайт).

Конструктивного научного объяснения этим феноменам до сих пор не найдено.

Большинство нейрофизиологов считает, что объяснить феномены работы мозга можно, изучая функционирование объединенных в единую сеть клеток, называемых **нейронами**.

Мозг состоит из различных типов клеток и включает 10^{10} — 10^{11} нейронов. **Количество связей** между ними может достигать 10^{22} . 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 высшей нервной системе (ВНС) человека явно различают три типа нейронных структур:

- **сенсорные;**
- **внутренние;**
- **эффекторные.**

Парадигма нейрокомпьютинга формулируется следующим образом (Горбань): алгоритмы, порождаемые данными в универсальном процессе обучения, специализированные для данного класса операций с образами, адаптированные под конкретные информационные задачи.

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. У правого полушария данные и метод составляют единое целое. Оно также отвечает за воображение и интуицию. Это более древние механизмы мозга по сравнению с логическим мышлением. Правое полушарие поддерживает параллельную обработку информации, оно же отвечает за творчество.

Структура работ в области нейрокибернетики:

- Разработка, реализация и использование математических моделей ИНС.
 - Разработка и использование нейропакетов.
 - Разработка и программная реализация математических моделей элементов ИНС.
 - Программная реализация нейропакетов.
 - Обучение нейропакетов решению различных классов задач.

- Разработка и использование нейрокомпьютеров.
 - Проектирование и конструирование нейрокомпьютеров.
 - Программная реализация нейрокомпьютеров.
 - Обучение нейрокомпьютеров решению различных классов задач.
- Разработка и реализация математических моделей ВНС человека.
 - Разработка и программная реализация математических моделей элементарных сенсорных систем человека.
 - Разработка и программная реализация математических моделей элементарной языковой системы человека.

Сравнение основных характеристик традиционных компьютеров и нейрокомпьютеров приведено в табл. 7.1.

Таблица 7.1.

Характеристики традиционных компьютеров и нейрокомпьютеров

Основные Характеристики	Традиционные компьютеры	Нейрокомпьютеры
Режим функционирования	В основном последовательный	Параллельный
Описание функционирования	Заданные алгоритмы	Алгоритмы формируются на основе обучения нейросети на примерах
Характер операций	Иерархическая структура алгоритмов. Разбиение сложных задач на простые. «Жесткие» математические модели	Непосредственное манипулирование образами. «Мягкие» математические модели
Аналог	Левое полушарие	Правое полушарие

Вопросы для самопроверки

1. На какой парадигме основан нейрокомпьютинг?
2. Что обычно относят к феноменам мозга?
3. Опишите структуру работ в области нейрокибернетики.

ЛЕКЦИЯ 8. ИНС. БАЗОВЫЕ ПОНЯТИЯ НЕЙРОТЕХНОЛОГИЙ

1. **Нейрон** – базовый элемент сети (рис. 9.1 и рис. 9.2), единственный ее нелинейный элемент;
2. **Синапс** – элемент, обеспечивающий связь между нейронами (может обладать весом и «задержкой»);
3. **Сумматор**, рассматриваемый как компонент нейрона или как специализированный нейрон;
4. **Обучающие примеры**, представляющие наборы значений вход – предписанный выход и целевую функцию, определяющую штраф за отклонение реального выхода от предписанных значений при данных входах;
5. **Цвета и цветовые группы**, ассоциируемые с нейронами (окрашенные нейроны по разному участвуют в обучении);
6. **Алгоритмы обучения.**

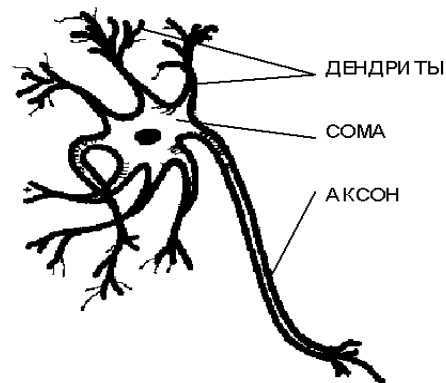


Рис. 8.1. Схема биологического нейрона

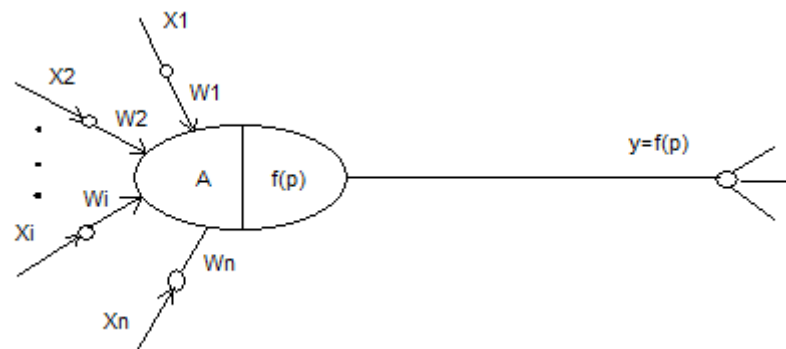


Рис. 8.2. Модель искусственного нейрона

ИНС представляет собой совокупность искусственных нейронов, организованных слоями. **ИНС** могут быть **одно-** и **многослойные, с обратными связями и без.**

В 1958 году Фрэнк Розенблатт ввел понятие **перцептрона** – модели ИНС и рассмотрел возможность модификации межнейронных связей, что сделало сеть обучаемой (рис. 9.3).

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 нейрокомпьютеров (НК) (универсальных и специализированных); учителя ИНС.
3. Выделяют два подхода к организации обучения ИНС: обучение и самообучение на примерах; обучение в процессе игры.

Под обучением ИНС понимается процесс нахождения экстремума некоторой функции, отображающей взаимодействие типа вход-выход.

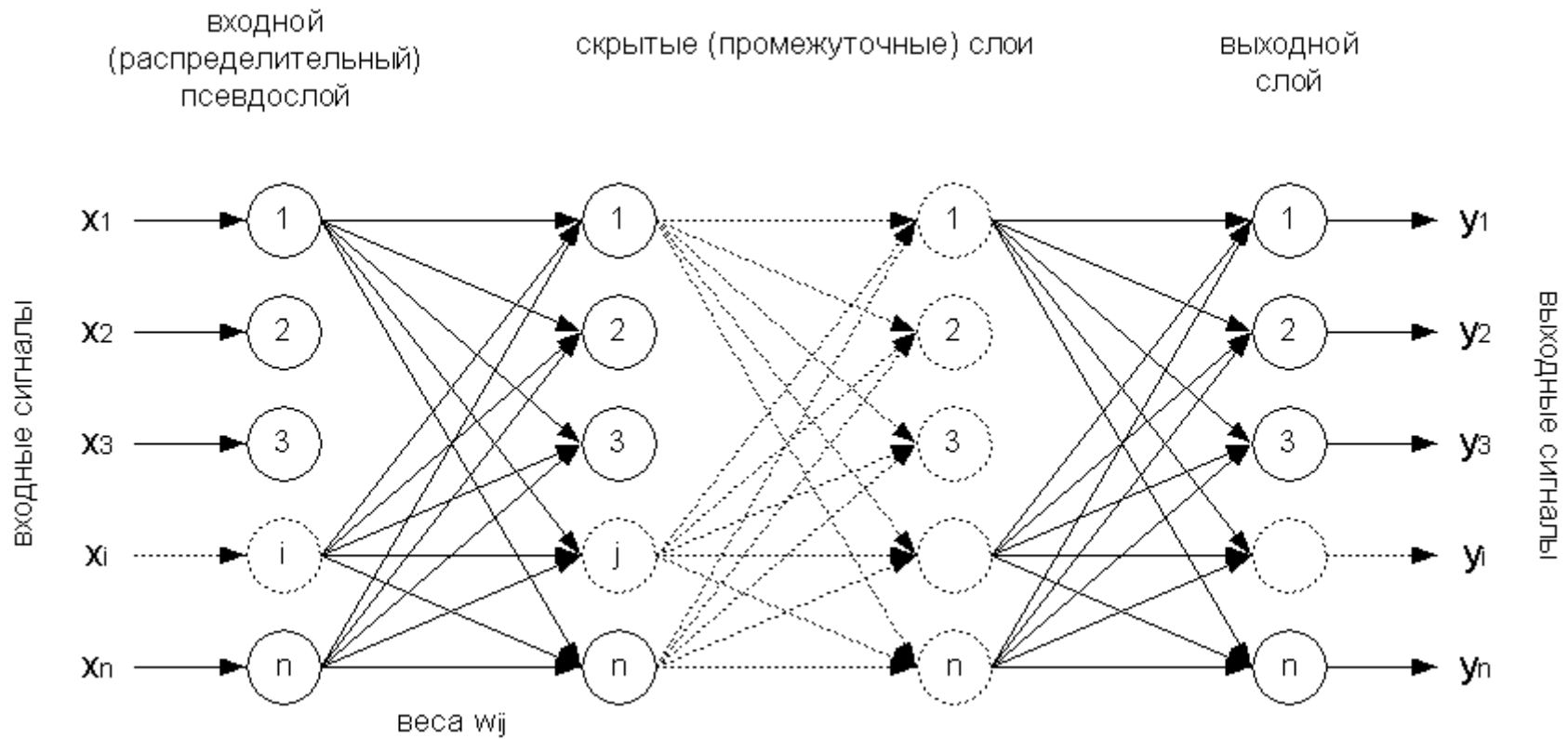


Рис. 8.3. Схема многослойного персептрона

Достижения в области обучения ИНС:

- Возможность управления процессом обучения в режиме двойственного функционирования ИНС, при котором нейропакет (НП) или НК самостоятельно оценивает ошибки и сообщает об этом учителю;
- Возможность восприятия и отражения на ИНС информации из внешней среды.

Спецификой является большая размерность ИНС, что обуславливает целесообразность применения алгоритмов оптимизации, которые допускают параллельное выполнение аналогично естественным нейросетям. Кроме методов оптимизации при обучении ИНС используют генетические алгоритмы.

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.

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.

Три базовых подхода к представлению результатов обучения нейросетью:

- 1. Коннекционизм** — модифицируются веса синаптических связей, параметры нейронов не меняются;
- 2. Гетерогенные ИНС** — модифицируются параметры нейронов, связи не меняются;
- 3. Комплексный подход**, объединяющий первые два подхода.

НЕЙРОПАКЕТЫ

Нейропакетом называется программная система, эмулирующая среду НК на обычном компьютере.

Классификация НП

1. НП для разработки других НП (инструментарий построения НП).
2. Универсальные НП (рис. 9.4). 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:

- **NeuroSolutions** (разработчик — *NeuroDimension, Inc.*);
- **Neuro Windows** (разработчик — *Ward Systems Group*);
- **NNet+** (разработчик — *NeuroMetric Vision System*);
- **Neural Network Toolbox for Matlab** (разработчик — *Math Works*);
- **Neuro Office** (разработчик — ЗАО «АльфаСистем»).

Обычно процесс разработки с помощью НП 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).**

Наиболее распространенный способ обучения ИНС основан на методе обратного распространения ошибки.

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.

Критерии для сравнения универсальных НП:

1. Скорость обучения ИНС – главный показатель эффективности функционирования НП;

Критерии оценки НП с точки зрения начинающих пользователей

2. Простота формирования и обучения ИНС при использовании интуитивно понятного графического интерфейса НП;
3. Простота подготовки обучающей выборки;
4. Наглядность и полнота представления информации в процессе формирования и обучения ИНС;

Критерии оценки НП с точки зрения опытных пользователей

5. Состав поддерживаемых нейронных моделей, критериев и алгоритмов обучения;
6. Возможность создания собственных (т.е. нетиповых) нейронных структур;
7. Возможность использования собственных критериев оптимизации;
8. Возможность использования собственных алгоритмов обучения ИНС;
9. Простота обмена информацией между НП и другими приложениями;

Критерии оценки НП с точки зрения профессиональных разработчиков НП

10. Открытая архитектура пакета (возможность его расширения за счет внешних программных модулей);
11. Наличие генератора исходного кода;
12. Наличие макроязыка для ускорения работы с НП.

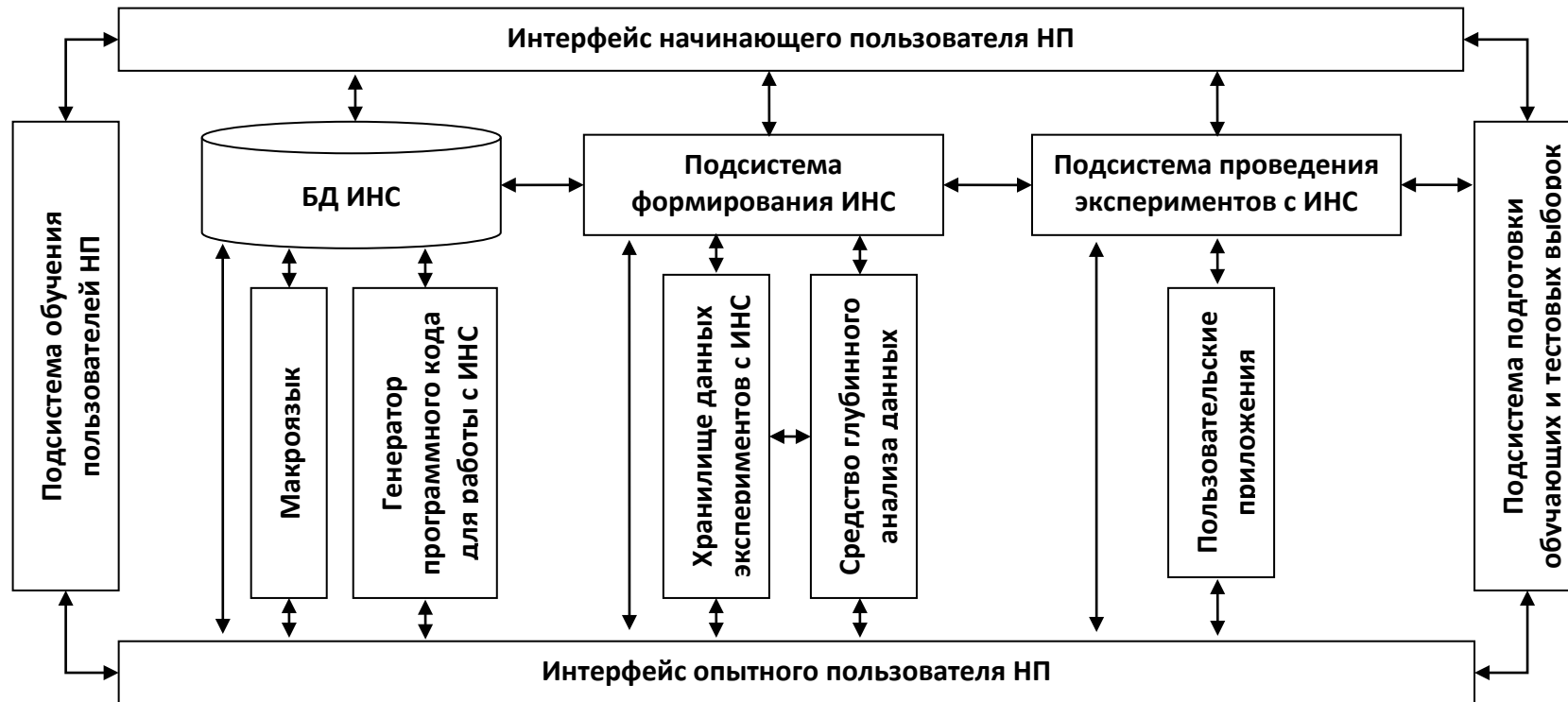


Рис. 9.4. Архитектура универсального НП

Вопросы для самопроверки

1. Что понимается под обучением ИНС? Какую роль оно играет в нейротехнологиях?
2. Какие существуют подходы к представлению результатов обучения ИНС?
3. Перечислите основные классы НП.
4. Назовите основные модули, входящие в архитектуру универсального НП.
5. По каким критериям сравнивают универсальные НП?

ЛЕКЦИЯ 9. ТЕХНОЛОГИЯ БАЗ ЗНАНИЙ (БЗ) И СИСТЕМЫ УПРАВЛЕНИЯ ЗНАНИЯМИ (СУЗ)

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 БЗ 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).

БЗ подразделяются на **замкнутые** и **открытые**.

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, всегда соотносят ее со знаниями о некоторой ПрО. 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.

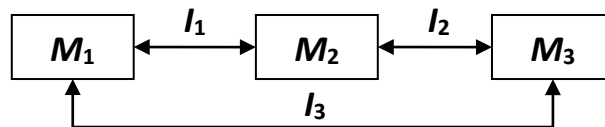


Рис. 10.1. Обобщенная структура БЗ

Математически БЗ (рис. 10.1) представляется шестеркой $(M_1, M_2, M_3, I_1, I_2, I_3)$, 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 $M_1 = (M_{11}, M_{12})$, где 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 $M_3 = (M_{31}, M_{32}, M_{33})$, 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: $I_1 = (I_{11}, I_{12})$, $I_2 = (I_{21}, I_{22})$, $I_3 = (I_{31}, I_{32})$, где 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. В настоящее время для организации M_1 используется технология объектно-ориентированных БД. База фактов M_2 , как правило, реализуется на основе технологии реляционных БД. Для построения базы метазнаний M_3 в последние годы все чаще используются семантические сети и онтологии.

СИСТЕМА ОПЕРАЦИЙ ДЛЯ РАБОТЫ СО ЗНАНИЯМИ В БЗ

Рассмотрим систему операций для работы со знаниями в БЗ на примере обобщенной МПЗ о ПрО M_4 .

Система операций для работы со знаниями в БЗ является **многоуровневой**:

- 1. Интерфейсные операции**, обеспечивающие ввод и коррекцию знаний в БЗ в процессе диалога с пользователем интеллектуальной системы или приема информации из иных источников.
- 2. Элементарные операции**, отражающие специфику взаимосвязи базисных компонентов информационных структур (вещей, свойств и отношений).
- 3. Комплексные операции**. К ним относятся операции верификации БЗ (выявление ошибок и неточностей, разрешение противоречий), а также операции поиска, извлечения, пополнения и систематизации знаний.

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).

Комплексные операции

Верификация знаний

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 ИС;**
- **проверку выполнения фиксированного множества ограничений целостности.**

Методы интеллектуальной верификации в модели *M4*, подразделяют на четыре класса:

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.

Разрешение противоречий в БЗ

В рамках модели *M4* противоречие соотносится с описанием вещи и имеет место при наличии в нем семантически несовместимых элементов. Традиционно рассматриваются **бинарные противоречия**, из-за несовместимости пары свойств или отношений.

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».

Наследование в БЗ

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 такой интерпретации уровня информационных структур M_4 нетождественные объекты O_i и O_j могут быть соседями только в 2-х случаях:

- соответствующая O_i вещь A_i определяется через свойства или отношения, задаваемые посредством ссылки на объект O_j ;
- 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 .

СУЗ

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.

Понятие «**управление знаниями**» появилось в середине 90-х годов прошлого века. Возникновение этого направления интеллектуальных информационных технологий вызвано потребностями пользователей корпоративных ИС.

Традиционные корпоративные ИС оперируют не знаниями, а данными — документами, записями в БД, выборками, отчетами и т.п.

Управление знаниями рассматривается как совокупность процессов, управляющих созданием, распространением, обработкой и использованием знаний в рамках организации.

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 разделяют 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.**

В число задач СУЗ входит поддержка процессов:

- явного выражения (фиксации) знаний специалистов;
- формализации и автоматизированного извлечения знаний из ИР.

СУЗ – интегрирующая технология, объединяющая в комплекс множество информационных технологий (как традиционных, так и интеллектуальных).

Информационные технологии, которые объединяются в единый комплекс технологией СУЗ:

- 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).

Существующие в настоящее время продукты (*Fulcrum, Documentum i4, Knowledge Station*), относимые их разработчиками к классу СУЗ, 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*)**, структура которой приведена в табл. 9.1.

Таблица 9.1.

Структура корпоративной памяти

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, схемы)	БД, файлы	БЗ

Внедрение СУЗ в организациях, значительное число сотрудников которых занято обработкой информации, приносит ощутимый экономический эффект. СУЗ позволяет ежедневно экономить в среднем 40-50 мин. рабочего времени одного сотрудника, что эквивалентно повышению производительности труда на 8-10%. Общий выигрыш от использования СУЗ составляет 8-10% от соответствующего фонда заработной платы. СУЗ, стоимость которой равна месячному фонду заработной платы, окупится примерно за год.

ТЕХНОЛОГИИ ХРАНИЛИЩ ДАННЫХ И ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

Хранилище данных (Б. Инмоном) – предметно-ориентированное, привязанное ко времени и неизменяемое собрание данных для поддержки принятия управляющих решений.

Хранилище 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.

Объем данных в хранилище как минимум на порядок превосходит объемы данных в оперативных БД (так называемых *OLTP*-системах: *On-Line Transaction Processing* – оперативная обработка транзакций).

Большей сложностью отличаются и запросы к хранилищу. Необходима высокая производительность обработки запросов и масштабируемость алгоритмов.

При загрузке в хранилище новых данных должна выполняться их верификация.

Хранилище данных может включать 2 или 3 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.

Наиболее распространенный тип знаний, извлекаемых с помощью технологий ИАД, – это закономерности ПрО.

В зависимости от характера закономерностей 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.

Данные в ИАД представляются тремя способами: атрибутивным; структурным; полнотекстовым.

Методы ИАД подразделяют на три класса:

- Алгебраические методы.
- Статистические методы.
- Методы мягких вычислений.

Методы ИАД реализуются в трех технологиях:

- интерактивной аналитической обработки данных (*On-Line Analytical Processing — OLAP*);
- глубинного анализа данных (*Data Mining — DM*);
- визуализации данных.

ТЕХНОЛОГИЯ *OLAP* И МНОГОМЕРНЫЕ МОДЕЛИ ДАННЫХ

Технология *OLAP* ориентирована, главным образом, на обработку нерегламентированных запросов к хранилищам данных.

Основной задачей хранилища является представление данных для анализа в одном месте в рамках простой и понятной структуры.

Структура типичного хранилища данных (сплошные стрелки обозначают потоки данных, пунктирные – метаданных) приведена на рис. 9.1.

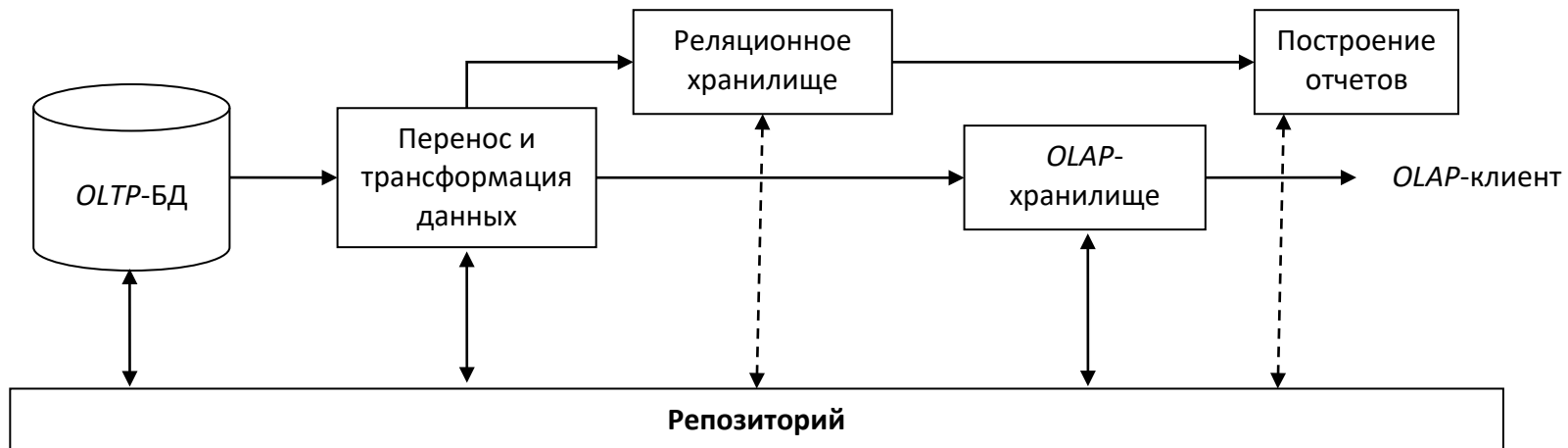


Рис. 9.1. Хранилище данных

Основная цель анализа данных — качественная и количественная оценка достигнутых результатов и (или) динамики деятельности компании.

Принципы *OLAP* 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.

На пересечении осей-измерений (*dimensions*), т.е. в ячейке гиперкуба, содержатся данные, количественно характеризующие анализируемый процесс. Эти данные называются мерами (*measures*) или показателями.

В процессе анализа выполняются операции построения сечений (проекций) гиперкуба путем фиксации значений наборов атрибутов-координат.

Многомерность в *OLAP*-приложениях воплощается в рамках 2-х или 3-х уровневой 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. Он обычно встраивается в **OLAP-клиент** или в **OLAP-сервер**;

Третий уровень реализует физическую организацию хранения многомерных данных. В рамках него для поддержки многомерных моделей данных используются либо специальные *OLAP-СУБД*, либо обычные реляционные структуры. 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 технологии хранилищ данных важную роль играет **управление метаданными**.

Метаданные хранилищ делятся на три группы:

- Административные описывают *OLTP*-БД, служащие источниками для *OLAP*, схемы данных хранилища, измерения гиперкубов, физическую организацию данных, формы стандартных отчетов, полномочия пользователей, типовые запросы;
- Операционные отражают информацию о текущем состоянии данных, статистике функционирования;
- Бизнес-метаданные содержат словарь терминов с их определениями, описания источников и владельцев данных и т.п.

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.

Основные задачи *DM*:

- **классификация;**
- **кластеризация;**
- **поиск ассоциаций и корреляций;**
- **выявление типовых образцов на заданном множестве;**
- **обнаружение объектов данных, не соответствующих установленным характеристикам и поведению;**
- **исследование тенденций во временных рядах и др.**

В рамках *DM* для сегментирования данных применяются ИНС и методы кластерного анализа, для индуктивного вывода — деревья принятия решений, для выявления в информационных массивах часто встречающихся пар объектов — статистические и ассоциативные методы.

Процесс ИАД включает четыре основных этапа (рис. 11.2):

- 1. На первом этапе аналитик формулирует постановку задачи в терминах целевых переменных;**
- 2. На втором этапе осуществляется подготовка данных для анализа;**
- 3. На третьем этапе проводится анализ данных с помощью методов *DM*;**
- 4. На четвертом этапе осуществляется верификация и интерпретация полученных результатов (извлеченных знаний). При верификации применяется тестовый набор записей, выделенных из исходных данных и не подвергавшихся анализу.**

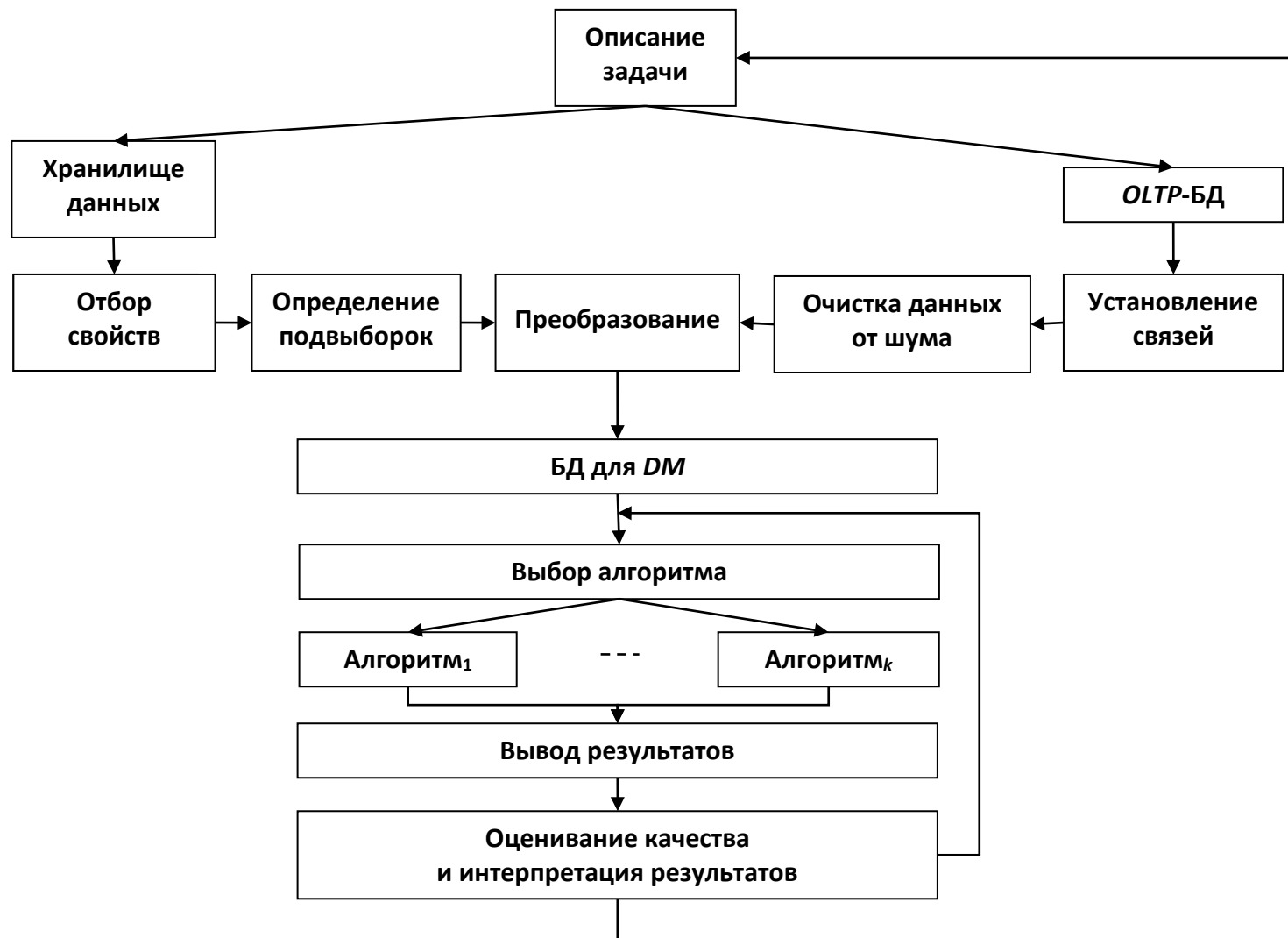


Рис. 9.2. Схема процесса ИАД на основе технологии *DM*

Пример некоторых зарубежных продуктов 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 **Poly-analyst** фирмы **Megaputer** (<http://www.megaputer.ru>).

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;
- методы эволюционного программирования.

Унификация и стандартизация технологий DM являются целями проекта **CRISP-DM** — **Cross Industry Standard Process for Data Mining** (<http://www.crisp-dm.org>). Его результаты реализуются в рамках CASE-системы для разработки средств DM.

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 *M4* отнесены ко второму уровню, и каким требованиям они отвечают?
6. Как классифицируются методы интеллектуальной верификации знаний, предусмотренные в *M4*?
7. Как в *M4* классифицируются бинарные противоречия?
8. Как определяется механизм наследования в БЗ?
9. Что понимается под управлением знаниями?
10. Что обеспечивает СУЗ для корпорации?
11. На какие слои разделяются корпоративные знания, и какие задачи СУЗ связаны с ними?
12. Какие информационные технологии интегрирует СУЗ?
13. Что является технологическим фундаментом СУЗ?
14. Что такое хранилище данных?
15. Что понимается под ИАД?
16. На какие классы подразделяются методы ИАД?
17. В каких технологиях реализуются методы ИАД?
18. Что такое *OLAP*?
19. Какие основные компоненты входят в типичное хранилище данных?
20. Что такое многомерное представление или гиперкубы данных?
21. Для чего предназначена технология *DM*? Охарактеризуйте основные этапы процесса *DM*.
22. Какие задачи решаются с помощью технологии *DM*?
23. Какие модели и методы используются в рамках *DM*?