

Федеральное агентство по образованию  
Государственное образовательное учреждение  
высшего профессионального образования  
«Санкт-Петербургский государственный  
инженерно-экономический университет»



Кафедра информационных систем в экономике

**С.С. Мигас**

# **ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ**

Конспект лекций

Специальности: 230201 – Информационные системы и технологии  
080801 – Прикладная информатика в экономике

Санкт-Петербург  
2009

*Допущено  
редакционно-издательским советом СПбГИЭУ  
в качестве методического издания*

Составитель  
канд. техн. наук, доц. *С.С. Мигас*

Рецензент  
канд. экон. наук, доц. *В.В. Шленов*

Подготовлено на кафедре  
информационных систем в экономике

Одобрено научно-методическим советом факультета  
информационных систем в экономике и управлении

Отпечатано в авторской редакции с оригинал-макета,  
представленного составителем

© СПбГИЭУ, 2009

## СОДЕРЖАНИЕ

Введение .....	5
<b>Тема 1. МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В ИИС .....</b>	<b>16</b>
1.1. Понятие «Знание» в ИИ .....	16
1.2. Основные модели представления знаний .....	25
1.3. Логическая МПЗ .....	28
1.4. Продукционная МПЗ.....	36
1.5. Фреймовая МПЗ.....	41
1.6. Семантические сети .....	45
Контрольные вопросы.....	46
<b>Тема 2. ЭКСПЕРТНЫЕ СИСТЕМЫ.....</b>	<b>47</b>
2.1. Назначения и основные свойства ЭС.....	47
2.2. Особенности построения и организации ЭС.....	50
2.3. Преимущества использования экспертных систем .....	52
2.4. Основные режимы работы экспертных систем.....	53
2.5. Отличия ЭС от традиционных программ .....	54
2.6. Классификация ЭС .....	55
2.7. Технология разработки экспертных систем .....	58
2.8. Приобретение и формализация знаний.....	65
Контрольные вопросы.....	79
<b>Тема 3. НЕЧЕТКИЕ ЗНАНИЯ.....</b>	<b>80</b>
3.1. Понятие нечеткости знаний.....	80
3.2. Неопределенность в экспертных системах.....	81
Контрольные вопросы.....	86
<b>Тема 4. НЕЙРОННЫЕ СЕТИ .....</b>	<b>87</b>
4.1. Понятие нейронной сети.....	87
4.2. Принципы организации и функционирования ИНС .....	89
4.3. Рекуррентные ассоциативные сети .....	95
4.4. Сеть с обратным распространением ошибки .....	101
4.5. Сеть Кохонена.....	106
Контрольные вопросы.....	109

<b>Тема 5. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ .....</b>	<b>110</b>
5.1. Аналитические технологии и аналитические информаци- онные системы .....	110
5.2. Методы и стадии ИАД .....	128
5.3. Задачи ИАД .....	132
5.4. Процесс ИАД .....	133
5.5. Организация создания и внедрения ИАД .....	142
Контрольные вопросы .....	146
Тестовые задания .....	147
Заключение .....	148
Список литературы .....	154
Терминологический словарь .....	155
<i>Приложение</i> Извлечение из рабочей программы дисциплины..	156

## Введение

Прогресс в сфере экономики немыслим без применения современных информационных технологий, представляющих собой основу экономических информационных систем. В качестве фундаментального можно принять понятие – *информационная система* (ИС) с дельнейшим ее расширением в сторону интеллектуализации с последующим определением понятия *Интеллектуальная ИС*.

*ИС* – это программная система, направленная на автоматизацию функционирования предприятия (бизнеса). ИС экономического объекта является основой системы управления, она постоянно видоизменяется, появляются новые информационные потоки, обусловленные широким внедрением средств вычислительной техники и расширением производственных и финансовых связей предприятия. Функциональное назначение и тип ИС зависят от того, чьи интересы и на каком уровне она обслуживает.

*Экономическая информационная система* - это среда, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, базы данных, персонал, технические и программные средства связи. В ПРО экономических систем в настоящее время принято включать:

1. Объект управления – тот экономический объект, функционированием которого осуществляется непосредственное управление для достижения требуемого результата (например, фирма или промышленное предприятие).

2. Окружающая среда – «дружественное» окружение объекта управления, функционирование которой направлено на помощь в достижении результата объектом управления и которая действует по известным алгоритмам (например, фирмы-поставщики или организации, торгующие произведенной продукцией).

3. Совокупность внешних неконтролируемых факторов – «нейтральное» и «враждебное» окружение, деятельность которого нейтральна или враждебна объекту управления, а алгоритмы функционирования либо отсутствуют, либо неизвестны (например, конкуренты или независимый рынок).

Степень структурированности и детерминированности решаемых задач уменьшается от 1 к 3, поэтому для их описания необходимо использование стохастических и эвристических методов. К стохастическим методам относятся вопросы моделирования и их целесообразно использовать для описания «нейтрального» окружения. Для описания «враждебного» окружения необходимо использовать эвристические методы, реализуемые в рамках искусственного интеллекта.

Таким образом, интеллектуализация информационных систем необходима для реализации их основных функций – поддержки принятия экономических решений.

*Интеллектуальные информационные системы (ИИС)* — естественный результат развития обычных ИС, сосредоточили в себе наиболее наукоемкие технологии с высоким уровнем автоматизации не только процессов подготовки информации для принятия решений, но и самих процессов выработки вариантов решений, опирающихся на полученные ИС данные. ИИС способны диагностировать состояние предприятия, оказывать помощь в антикризисном управлении, обеспечивать выбор оптимальных решений по стратегии развития предприятия и его инвестиционной деятельности. Благодаря наличию средств естественно-языкового интерфейса появляется возможность непосредственного применения ИИС бизнес-пользователем, не владеющим языками программирования, в качестве средств поддержки процессов анализа, оценки и принятия экономических решений. ИИС применяются для экономического анализа деятельности предприятия, стратегического планирования, инвестиционного анализа, оценки рисков и формирования портфеля ценных бумаг, финансового анализа, маркетинга и т.д.

Технически *интеллектуальные информационные системы* – технические и программные системы, ориентированные на решение большого и очень важного класса задач, называемых неформализованными.

Система становится интеллектуальной, если в ней: данные заменяются на знания; алгоритмы функционирования заменяются на методы искусственного интеллекта.

Отличительные особенности ИИС по сравнению с обычными ИС состоят в следующем:

интерфейс с пользователем на естественном языке с использованием бизнес-понятий, характерных для ПрО пользователя;

способность объяснять свои действия и подсказывать пользователю, как правильно ввести экономические показатели и как выбрать подходящие к его задаче параметры экономической модели;

представление модели экономического объекта и его окружения в виде базы знаний и средств дедуктивных и правдоподобных выводов в сочетании с возможностью работы с неполной или неточной информацией;

способность автоматического обнаружения закономерностей бизнеса в ранее накопленных фактах и включения их в базу знаний.

ИИС особенно эффективны в применении к слабо структурированным задачам, в которых пока отсутствует строгая формализация, и для решения которых применяются эвристические процедуры, позволяющие в большинстве случаев получить решение. Отчасти этим объясняется то, что диапазон применения ИИС необычайно широк: от управления непрерывными технологическими процессами в реальном времени до оценки последствий от нарушения условий поставки товаров по импорту.

Термин «искусственный интеллект» впервые появился в 1956 году в совместной работе Массачусетского технологического института и Станфордского университета философии «О методах механического повторения процесса познания».

### **Искусственный интеллект как научное направление.**

Понятие искусственного интеллекта (ИИ) как научного направления появилось в 60-70-е годы в общем комплексе кибернетических исследований. Развитие средств вычислительной техники, связанное с ним интенсивное совершенствование программирования, расширение областей использования ЭВМ, а также наличие поверхностной аналогии между структурой ЭВМ и структурой мозга человека явились побудительными причинами возникновения этого направления.

Понятие искусственного интеллекта можно трактовать в широком и узком смыслах слова.

В широком смысле слова *искусственный интеллект* есть научное направление, в рамках которого ставятся и решаются задачи аппаратного или программного планирования тех видов человеческой деятельности, которые традиционно считаются интеллектуальными (или разумными). При этом термин интеллектуальные в применении к видам человеческой деятельности означает, что их до недавнего времени способен был осуществлять только человек. Таким образом, ИИ - это научная дисциплина. Предметом изучения этой дисциплины являются методы анализа и синтеза ЭВМ с интеллектуальными способностями.

В узком смысле слова под *искусственным интеллектом* понимается программная система, имитирующая на компьютере мышление человека.

Объект изучения в дисциплине — рассмотрение процессов решения задач, характеризующихся высокой степенью сложности и эвристикой.

В ТИИ можно выделить (рис. В.1) два основных направления их исследования: **общетеоретическое**, связанное с разработкой моделей и методов реализации отдельных аспектов получения и преобразования знаний и **прикладное**, нацеленное на разработку комплексных технологий конечных пользователей.

В рамках первого направления разрабатываются следующие технологии:

1. Технологии извлечения и представления знаний. В первом случае основной задачей является разработка методов: формального описания "признаков знаний" (поисковых образов); формализации ПрО; распознавания и сравнения образов; извлечения знаний из экспертов, статистики, текстов, "опыта" и т.п. Во втором - решаются задачи, связанные с формализацией знаний для их представления в памяти интеллектуальных систем (ИС). Решение этих задач позволяет разработчикам комплексных технологий получить ответы на три принципиально важных вопроса: какие знания необходимо представлять в ИС, кто (что) является источником этих знаний, какие методы и модели обеспечивают адекватное представление этих знаний в ИС.

2. Технологии манипулирования знаниями, решение интеллектуальных задач предполагает не только представление знаний в ИС, но и их обработку, т.е. необходимо научить ИС опериро-



вать ими. Поэтому здесь изучаются вопросы пополнения знаний на основе их неполных описаний, классификации знаний в ИС, разрабатываются процедуры и методы обобщения знаний, достоверного вывода и др.

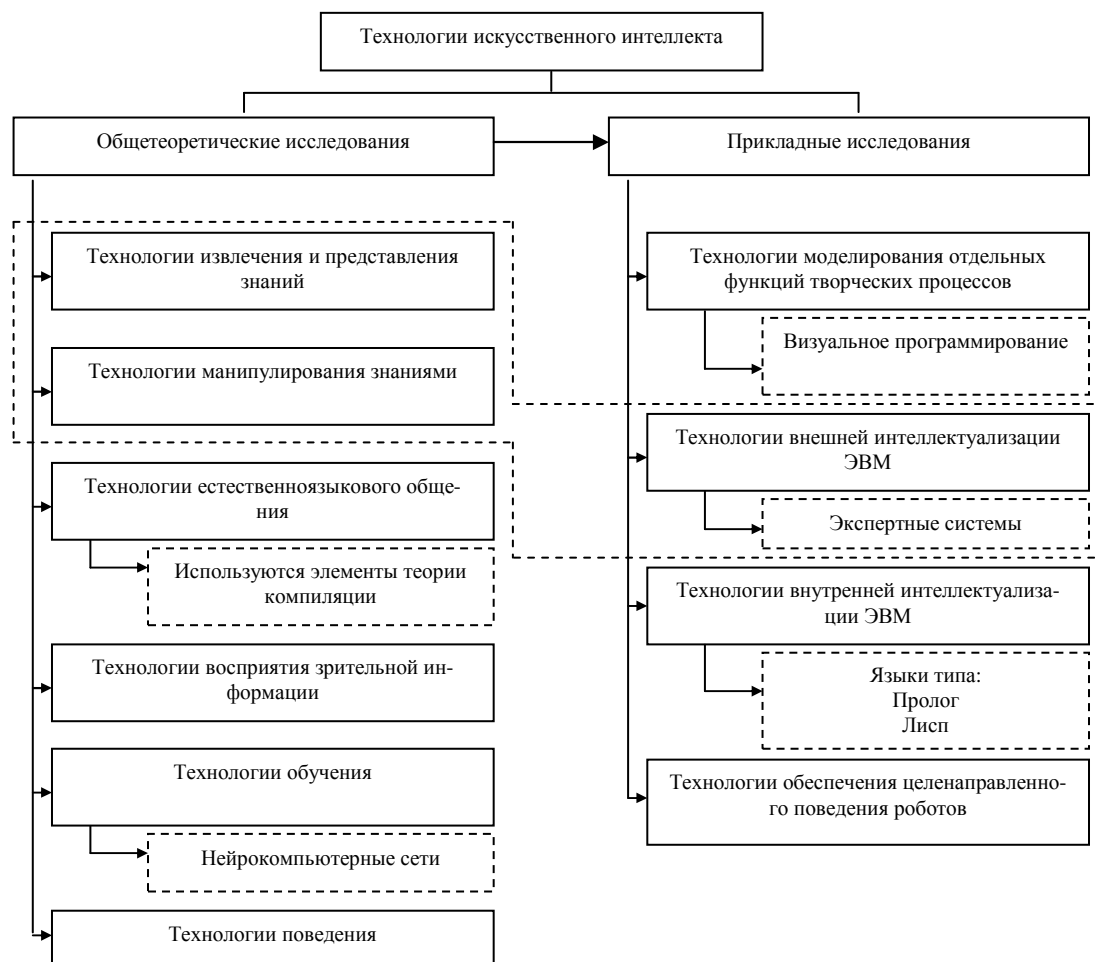


Рис. В.1. Основные направления развития ТИИ

3. Технология общения. Переход к ИС знаменует новую технологию общения конечных пользователей с ЭВМ и требует решения таких проблем, как понимание связных текстов на ограниченном и неограниченном естественном языке, понимание речи и ее синтез, разработка коммуникативных моделей "пользователь-ЭВМ", формирование объяснений и т.п. Главная цель данных исследований - обеспечение комфортных условий для общения человека и ИС.

4. Технологии восприятия. Разработка этих технологий предполагает создание методов: анализа трехмерных сцен, представления информации о зрительных образах в базе знаний ИС,

трансформации зрительных сцен в текстовые описания и обратно, а также разработку процедур когнитивной графики и др.

5. Технологии обучения. Отличительной особенностью ИС должна стать их способность решать задачи, в явном виде не представленные в БЗ, что требует наделения ИС способностью к обучению. Для этих целей необходимо: создать методы формирования условий задачи по описанию проблемной ситуации или по наблюдению за этой ситуацией, обеспечить переход от известного решения частных задач к решению общей задачи, наделить ИС способностью декомпозировать исходную задачу на более мелкие, решение которых известно, разработать нормативные и декларативные модели самого процесса обучения, создать теорию подражательного поведения и др.

6. Технологии поведения. Взаимодействие ИС со средой требует разработки специальных поведенческих процедур, которые бы позволили им адекватно реагировать на те или иные изменения в среде. Такое взаимодействие предполагает создание моделей целесообразного, нормативного и ситуативного поведения, а также разработку методов многоуровневого планирования и коррекции планов в динамических ситуациях.

Второе направление, по существу, связано с использованием результатов общетеоретических исследований на практике. Практические исследования в области ТИИ ведутся в следующих аспектах:

1. Технологии моделирования на ЭВМ отдельных функций (творческих процессов), реализуемых человеком: игровые задачи, автоматическое доказательство теорем, автоматический синтез программ, анализ и синтез элементов творчества человека и др.

2. Технологии внешней интеллектуализации ЭВМ. Исследования в данном направлении привели к созданию: интеллектуальных информационно-поисковых систем, интеллектуальных пакетов прикладных программ и расчетно-логических систем, систем поддержки принятия решений (СППР), экспертных систем (ЭС), а также к созданию комплексного диалогового интерфейса.

В системах управления в основном получили распространение СППР и ЭС. Системы поддержки принятия решений, по существу, являются человеко-машинными системами интеллекту-

альной поддержки, которые позволяют ЛПР использовать данные, знания, объективные и субъективные модели для анализа и решения слабоструктуризованных проблем. Модель СППР, соответствующая данному определению, включает в себя: интерфейс "пользователь-система", который содержит средства для генерации и управления диалогом, и блоки анализа проблем и принятия решений. Последние содержат процедуры и методы, позволяющие сформулировать поставленную проблему, с помощью баз данных (БД), моделей (БИ) и знаний (БЗ) проанализировать возможности ее решения и получить результат. В СППР включаются также средства для извлечения данных и знаний, построения моделей и манипулирования ими.

Экспертные системы - это программно-аппаратные комплексы, позволяющие пользователю осуществлять деятельность в некоторой Про на основе априорно формализованных и представленных в системе знаний ведущих специалистов. В составе ЭС обычно выделяют базу данных; базу знаний; механизм вывода; механизм пополнения знаний, механизм диалога и объяснения. Наличие механизма объяснения, позволяющего получить ответ на вопросы как и почему ЭС сформировано то или иное решение. Последнее принципиально отличает ее от всех других интеллектуальных систем. Другими характерными особенностями ЭС являются ее способности: получать решения и в условиях неопределенности; объяснять ход и результат решения понятным для пользователя способом: пополнять базу знаний; ориентироваться на решение неформализованных (способ формализации до создания ЭС неизвестен) задач.

3. Технологии внутренней интеллектуализации ЭВМ они направлены на разработку ЭВМ новой архитектуры, ориентированных на обработку символьной информации: ЛИСП, ПРОЛОГ и РЕФАЛ-машины, лингвистические процессоры, машины баз данных и машины баз знаний.

4. Технологии обеспечения целенаправленного поведения роботов. Здесь создаются ЭВМ для специальных механических и энергетических систем (сенсоры, двигатели и т.п.). Работы в данном направлении особенно важны, т.к. существенно расширяют возможности человеческой деятельности в агрессивных

средах, где присутствие человека невозможно или нецелесообразно.

Схема ИС представлена на рис. В.3.

Блок «интерфейс пользователя» предназначен для связи компьютера с пользователем, для которого предпочтительным языком «разговора» является естественный язык или близкий к нему.

Следует отметить специфическое понятие «естественный язык», под которым понимается фактически искусственный язык, полученный из естественного человеческого языка (русского, английского или любого другого) путем удаления неоднозначностей (синонимов, омонимов, идиоматических выражений). Следует заметить, что такой язык в теории автоматизированных систем носит название «искусственный язык» или «информационный язык».

Основу ИС составляют блоки «база данных - БД», «база правил», «машина логического вывода - МЛВ». БД хранит исходные данные. В базе правил фиксируются знания и опыт эксперта. МЛВ выводит результат, взаимодействуя с БД и правил.

Все три блока должны быть описаны математически. Правила записываются через интерфейс эксперта (рис. В.2) в виде сложных правильно построенных формул (ППФ) при посредничестве инженера по знаниям. Форма, в которой записываются ППФ, определяется синтаксисом, а истинностное значение алгоритмов логического вывода семантикой или смыслом.

По запросу пользователя компьютер с помощью блока «объяснения» может дать ответ на вопрос КАК (с помощью набора каких правил) получен результат и ПОЧЕМУ компьютер задает пользователю уточняющие вопросы, связанные, как правило, с данными.

Для стыковки интерфейса пользователя (работающего на естественном языке) с перечисленными блоками используют лингвистический и лексический процессоры. Лингвистический процессор – непосредственно работает с естественным языком, преобразуя результат в «машинный вид», словарный состав языка.

Иногда включают и синтаксический процессор, при этом под синтаксисом понимают правила сочетания слов внутри предложения и построения предложений.

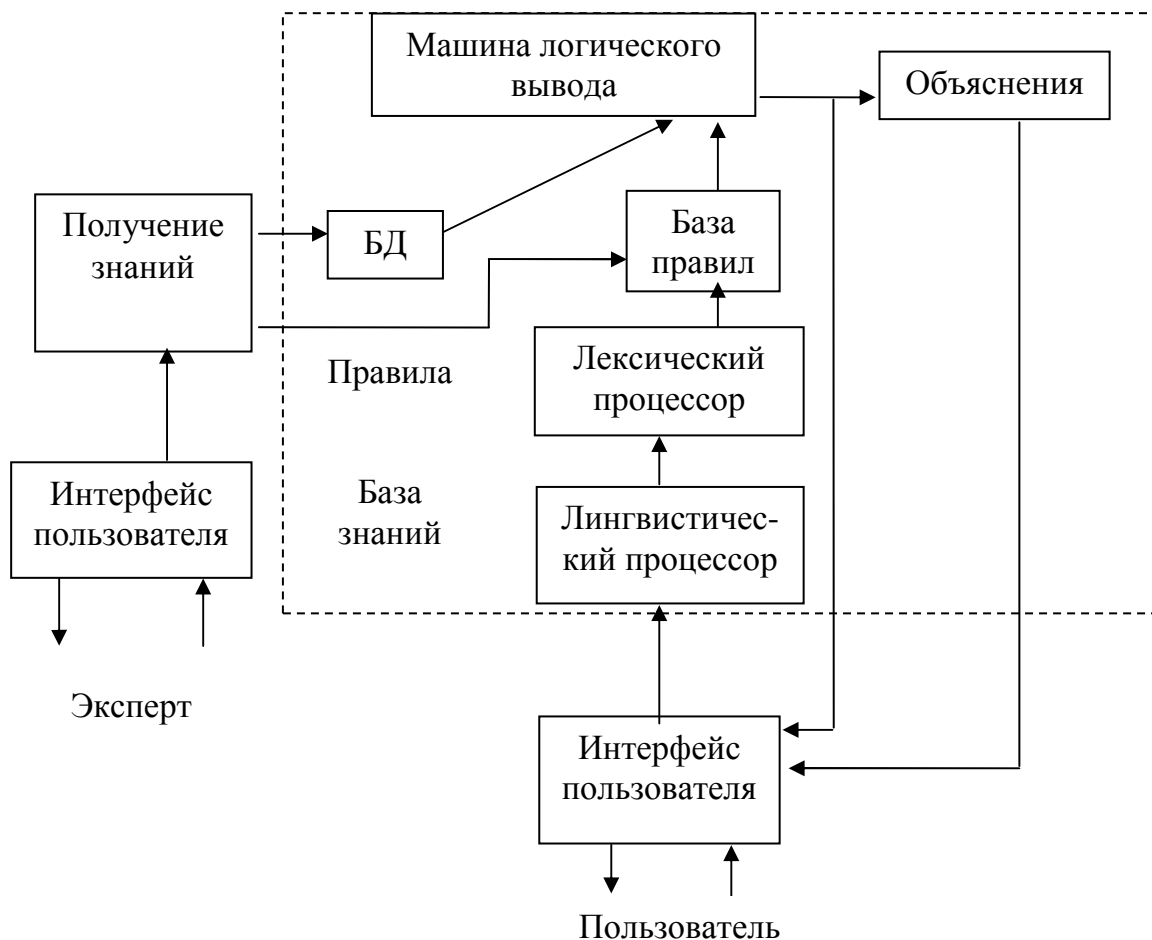


Рис. В.2. Схема интеллектуальной системы

В блоке «Получение знаний» выделяют два понятия:  
 если информация поступает из книг (документов), то говорят о выявлении знаний;

если она получается на основе работы эксперта, то говорят об извлечении знаний.

Преобразование полученной информации инженером по знаниям называется получением знаний.

Сложность здесь заключается в том, что эксперт может не владеть языками программирования, в то время как инженер по знаниям может недостаточно ориентироваться в данной предметной области.

Для описания правил (база правил) используется математическое формальное описание знаний. Для математического описания блока МЛВ применяется специальный математический аппарат – аппарат логического вывода.

Необходимо, чтобы эти два вида математического описания «понимали» друг друга.

### Классификация интеллектуальных систем

Существует множество классификаций интеллектуальных систем, из которых воспользуемся классификацией, показанной на рис. В.3.

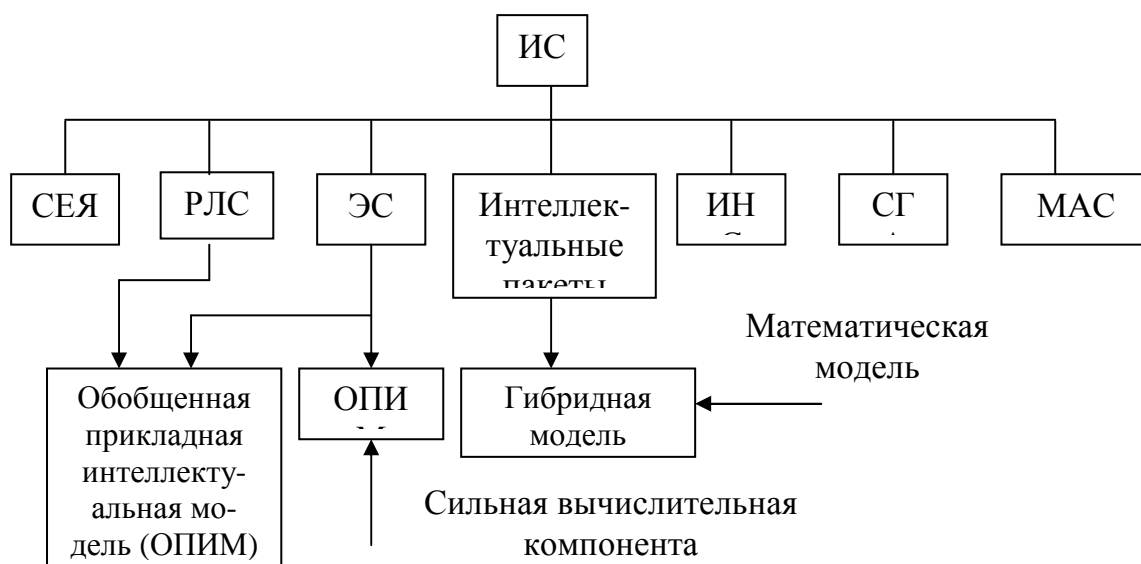


Рис. В.3. Классификация интеллектуальных систем

Системы на естественном языке (СЕЯ) специфичны и предназначены преимущественно для таких целей, как машинный перевод, генерация документов, автоматическое аннотирование и реферирование.

Экспертные системы (ЭС) предполагают высокую степень формализации процессов.

Разновидностью экспертных систем можно считать расчетно-логические системы (РЛС), оперирующими с функциями вместо правил.

Интеллектуальные пакеты в настоящее время стали оболочками экспертных систем (ЭС без базы правил).

Искусственные нейронные сети (ИНС) представляют собой фактически разновидность систем автоматического управления, использующие свойства нейрона.

Генетические алгоритмы (СГА) относятся к разновидности эволюционных эвристических методов.

Мультиагентные системы преследуют (МАС) цель согласования теорий баз данных и баз знаний.

ИС чаще всего оперируют с дискретными величинами, но все больше необходима связь дискретных и непрерывных величин, появились дополнительные классы (ОПМ, гибридные модели). Существует множество вариантов реализации гибридных систем.

Следует отметить, что на роль гибридных систем претендуют ИНС, СГА и экспертные системы реального времени (ЭСРВ).

В настоящем пособии будут рассмотрены почти все классы систем, за исключением специфических систем на естественном языке и гибридных систем.

Теория наиболее развита для широко распространенных экспертных систем, в связи с чем значительная часть учебного пособия посвящена именно этому классу систем.

Исходя из вышесказанного, целью изучения дисциплины «Интеллектуальные информационные системы» является:

ознакомление студентов с общими тенденциями развития и использования ИИ для создания ИС;

получение студентами представления о теоретических основах создания ИИС;

рассмотрение основных средств и приемов разработки ЭС как основы для построения ИИС;

получение начальных навыков использования ЭС.

Задачей дисциплины является подготовка будущих специалистов к использованию принципов ИИ и средств их реализации в составе систем поддержки принятия решений для решения экономических задач.

## Тема 1. МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В ИИС

В данной главе рассмотрено понятие «знание» в искусственном интеллекте и связанные с ним особенности. Даны структура и классификация систем, основанных на знаниях. Классификация знаний. Определены и описаны основные модели представления знаний.

### 1.1. Понятие «Знание» в ИИ

*Знания* – это совокупность сведений о сущностях (объектах, предметах) реального мира, их свойствах и отношениях между ними в определенной *предметной области*. Иными словами, *знания* – это выявленные закономерности *предметной области* (принципы, связи, законы), позволяющие решать задачи в этой области. С точки зрения *ИИ* *знания* можно определить как формализованную информацию, на которую ссылаются в процессе логического вывода.

В этом случае, под *ПрО* понимается область человеческих знаний, в терминах которой формулируются задачи и в рамках которой они решаются. Т.е. *ПрО* представляется описанием части реального мира, которое в силу своей приближенности рассматривается как ее *информационная модель*.

*Проблемная область* – это содержательное описание в терминах *ПрО* проблемы совместно с комплексом условий, факторов и обстоятельств, вызвавших ее возникновение.

В исследованиях по *ИИ* можно выделить два основных направления: *программно-прагматическое* («не имеет значения, как устроено «мыслящее» устройство, главное, чтобы на заданные входные воздействия оно реагировало, как человеческий мозг») и *бионическое* («единственный объект, способный мыслить – это человеческий мозг, поэтому любое «мыслящее» устройство должно каким-то образом воспроизводить его структуру»). В рамках данного подхода сформировалась новая наука *нейроинформатика*.

Ярким же представителем *программно-прагматического направления* можно считать *экспертные системы* – это сложные



программные комплексы, аккумулирующие *знания* специалистов-экспертов для обеспечения высокоэффективного решения неформализованных задач в узкой *предметной области*.

Приведенные выше понятия могут рассматриваться как пример системного изложения аксиоматических основ дисциплины в авторском изложении.

Общепризнанного определения знания, как и определения искусственного интеллекта, не существует. Известные трактовки этого понятия отражают его различные аспекты, поэтому приведем несколько определений.

Наиболее общее определение трактует **знание** как всю совокупность данных (информации), необходимую для решения задачи. В этом определении подчеркивается, что данные в привычном понимании также являются знаниями. Однако знания в информационном плане не ограничиваются рамками данных. В полном объеме информация, содержащаяся в знаниях, должна включать сведения о: системе понятий предметной области, в которой решаются задачи; системе понятий формальных моделей, на основе которых решаются задачи; соответствии систем понятий, упомянутых выше; методах решения задачи; текущем состоянии предметной области.

Из перечисленных компонентов только последний в явном виде соответствует понятию "данные".

В целом обо всей приведенной выше информации иногда говорят, что она составляет **проблемную область** решаемой задачи.

Несмотря на сложности формулировки определения знания считается общепризнанным, что знания имеют ряд свойств, позволяющих отличать их от данных: внутреннюю интерпретируемость; внутреннюю (рекурсивную) структурированность; внешнюю взаимосвязь единиц; шкалирование; погружение в пространство с семантической метрикой; активность.

Если данные обладают этими свойствами, можно говорить о перерастании данных в знания.

**Внутренняя интерпретируемость** означает наличие в памяти ЭВМ сведений не только о значении, но и о наименовании информационной единицы. Следует отметить, что это свойство

присуще некоторым моделям представления данных, например реляционной.

**Внутренняя (рекурсивная) структурированность** отражает вложенность одних информационных единиц в другие или в самих себя. Она предусматривает установку отношений принадлежности элементов к классу, родовидовые отношения типа «часть-целое» и т.п. В целом внутренняя структурированность характеризует структуру знания.

**Внешняя взаимосвязь единиц** определяет, с какой информационной единицей имеет связь данная информационная единица и какова эта связь. С помощью этого свойства устанавливается связь различных отношений, отражающих семантику и прагматику связей понятий, а также отношений, отражающих смысл системы в целом.

Отдельные информационные единицы не могут описывать динамические ситуации, когда некоторые факты, содержащиеся в структуре одной единицы, вступают в ситуативную связь с фактами или явлениями, описанными в структуре другой единицы. Для описания таких связей используются специальные информационные элементы, в которых указываются имена взаимосвязанных информационных единиц и имена существующих отношений.

**Шкалирование** означает использование шкал, предназначенных для фиксации соотношения различных величин. Прежде всего шкалирование необходимо для фиксации соотношений качественной информации.

**Погружение в пространство с семантической метрикой** используется для задания меры близости информационных единиц. Это свойство можно проиллюстрировать на следующем примере. Пусть разработано несколько вариантов построения системы связи. Требуется определить, насколько структура существующей системы связи близка к одному из имеющихся вариантов. Для этого можно использовать метод матриц сходства - различия, в соответствии с которым матрицы заполняются оценками попарного сходства и различия элементов структур реальной системы связи и рассматриваемого варианта. В качестве оценок обычно выступают числа в диапазоне от -1 до +1 при условии, что -1 характеризует полное различие, а +1 - полное сходство.

На основании матриц сходства - различия определяется степень сходства текущей ситуации с заранее заданной (планируемой).

**Активность** знаний выражается в возможности вызова той или иной процедуры в зависимости от структуры, сложившейся между информационными единицами.

Активность знаний обусловлена тем, что в отличие от обычных программ, в которых процедуры играют роль активаторов данных, в интеллектуальных системах определенная структура данных активизирует выполнение той или иной процедуры. Практически это осуществляется включением в состав информационной единицы элемента, содержащего имя процедуры, или представлением знаний в виде правил, причем правила записываются в следующем виде: "если произошли события  $A_1$  и  $A_2$  и ... и  $A_N$ , то необходимо выполнить процедуру В". Использование правил значительно упрощает объяснение того, как и почему получено то или иное заключение (вывод).

Перечисленные особенности информационных единиц определяют ту грань, за которой данные превращаются в знания, а базы данных перерастают в *базы знаний* (БЗ). Совокупность средств, обеспечивающих работу со знаниями, образует *систему управления базой знаний* (СУБЗ). В настоящее время не существует баз знаний, в которых в полной мере были бы реализованы все пять особенностей знаний.

*Базы данных* фиксируют экстенциональную семантику заданной проблемной области, состояние конкретных объектов, конкретные значения параметров для определенных моментов времени и временных интервалов. *База знаний* определяет интенциональную семантику моделей и содержит описание абстрактных сущностей: объектов, отношений, процессов.

Если рассматривать знания с точки зрения решения задач в некоторой предметной области, то их удобно разделить на две большие категории — факты и эвристику. Первая категория указывает обычно на хорошо известные в данной предметной области обстоятельства, поэтому знания этой категории иногда называют текстовыми, подчеркивая их достаточную освещенность в специальной литературе или учебниках. Вторая категория знаний основывается на собственном опыте специалиста (эксперта) в

данной предметной области, накопленном в результате многолетней практики.

Знания можно разделить на процедурные и декларативные. Декларативные знания – это совокупность сведений о качественных и количественных характеристиках конкретных объектов, явлений и их элементов, представленных в виде фактов и эвристик. Традиционно такие знания накапливались в виде разнообразных таблиц и справочников, а с появлением ЭВМ приобрели форму информационных массивов (файлов) и баз данных. Процедурные знания хранятся в памяти ИИС в виде описаний процедур, с помощью которых их можно получить. В виде процедурных знаний обычно описывается информация о предметной области, характеризующая способы решения задач в этой области, а также различные инструкции, методики и тому подобная информация. Другими словами, процедурные знания – это методы, алгоритмы, программы решения различных задач, последовательности действий (в выбранной проблемной области) – они составляют ядро баз знаний.

Таким образом, при использовании знаний происходит переход к формуле

**знания + вывод = система.**

Работа со знаниями, иначе называемая обработкой знаний, лежит в основе всего современного периода развития ИИ. В свою очередь обработка знаний включает в себя:

*извлечение знаний* из источников (под источниками понимаются материальные средства хранения знаний, а также события и явления, но при этом считается, что человек источником не является);

*приобретение знаний* от профессионалов (экспертов);

*представление знаний*, т.е. их формализация, позволяющая в дальнейшем использовать знания для проведения логического вывода на ЭВМ;

*манипулирование знаниями*, включающее пополнение, классификацию, обобщение знаний и вывод на знаниях;

*объяснение на знаниях*, позволяющее дать ответ, как и почему проведен тот или иной вывод.

В памяти ЭВМ знания представляются в виде некоторой знаковой системы. С понятием "знак" связываются понятия "экс-

тенсионал" и "интенционал". Экстенционал знака - это его конкретное значение или класс допустимых значений. Интенционал знака - это его смысл, характеристика содержания. Интенционал знака определяет содержание связанного с ним понятия.

Соответственно различают два типа знаний: экстенциональные и интенциональные.

**Экстенциональные знания** - это набор количественных и качественных характеристик различных конкретных объектов. Они представляются перечислениями объектов предметной области, экземпляров объектов, свойств объектов. Иными словами, экстенциональные знания - это данные, хранящиеся в базах данных.

Иногда экстенциональные знания называются предметными, или фактографическими знаниями.

**Интенциональные знания** - это совокупность основных терминов, применяемых в проблемной области, и правил над ними, позволяющих получать новые знания. Интенциональные знания описывают абстрактные объекты, события, отношения.

Интенциональные знания подразделяются на декларативные, процедуральные и метазнания.

**Декларативные знания** отражают понятия проблемной области и связи между ними. Они не содержат в явном виде описания каких-либо процедур. Иначе декларативные знания называются понятийными, или концептуальными.

**Процедуральные знания** описывают процедуры, т.е. указывают операции над понятиями, позволяющие получать новые понятия. В отличие от декларативных знаний они содержат в явном виде описания процедур. Примером процедуральных знаний является программа, хранящаяся в памяти ЭВМ. Иногда процедуральные знания называются алгоритмическими.

**Метазнания** - это знания об организации всех остальных типов знаний. Иначе они называются специальными. Метазнания содержат признаки декларативных и процедуральных знаний.

**Поверхностные** — знания о видимых взаимосвязях между отдельными событиями и фактами в предметной области.

**Глубинные** — абстракции, аналогии, схемы, отображающие структуру и природу процессов, протекающих в предметной

области. Эти знания объясняют явления и могут использоваться для прогнозирования поведения объектов.

## Структура системы искусственного интеллекта

Обобщенная структура СИИ представлена на рис. 1.1.

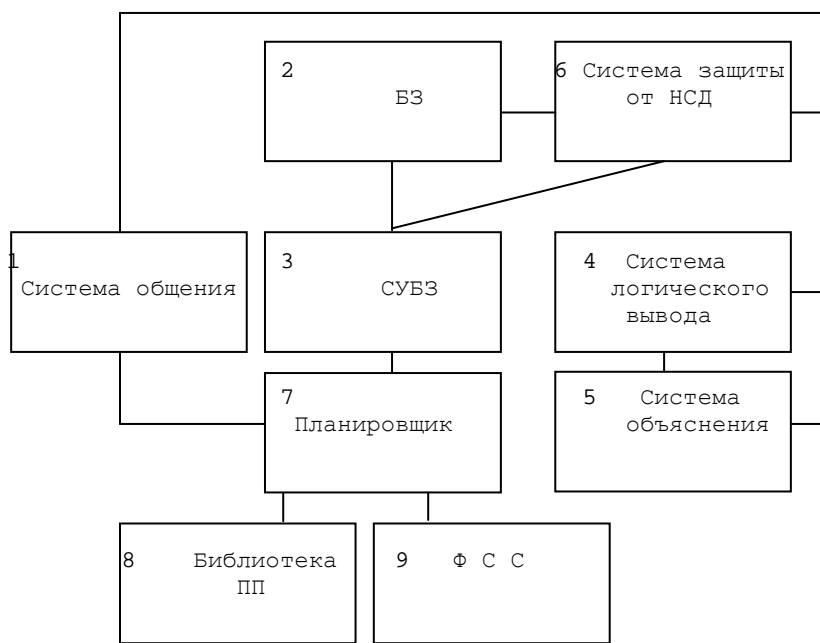


Рис. 1.1. Обобщенную структуру СИИ

**Система общения** (компонент 1) предназначена для организации взаимодействия "пользователь - СИИ".

**База знаний** (компонент 2) - предназначена для накопления и хранения знаний, представленных в формализованном виде.

Знания о проблемной области материализуются в базе знаний и представляются в памяти ЭВМ с помощью специальных лингвистических средств. База знаний - это специальным образом организованная система знаний, содержащая сведения о ПрО и предназначенная для многоаспектного использования в интересах решения задач средствами и методами ИИ.

Концепция БЗ является результатом естественного дальнейшего научно-технического развития концепции (БД) в направлении увеличения независимости содержащейся информации от обрабатывающих программ и их целостности. База знаний - это целостная и независимая модель проблемной области.

Обобщенная структура БЗ, соответствующая приведенному выше разбиению знаний по типам, представлена на рис. 1.2.

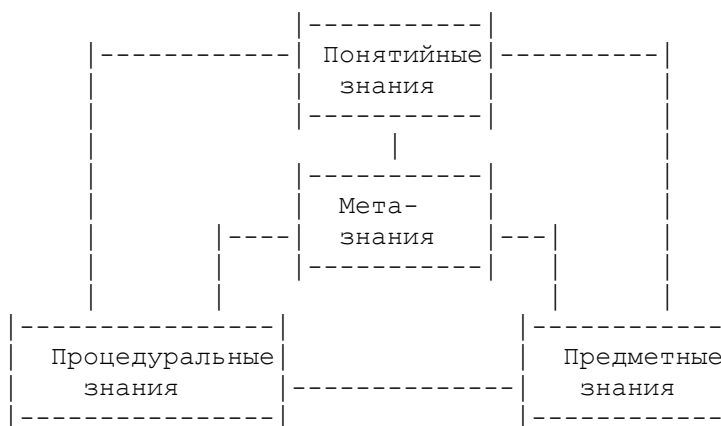


Рис. 1.2. Уровня рассмотрения построения БЗ

Так как концепция БЗ является развитием концепции БД, принципы организации БД и БЗ во многом схожи между собой. Так, по аналогии с БД можно выделить 3 уровня рассмотрения построения БЗ (с точки зрения детальности):

уровень общей организации знаний, учитывающий предметную область и характер решаемых задач (результатом представления структуры базы на этом уровне является концептуальная модель БЗ);

уровень логической организации знаний, учитывающий выбранную модель представления знаний (на этом уровне говорят о формальной модели БЗ);

уровень физической организации знаний, учитывающий выбранные средства программной и аппаратной реализации (результатом рассмотрения является физическая модель БЗ).

По аналогии с системой управления базами данных (СУБД), обеспечивающей управление и функционирование БД, управление базой знаний осуществляет СУБЗ, представляющая собой программный комплекс, который обеспечивает управление и функционирование БЗ как компонента СИИ. Она обеспечивает связь между структурами знаний, полученных от пользователя, и реальными структурами знаний, хранящимися в базе.

Основными функциями СУБЗ являются:

накопление и извлечение знаний;

поддержка естественно-языкового общения;  
обеспечение разработки ориентированного на пользователя программного обеспечения;  
поддержка интерактивного процесса разработки СИИ;  
обеспечение разработки (развития) механизма логического вывода;  
контроль достоверности и непротиворечивости знаний.

Данный список не является исчерпывающим. Кроме того, в существующих реализациях СУБЗ, в подавляющем большинстве являющихся экспериментальными, состав выполняемых ими функций различен.

Необходимо отметить, что в последнее время БЗ и СУБЗ тесно интегрируются друг с другом, образуя единую систему, которая носит название системы баз знаний, или системы, базирующейся на знаниях.

**Система управления базой знаний (СУБЗ, компонент 3)** - предназначена для осуществления операций манипулирования знаниями, содержащимися в СИИ, и их использования для решения задач.

**Система логического вывода (компонент 4)** - предназначена для осуществления процесса получения решения с использованием хранящихся в БЗ знаний (называемого логическим выводом).

**Система объяснения (компонент 5)** - позволяет получить объяснение, "как" и "почему" был проведен тот или иной логический вывод.

**Система защиты от несанкционированного доступа (компонент 6)** предназначена для защиты знаний, хранящихся в СИИ, при наличии возможности коллективного использования СИИ.

**Планировщик (компонент 7)** осуществляет формирование **функциональной семантической сети (компонент 9)**.

В библиотеке прикладных программ (компонент 8) содержатся программные процедуры, позволяющие решать частные, более мелкие задачи.

Общими компонентами для всех типов СИИ (и единственными для ИИПС) являются компоненты 1 - 4. Остальные имеют



место только в отдельных типах СИИ: 5-й - в ЭС; 6-й - в РЭС и РЛС; 7, 8, 9-й - в ИППП и РЛС.

Логический и эвристический методы рассуждения в ИИС. Рассуждения на основе дедукции, индукции, аналогии.

## **1.2. Основные модели представления знаний**

Центральной проблемой обработки знаний, а также создания БЗ (и ИИС вообще) является проблема представления знаний. Проблема представления знаний – это проблема представления взаимосвязей в конкретной предметной области в форме, понятной системе искусственного интеллекта. Представление знаний – это их формализация и структурирование (в целях облегчения решения задачи), с помощью которых отражаются характерные признаки знаний: внутренняя интерпретируемость, структурированность, связность, семантическая метрика и активность. *Представление знаний* - это процесс (способ) описания знаний человека о проблемной области посредством выражений на формальном языке, называемом *языком представления знаний*.

При проектировании модели представления знаний следует учитывать такие факторы, как – однородность представления и простота понимания. Однородность представления приводит к упрощению механизма управления логическим выводом и управлением знаниями. Простота понимания предполагает доступность понимания представления знаний и экспертам, и пользователем системы. В противном случае затрудняется приобретение знаний и их оценка.

Способ представления знаний определяет, каким образом знания описываются в памяти ЭВМ, а также каковы возможности БЗ. Для того, чтобы ЭВМ имела возможность манипулирования знаниями о проблемной области, они должны быть представлены в виде модели.

*Модель представления знаний (МПЗ)* - это способ и результат формального описания знаний в БЗ. Она должна быть понятной пользователю и обеспечивать однородность представления знаний, за счет чего упрощаются управление знаниями и логический вывод, а также удовлетворять ряду других требований.

К настоящему времени разработано достаточно много различных МПЗ, и работа по созданию новых моделей продолжается. Однако наибольшее распространение получили четыре модели: модель семантической сети, фреймовая, продукционная и логические – рис. 1.3.

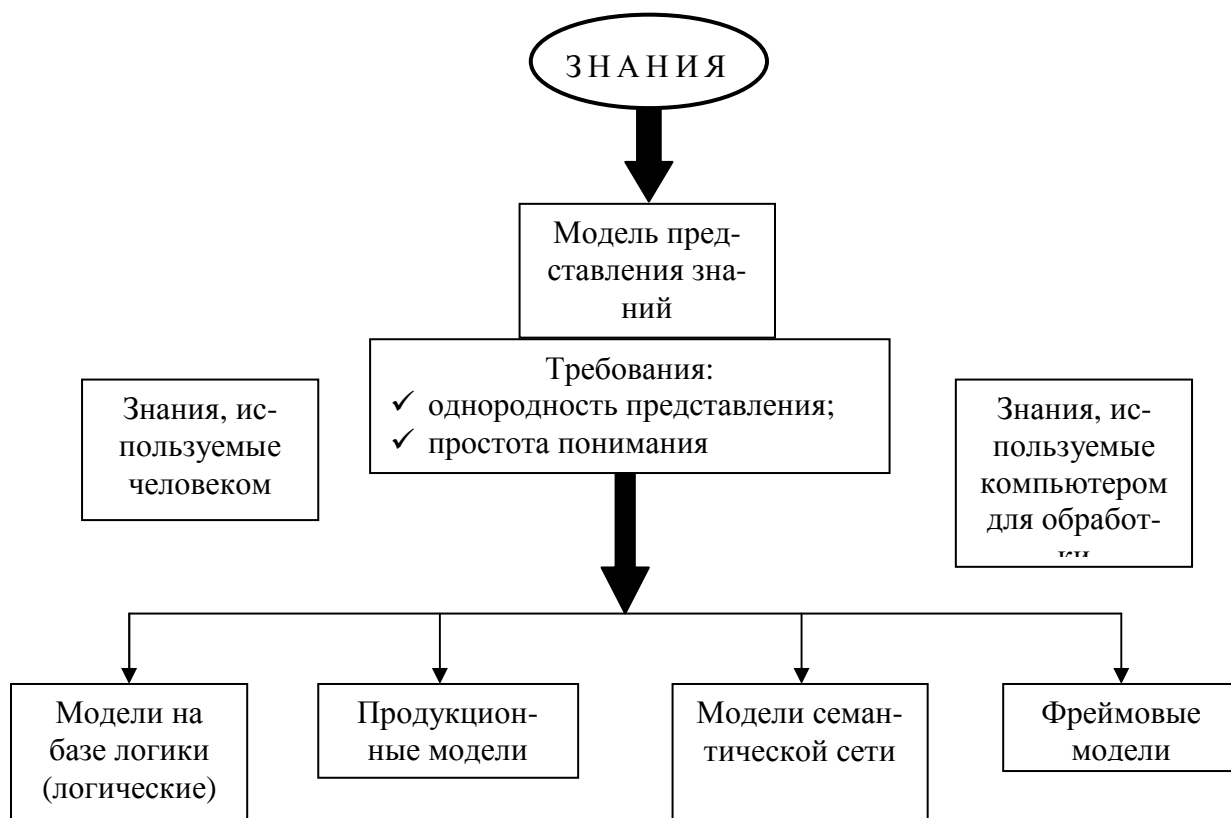


Рис. 1.3. Основные МПЗ

В основе использования МПЗ лежит аксиоматический метод. Аксиоматический метод в любой науке состоит в том, что выделяется некоторое небольшое множество истинных утверждений, опираясь на которые можно вывести все истинные утверждения данной науки. Классическим примером аксиоматического метода является аксиоматическое построение геометрий Евклида и Лобачевского, каждая из которых базируется на ряде постулатов. Следует отметить, что одну и ту же теорию можно строить, исходя из различного набора аксиом.

Таким образом, необходимо:

1). Построить алфавит теории, т.е. задать счетное множество символов (буквы и знаки включены в этот термин) и определить множество объектов языка - выражений. Под выражением имеет-

ся ввиду конечная последовательность символов языка. Понятно, что символ может несколько раз появляться в выражении, а может и не появляться (роль формальных выражений в символическом языке аналогична роли слов в обычном языке).

2). Выделить подмножество таких выражений, которые будем называть формулами (обычно имеется хорошо разработанная процедура, позволяющая по данному выражению определить, является ли оно формулой). С формулами не будем связывать никакого значения (смысла); никакого значения не будем связывать и с входящими в них буквами и символами. Все операции будут формализованы.

3). Из бесконечного множества истинных формул (тавтологий) выделим небольшую группу (1 ÷ 10) так называемых аксиом теории (как правило, всегда имеется возможность эффективно выяснить, является ли данная формула аксиомой). За аксиомы берутся некоторые тавтологии, из которых по формальным правилам выводятся все остальные тавтологии.

4). Указать конечное множество отношений между формулами, которые называют правилами вывода. Правила вывода сопоставляют некоторым последовательностям формул новые формулы. Записывают правила вывода в форме фигуры, где формулы, стоящие над чертой называются посылками, а формулы, стоящие под чертой, называется следствием посылок по данному правилу вывода. С помощью правил вывода из аксиом получают новые истинные формулы, называемые теоремами.

Доказательством (выводом) называется конечная последовательность формул  $A_1, A_2, \dots, A_n$  такая, что каждая  $A_k$  есть либо аксиома теории, либо непосредственное следствие каких-либо предыдущих формул по одному из правил вывода. Теоремой называется такая формула  $A$  теории, что существует вывод, в котором последней формулой является формула  $A$ . Теоремы, как правило, выражаются равенствами, импликациями и эквивалентностями. Доказательство теорем превращается в последовательность таких формул, и построение формальных доказательств можно поручить ЭВМ.

К МПЗ предъявляются требования полноты и непротиворечивости.

### 1.3. Логическая МПЗ

Логические МПЗ - это модели, основанные на правилах формальной логики. Все логические МПЗ представляются четверкой (формальной системой):

$$M = \langle T, P, A, F \rangle,$$

где  $T$  - алфавит (множество базовых элементов);

$P$  - множество правил построения синтаксически правильных выражений;

$A$  - априорно истинные выражения (аксиомы);

$F$  - правила вывода новых истинных выражений в рамках  $M$ .

В качестве  $T$  и  $P$  в логических МПЗ в настоящий момент чаще всего используются исчисление высказываний и исчисление предикатов первого порядка, преобразованные к определенному виду – хорновским дизъюнктам,  $F$  – метод линейных резолюций.

Для описания множеств  $T$  и  $P$  необходимо ввести ряд специфических расширения логики предикатов.

Элементарной конъюнкцией или конъюнктом называется конъюнкция литералов. Элементарной дизъюнкцией или дизъюнктом называется дизъюнкция литералов.

Пропозициональная формула  $A$  имеет дизъюнктивную нормальную форму, если она представляет собой дизъюнкцию, состоящую из одного или нескольких конъюнктов.

Формула  $A$  имеет конъюнктивную нормальную форму, если она представляет собой конъюнкцию, состоящую из одного или нескольких дизъюнктов.

Дизъюнктивная (конъюнктивная) нормальная форма называется совершенной, если каждый ее конъюнкт (дизъюнкт) содержит ровно один литерал каждой пропозициональной буквы этой формулы.

Все процедуры опровержения применяются к стандартной форме формулы. Бескванторную формулу несложно представить в конъюнктивной нормальной форме.

Тогда доказательство любой теоремы состоит в том, что в рамках данной теории к ее аксиомам добавляется логическое выражение, определяющее теорему, а затем подтверждается общезначимость полученной системы. Таким образом, если удастся найти алгоритм, который обеспечит подтверждение общезначимости, то проблема автоматизации доказательства будет решена.

Попытки решить эту задачу предпринимал еще Г.Ф. Лейбниц (1646 - 1716), затем данной проблемой занимались Д. Пеано и математики школы Д. Гильберта. А. Чёрчем и А. Тьюрингом (независимо друг от друга) было показано, что не существует никакой общей разрешающей процедуры (никакого алгоритма) проверяющей общезначимость формул в логике первого порядка. Говорят, что проблема проверки общезначимости любой формулы в логике первого порядка алгоритмически неразрешима.

Эрбраном (J. Herbrand) в 1930 году был получен алгоритм, позволяющий за конечное число шагов найти интерпретацию, которая опровергает некоторую логическую формулу. Если формула общезначима, то, как известно, не существует её опровергающей интерпретации. Исходя из этого, обычно вместо доказательства общезначимости некоторой формулы доказывается её противоречивость. Конечность такого процесса и гарантируется методами Эрбрана. Целью является доказательство противоречивости формулы. Формула невыполнима (противоречива), если она ложна на всех её интерпретациях. Нет необходимости рассматривать все интерпретации (что в принципе и невозможно). Достаточно рассматривать лишь все интерпретации над некоторой специальной областью, называемой Эрбрановским универсумом множества дизъюнктов этой формулы (множества  $S$ ), противоречивость которого следует зафиксировать.

Эрбрановский универсум определяется так:

$H_0$  - множество констант, встречающихся в  $S$  (если  $H_0 = \emptyset$ , то принять его равным множеству, состоящему из какой-либо произвольной константы - например,  $a$ );

$H_{i+1}$  для  $i = 0, 1, 2, \dots$  есть объединение  $H_i$  и множества термов  $f^n(t_1, \dots, t_n)$ , где  $f^n$  - функциональный ( $n$  - мерный) символ из  $S$ , и  $t_j \in H_i$ ,  $j=1, 2, \dots, n$ .

Когда в логическом выражении отсутствуют переменные, его называют основным выражением (основным примером). Таким образом, можно говорить об основных термах, атомах, литеррах и дизъюнктах.

Пусть  $S$  - множество дизъюнктов, тогда множество основных атомов вида  $P^n(t_1, \dots, t_n)$  для всех  $n$ -местных предикатов  $P^n$ , встречающихся в  $S$ , где  $t_1, \dots, t_n$  - элементы Эрбрановского универсума множества  $S$ , называется Эрбрановским базисом для множества  $S$ .

Пусть  $S$  - множество дизъюнктов,  $H$  - Эрбрановский универсум  $S$  и  $I$  - интерпретация  $S$  над  $H$ . Говорят, что  $I$  есть  $H$  - интерпретация множества  $S$ , если она удовлетворяет следующим требованиям:

- 1).  $I$  отображает все константы из  $S$  в самих себя;
- 2). в  $I$  через  $f^n$  отображается функция, которая отображает элемент из  $H_n$  в  $H$  (т.е.  $(h_1, \dots, h_n)$  отображается в  $f^n(h_1, \dots, h_n)$ ).

Пусть  $A = \{A_1, A_2, \dots, A_n\}, \dots$  - Эрбрановский базис  $S$ , тогда  $H$  - интерпретацию  $I$  удобно представить в виде

$$I = \{m_1, m_2, \dots, m_n, \dots\}, \text{ где } m_j = A_i \text{ или } \neg A_i.$$

Можно показать, что множество дизъюнктов  $S$  невыполнимо тогда и только тогда, когда для каждой  $H$ -интерпретации  $I$  существует по крайней мере один такой основной пример  $C'$  некоторого дизъюнкта  $C$  в  $S$ , что  $C'$  не выполняется в  $I$ .

Пусть  $S$  - множество дизъюнктов и  $A$  - его Эрбрановский базис. Семантическое дерево для  $S$  есть растущее вниз дерево  $T$ , в котором каждому ребру приписано конечное множество атомов или отрицаний атомов из  $A$  таким образом, что:

1). Из каждого узла  $N$  выходит только конечное число ребер  $l_1, \dots, l_n$ . Пусть  $Q_i$  - конъюнкция всех литер, приписанных к  $l_i$  ( $i = 1, \dots, n$ ), тогда  $Q_1 \vee Q_2 \vee \dots \vee Q_n = \text{true}$  (общезначимая формула).

2). Пусть для каждого узла  $N$   $I(N)$  есть объединение всех множеств, приписанных ребрам ветви, ведущей к  $N$ . Тогда  $I(N)$  не содержит контрарных пар (под контрарной парой понимается какая-нибудь литера и её отрицание - например,  $A$  и  $\neg A$ ).

Семантическое дерево для  $S$  будет полным тогда и только тогда, когда для каждого  $i$  и каждого конечного узла  $N$   $I(N)$  содержит либо  $A_i$ , либо  $\neg A_i$ .

Нетрудно заметить, что в полном дереве любая  $I(N)$  есть представление интерпретации для  $S$ . Если  $S$  - невыполнимо, то оно не может быть истинным в каждой из интерпретаций (может быть - частных), представленных некоторыми узлами дерева.

Узел  $N$  называется опровергающим, если  $I(N)$  опровергает некоторый основной дизъюнкт в  $S$ , но для любого предшествующего  $N$  узла  $N'$   $I(N')$  не опровергает никакого основного дизъюнкта в  $S$ . Семантическое дерево  $T$  будет замкнутым (закрытым) тогда и только тогда, когда каждая его ветвь оканчивается опровергающим узлом. Тогда теорема Эрбрана может быть сформулирована следующим образом:

Вариант 1: множество дизъюнктов  $S$  невыполнимо тогда и только тогда, когда любому полному семантическому дереву  $S$  соответствует конечное замкнутое семантическое дерево.

Вариант 2: множество дизъюнктов невыполнимо тогда и только тогда, когда существует конечное невыполнимое множество  $S'$  основных примеров дизъюнктов из  $S$ .

### **Метод Девиса и Патнема**

Следуя второму варианту формулировки теоремы Эрбрана, для доказательства противоречивости множества дизъюнктов  $S$  необходимо создать программу, которая:

порождает множества  $S_0', S_1', \dots, S_n', \dots$  основных примеров дизъюнктов из  $S$ ;

по мере создания этих множеств пытается определить их невыполнимость.

Данный процесс чрезвычайно неэффективен, однако существует метод Девиса и Патнема, позволяющий улучшить описанный алгоритм.

1). Правило тавтологии. Вычёркиваются все тавтологические основные дизъюнкты из  $S$ .

2). Правило однолитерных дизъюнктов. Пусть  $L \in S$  - однолитерный дизъюнкт. Построим  $S'$  путем удаления из  $S$  всех основных дизъюнктов, содержащих  $L$ . Если  $S'$  пусто, то  $S$  выполнимо; иначе строим  $S''$ , выбрасывая из  $S'$  вхождение  $\wedge L$  (если  $\wedge L$  - единичный дизъюнкт, то при вычеркивании из  $S'$  он превращается в  $\vee$ ).

3). Правило чистых литер. Литера  $L$  называется чистой тогда и только тогда, когда литера  $\neg L$  не появляется ни в одном из дизъюнктов  $S$ . Все дизъюнкты, содержащие чистую литеру  $L$ , вычеркиваются из  $S$ .

4). Правило расщепления. Пусть  $S$  можно представить в виде  $(A_1 \vee L) \wedge \dots \wedge (A_m \vee L) \wedge (B_1 \vee \neg L) \wedge \dots \wedge (B_n \vee \neg L) \wedge R$ , где  $A_i$  и  $B_i$  свободны от  $L$  и  $\neg L$ . Если  $S_1 = A_1 \wedge \dots \wedge A_m \wedge R$ ,  $S_2 = B_1 \wedge \dots \wedge B_n \wedge R$ , то  $S$  - невыполнимо, если и только если  $(S_1 \vee S_2)$  невыполнимо.

Применение перечисленных правил позволяет значительно улучшить алгоритм логического вывода, и на его основе создан метод резолюции.

### **Метод резолюций. Унификации и подстановки**

По своей сути метод резолюций - это обобщение правила однолитерных дизъюнктов Девиса и Патнема.

Обобщением правила однолитерных дизъюнктов является следующее правило резолюций: для любых двух дизъюнктов  $C_1$  и  $C_2$ , если существует литера  $L_1$  в  $C_1$ , которая контрарна литере  $L_2$  в  $C_2$ , то, вычеркнув  $L_1$  и  $L_2$  из  $C_1$  и  $C_2$  соответственно, можно построить дизъюнкцию оставшихся дизъюнктов, и построенный дизъюнкт есть резольвента  $C_1$  и  $C_2$ .

Для распространения метода резолюций на логику первого порядка необходимо ввести понятия подстановки и унификации. Например, в дизъюнктах:  $C_1: P(x) \vee Q(x)$ ,  $C_2: \neg P(f(x)) \vee R(x)$ . контрарных пар литер не существует. Если заменить в  $C_1$  "x" на "f(a)", а в  $C_2$  - "x" на "a", то  $C_1': P(f(a)) \vee Q(f(a))$ ,  $C_2': \neg P(f(a)) \vee R(a)$ . Теперь уже можно получить резольвенту  $C_1'$  и  $C_2'$  -  $C_3': Q(f(a)) \vee R(a)$ .

Поскольку  $C_1'$  и  $C_2'$  - основные примеры дизъюнктов  $C_1$  и  $C_2$ , а  $C_3'$  - резольвента этих основных примеров, то подставив в  $C_1$  "f(x)" вместо "x":  $C_1'': P(f(x)) \vee Q(f(x))$ ,  $C_2'': \neg P(f(x)) \vee R(x)$ , а их резольвентой будет  $C_3'': Q(f(x)) \vee R(x)$ . Легко заметить, что  $C_3'$  - это основной пример дизъюнкта  $C_3''$ , поэтому  $C_3''$  является наиболее общим дизъюнктом.

Тогда подстановка - это конечное множество вида  $\{t_1/v_1, \dots, t_n/v_n\}$ , где каждая  $v_i$  - переменная, каждый  $t_i$  терм, отлич-



ный от  $v_i$ , и все  $v_i$  различны. Если  $\{t_1, \dots, t_n\}$  - основные термы, то подстановка называется основной.

Подстановка, которая не содержит элементов, называется пустой и обозначается  $e$ .

Пусть  $\Theta = \{t_1/v_1, \dots, t_n/v_n\}$  - подстановка и  $E$  - выражение. Тогда  $E\Theta$  - выражение, полученное заменой одновременно всех вхождений переменной  $v_i$  на терм  $t_i$ .  $E\Theta$  называется примером  $E$ .

Пусть  $\Theta = \{t_1/x_1, \dots, t_n/x_n\}$  и  $\lambda = \{u_1/y_1, \dots, u_m/y_m\}$  - две подстановки. Тогда композиция  $\Theta$  и  $\lambda$  есть подстановка  $\Theta \bullet \lambda$ , которая получается из множества  $\{t_1\lambda/x_1, \dots, t_n\lambda/x_n, u_1/y_1, \dots, u_m/y_m\}$  вычеркиванием всех элементов  $t_i\lambda/x_i$ , для которых  $t_i = x_i$ , и всех элементов  $u_j/y_j$  таких, что  $y_j \in \{x_1, \dots, x_n\}$ .

Подстановка  $\Theta$  называется унификатором для множества  $E_1, \dots, E_k$  тогда и только тогда, когда  $E_1\Theta = E_2\Theta = \dots = E_k\Theta$ . Говорят, что множество  $E_1, \dots, E_k$  унифицируемо, если для него существует унификатор.

$\sigma$  - наиболее общий унификатор, если для любого другого унификатора существует такая подстановка  $\lambda$ , что  $\Theta = \sigma \bullet \lambda$ .

Необходимо отметить, что существует регулярный метод унификации множества выражений.

### **Хорновские дизъюнкты и линейная резолюция**

Пусть  $C_1$  и  $C_2$  - два дизъюнкта (называемые дизъюнктами - посылками), которые не имеют общих переменных. Пусть  $L_1$  и  $L_2$  - две литеры в  $C_1$  и  $C_2$  соответственно. Если  $L_1$  и  $L_2$  имеют наиболее общий унификатор  $\sigma$ , то дизъюнкт  $(C_1\sigma - L_1) \vee (C_2\sigma - L_2)$  называется (бинарной) резольвентой  $C_1$  и  $C_2$ , а литеры  $L_1$  и  $L_2$  называются отрезаемыми литерами.

В общем случае резольвентой дизъюнктов - посылок  $C_1$  и  $C_2$  является одна из следующих резольвент:

- 1). бинарная резольвента  $C_1$  и  $C_2$ ;
- 2). бинарная резольвента  $C_1$  и склейки  $C_2$ ;
- 3). бинарная резольвента склейки  $C_1$  и  $C_2$ ;
- 4). бинарная резольвента склейки  $C_1$  и склейки  $C_2$ .

Множество  $S$  дизъюнктов невыполнимо тогда и только тогда, когда существует вывод пустого дизъюнкта из  $S$ .

## **Поиск ответов на вопросы**

Выделяют вопросно-ответные системы следующих классов:

1. Класс А - вопросы, требующие ответа "да/нет". При ожидании положительного ответа строится его логическая формула, берется ее отрицание и преобразуется в множество дизъюнктов. Если после подключения этого множества дизъюнктов к множеству  $S$  получается противоречивое множество  $S'$ , то ответ "да" обоснован. При ожидании ответа "нет" вновь строимое множество дизъюнктов формируется путем добавления к  $S$  дизъюнктов, получаемых не из отрицания ответа, а из его прямой формы.

2. Класс В - вопросы, требующие ответа "кто", "где" или "при каких условиях". Результатом является не ответ типа да/нет, а результат выполненной подстановки. Естественно, в этом случае в множество дизъюнктов добавляется отрицание вопроса.

3. Класс С - вопросы, требующие ответа в виде последовательности действий. Аналогичен классу В, но результат – совокупность подстановок.

## **Клозы, факты, правила и теоремы (запросы)**

Основой любой дедуктивной системы логического вывода является формулирование (представление) знаний в виде некоторой (иногда достаточно объемной) логической формулы. Непротиворечивость формулы говорит об обоснованности этих знаний. Полученная формула знаний может быть приведена к конъюнктивной нормальной форме и затем представляться как некоторое непротиворечивое множество дизъюнктов. Подтверждение правильности суждения, сформулированного в рамках заданного формулой знаний "мира" производится следующим образом:

суждение (теорема) формулируется в виде некоторой логической формулы;

строится отрицание формулы предполагаемого суждения, которое затем приводится к виду множества дизъюнктов (обычно одного дизъюнкта);

исходное множество дизъюнктов, представлявшее знания, объединяется с множеством дизъюнктов вопроса (проверяемого суждения);

на основе метода резолюций (или какого-либо другого метода) осуществляется резолютивный вывод пустого дизъюнкта

(или каким-то другим способом определяется противоречивость полученного множества);

факт получения противоречивого множества дизъюнктов трактуется как подтверждение ожидаемой ложности отрицания утверждения-запроса, или (что то же самое) как подтверждение истинности посылки-запроса. Применяемый при этом метод резолюций гарантирует (при правильном его применении) получение решения (вывод пустого дизъюнкта).

Сущность **линейной резолюции** состоит в том, что всегда одной из посылок текущей резолюции является только что полученная резольвента. При выборе посылок самой первой резолюции вывода одной из них всегда берется целевой дизъюнкт. Таким образом, дерево вывода пустого дизъюнкта приобретает линейную структуру, что и определяет название "линейная".

Кроме того, в Пролог-системах выбирается такая структура целевого дизъюнкта и всех дизъюнктов-посылок из базы знаний, которая обеспечивает получение дизъюнктов-резольвент, содержащих лишь отрицательные литеры. Для выполнения этого условия все дизъюнкты базы знаний должны иметь лишь одну положительную литеру. Такие дизъюнкты называются **Хорновскими** (а применительно к языку Пролог подобным образом записанный дизъюнкт называется клозом).

Таким образом, в состав множества представляющих базу знаний дизъюнктов  $S$  могут входить лишь дизъюнкты вида  $\neg V_1 \vee \neg V_2 \vee \dots \vee \neg V_m \vee A$  ( $m \geq 0$ ). При  $m=0$  мы имеем дело с "вырожденным" дизъюнктом - "фактом".

В программах на Прологе используют "импликационное" представление клозов в специальной форме записи - дизъюнкт  $\neg V_1 \vee \neg V_2 \vee \dots \vee \neg V_m \vee A$ , эквивалентный  $V_1 \wedge V_2 \wedge \dots \wedge V_m \rightarrow A$ , записывается на Прологе в следующем виде:

$A:- V_1, V_2, \dots, V_m.$  (при  $m > 0$ )

$A.$  (при  $m = 0$ )

Данная запись может быть прочитана следующим образом:  $A$  - выполнимо, если выполнимы все  $V_1, V_2, \dots, V_m$ .

Так как целевой дизъюнкт содержит лишь отрицательные литеры, он представляется в программе клозом вида:

$:- K_1, K_2, \dots, K_n.$

Иными словами, общая цель (теорема) представляется в языке Пролог последовательностью подцелей, истинность (выполнимость) которых проверяется последовательно, начиная с первой (левой) позиции.

Необходимо обратить внимание на следующее:

при построении каждой очередной резольвенты поиск дизъюнкта - посылки с положительной литерой осуществляется путем последовательного просмотра клозов базы сверху вниз;

клозы, уже использованные для построения резольвент в том или ином процессе логического вывода могут участвовать в построении новых резольвент при дальнейшем доказательстве этой же цели;

неполучение не означает, что исходная цель ложна - это лишь свидетельствует о том, что на основании содержащихся в базе сведений невозможно сделать вывод о ее истинности.

#### **1.4. Продукционная МПЗ**

МПЗ, основанные на правилах, являются наиболее распространенными и более 80% ЭС используют именно их. Продукционная модель основана на правилах, позволяющая представить знания в виде предложений типа "Если (условие), то (действие)".

Под "условием" (антецедентом) понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под "действием" (консеквентом) - действия, выполняемые при успешном исходе поиска (они могут быть промежуточными, выступающими далее как условия, и терминальными или целевыми, завершающими работу системы).

Чаще всего вывод на такой БЗ бывает прямой (от данных к поиску цели) или обратный (от цели для ее подтверждения - к данным). Данные - это исходные факты, хранящиеся в базе фактов, на основании которых запускается машина вывода или интерпретатор правил, перебирающий правила из продукционной базы знаний.

Продукционная МПЗ часто применяется в промышленных экспертных системах, поскольку привлекает разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического выво-

да. Интеллектуальная система, основанная на продукционной МПЗ в общем случае содержит 3 основных компонента: механизм вывода; базу знаний (набор правил); рабочую память.

В зависимости от направления, в котором проходит вывод (от фактов к цели или, наоборот, от цели к фактам) различают 3 типа выводов: прямой; обратный; двунаправленный.

*Прямой вывод* проходит в направлении от фактов к цели. Он имеет следующую последовательность:

- 1) извлечение из рабочей памяти фактов, предварительно записанных в нее;
- 2) применение к ним правил (продукций) и получение новых фактов;
- 3) продолжение вывода в соответствии с пп.1,2 до тех пор, пока не будет получен факт, который является целью или ее отрицанием, либо невозможно больше применять продукции.

*Обратный вывод* проходит в направлении от цели к фактам. Он имеет следующую последовательность действий:

- 1) предполагается, что цель - истинная, и из набора правил выбираются те продукции, в которых цель является заключением (т.е. входит в THEN-секцию);
- 2) предпосылки (т.е. IF-секции) этих правил принимаются за новые подцели и помещаются в рабочую память;
- 3) к новым подцелям вновь применяются пп.1,2 до тех пор, пока все правила не окажутся выбранными, а подцели будут ложными (тогда основная цель - ложная), либо пока все подцели, определяющие основную цель, не окажутся истинными, или достигнутыми, тогда основная цель - истинная.

*Двунаправленный вывод* является комбинированным способом, сочетающим элементы прямого и обратного вывода.

Продукционная МПЗ тесно связана с вопросами естественного языкового общения и задачами распознавания текста и машинного перевода (как с естественных, так и с формальных языков). Такая задача носит название задачи компиляции и ее целесообразно рассмотреть отдельно.

### **Основные понятия процессов компиляции**

**Грамматикой**  $G$  называется совокупность  $G = (N, T, P, S)$ , где  $N$  - множество нетерминальных символов алфавита,

$T$  - множество терминальных символов алфавита,  
 $P$  - множество правил продукции,  
 $S$  - начальный символ грамматики.

При этом  $N \cap T = \emptyset$ ,  $N \cup T =$  алфавит,  $S \in N$  - начальный символ грамматики принадлежит множеству нетерминальных символов, и  $P = \{U \rightarrow x\}$  ( $U \in N$ ,  $x \in N \cup T$ ).

**Продукция** - это упорядоченная пара  $(U, x)$ , записываемая как  $U \rightarrow x$ , где  $U$  - символ, а  $x$  - цепочка (конечная).

Пусть  $G$  - грамматика. Цепочка  $u$  **непосредственно порождает** цепочку  $w$  (или  $w$  **непосредственно выводима** из  $u$ ), т.е.  $u \Rightarrow w$ , если для некоторых цепочек  $x$  и  $y$  можно записать  $u = xUy$ ,  $w = xuy$ , и  $U \rightarrow u$  - правило грамматики.

Говорят, что  $v$  **порождает**  $w$  (запись  $v \Rightarrow^+ w$ ), если существует последовательность выводов  $v \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow u_n \Rightarrow w$ . Запись  $v \Rightarrow^* w$  означает, что либо  $v \Rightarrow w$ , либо  $v = w$ .

Цепочка  $x$  является **сентенциальной формой** в грамматике  $G$ , если существует вывод  $S \Rightarrow^* x$ , т.е. сентенциальная форма - это любой промежуточный или конечный результат вывода из  $S$  - начального символа грамматики  $G$ .

Если  $x$  - сентенциальная форма в грамматике  $G$  и  $x$  состоит только из терминальных символов, то цепочка  $x$  - это **предложение** в грамматике  $G$ .

**Языком** в грамматике  $G$  называется множество возможных предложений в  $G$ :  $L(G) = \{x \mid S \Rightarrow^* x, x \in T^+\}$ .

Пусть  $G$  - грамматика, а  $w = xuy$  - сентенциальная форма. Тогда  $u$  называется **фразой** сентенциальной формы  $w$  для нетерминального символа  $U$ , если  $S \Rightarrow^* xUy$  и  $U \Rightarrow^+ u$ ; и простой фразой, если  $S \Rightarrow^* xUy$  и  $U = u$ .

**Основой** всякой сентенциальной формы называется самая левая простая фраза.

Каждая грамматика описывает (задает) некоторый язык, и каждый язык описывается (задается, порождается) некоторой грамматикой.

### **Синтаксические деревья. Задачи разбора и вывода**

Задачи анализа текстов на тех или иных языках достаточно наглядно демонстрируются с помощью синтаксических деревьев.

Правила построения синтаксического дерева для произвольного предложения в некоторой грамматике следующие:

- 1) В качестве корневого узла строится узел  $S$  (где  $S$  - начальный символ грамматики);
- 2) В имеющемся дереве выбирается узел  $U$ , для которого в  $P$  имеется продукция  $U \rightarrow \alpha_1 \dots \alpha_k$  (если такого узла нет, то построение завершено);
- 3) К дереву добавляются узлы  $\alpha_1, \dots, \alpha_k$  и связи  $(U, \alpha_1), \dots, (U, \alpha_k)$ , и процесс повторяется, начиная с п.2.

Для правильно организованной грамматики процесс построения должен завершиться или иметь возможность завершиться в таком состоянии, когда все концевые узлы связаны с терминальными символами. При этом просмотр концевых узлов слева дает предложение грамматики, а концевые узлы в любой момент построения дерева образуют сентенциальную форму.

Процесс порождения предложений грамматики, основанный на описанном выше алгоритме построения дерева называется **ВЫВОДОМ**.

Как правило, интересует получение не любого, а некоторого, вполне определенного, предложения. Дело в том, что исходные программы, поступающие на вход транслирующей системы, являются совокупностями символьных цепочек. Для успешного осуществления процесса компиляции необходимо:

ответить на вопрос "принадлежит ли текст программы входному языку?", т.е. является ли исходная программа совокупностью предложений входного языка;

определить синтаксическую структуру программы.

Очевидно, что решение этих задач требует построения синтаксического дерева. Сам факт построения такого дерева будет свидетельствовать о том, что исходная программа является синтаксически правильной (с позиций грамматики входного языка).

Практически используют две стратегии вывода:

левый вывод – на каждом шаге вывода производится разрывание самого левого концевого узла (допускающего это);

правый вывод - осуществляется путем замены самого правого концевого узла, не соответствующего терминальному символу, правой частью одного из правил.

Если на каждом шаге производится свертка строки к одному нетерминальному символу - этот процесс называется **редукцией**.

В зависимости от порядка развертки (свертки) основ в сентенциальных формах различают **левосторонние** и **правосторонние выводы** или **редукции**.

### **Понятие автоматной грамматики.**

В общем случае правила продукции  $P$  КС-граммматики  $G=(N,T,P,S)$  имеют вид  $U \rightarrow \alpha$ , где  $U \in N$ ,  $\alpha \in A^*=(N \cup T)^*$ . Многообразие возможных конфигураций цепочки  $\alpha$  определяет "мощность" каждого "определения", а следовательно и "порождающую" силу грамматики. В ряде частных случаев удается наложить ряд ограничений на структуру правых частей продукций и за счет этого получить некоторые качественные свойства выводов (редукций), которые значительно упрощают задачу анализа входных текстов.

Одним из частных классов КС-грамматик являются автоматные или регулярные грамматики (А-грамматики). Аппарат автоматных грамматик наиболее широко используется при проектировании лексических анализаторов (сканеров). Объектом анализа блока лексического анализа являются лексемы, в качестве которых для языка программирования обычно выступают идентификаторы переменных, числовые константы, ключевые слова.

Все лексемы строятся из литер (символов). Множество используемых в языке литер может быть разбито на ряд классов – например, классы букв, цифр, разделителей. После прочтения очередного символа анализируемой программы сканер осуществляет определение класса, к которому она принадлежит. Дальнейший анализ предполагает определение допустимости вхождения литеры данного класса в обрабатываемую (анализируемую) лексему. Такой анализ существенно упрощается, если синтаксис лексем определяется регулярными выражениями, т.е. описывается автоматной грамматикой.

Конечным автоматом (КА) является пятеркой вида:

$$КА = (K, T, M, S, Z)$$

где  $K$  - алфавит состояний КА,

$T$  - входной алфавит,



$M$  - отображение декартова произведения множеств  $K$  и  $T$  (множества  $K \times T$ ) на множество  $K$ , представляемое графом переходов  $KA$  или таблицей состояний,

$S$  - начальное состояние  $KA$  ( $S \in K$ ),

$Z$  - непустое множество конечных состояний ( $Z \subset K$ ).

Если  $KS$ -грамматика является  $A$ -грамматикой, то алфавит состояний  $K \equiv N$ , входной алфавит  $T \equiv T$ . Конечный автомат, реализующего разбор предложений этой грамматики путем выполнения процесса редукции, можно построить следующим образом.

Если в множестве правил продукции есть хотя бы одно правило вида  $U \rightarrow a$  (где  $a$  – терминальный символ), то ввести дополнительный нетерминальный символ (например,  $Q$ ), все правила указанного вида преобразовать в  $U \rightarrow Qa$ .

Изобразить множество вершин графа состояний, обозначив каждую вершину одним нетерминальным символом. Вершина  $Q$  соответствует начальному состоянию автомата, а вершина  $S$  – заключительному ( $S$  – начальный символ грамматики).

Для каждого правила продукции  $U \rightarrow Va$  провести дугу от вершины  $V$  к вершине  $U$  и пометить ее символом  $a$ .

В зависимости от правил поведения конечные автоматы ( $KA$ ) могут быть как детерминированными, так и недетерминированными.

## 1.5. Фреймовая МПЗ

Фреймовая МПЗ базируется на понятии функционального программирования - способа составления программ, в которых единственным действием является вызов функции, единственным способом расчленения программ на части является введение имени для функции и задание для этого имени выражения, вычисляющего значение функции, а единственным правилом композиции - оператор суперпозиции других функций. Оно следует из такого раздела математики, как лямбда-исчисление, созданное американским ученым А. Чёрчем. Продемонстрировать мощь функционального программирования можно на следующем примере. Чёрч длительное время пытался сформулировать в лям-

бда-исчислении операцию вычитания единицы из натурального числа:  $0-1=0$ ,  $(n+1)-1=n$ . Оригинальное решение этой задачи нашел в 1932 году Стефан Клани. В его трактовке задача имеет следующее функциональное решение:

$$f(x,y,z) \equiv \begin{cases} \text{if } x=0 & \text{then } 0 \\ \text{else} & \\ & \text{if } y+1=x & \text{then } z \\ & \text{else } f(x,y+1,z+1); \end{cases}$$
$$n-1 = f(n,0,0).$$

Динамика вычислений для примера  $n=3$ :  $3-1 = f(3,0,0) \rightarrow f(3,1,1) \rightarrow f(3,2,2) = 2$ .

Объектом, подлежащим обработке в Лисп-системах, является так называемое *S-выражение* (от Symbolic - "символьный").

Последовательность S-выражений, заключенная в скобки, также является S-выражением (списком).

Очень важно понятие пустого списка - списка, не содержащего элементов. Такой список обозначается () или NIL. С позиций языка Лисп пустой список является атомом.

Поскольку любая синтаксически правильная Лисп-программа представляет собой S-выражение (или последовательность S-выражений), ее выполнение сводится к вычислению данного выражения (или последовательному вычислению каждого из S-выражений).

Вычисление (интерпретацию) S-выражений осуществляет Лисп-система. Механизм работы любой Лисп-системы весьма прост. Он состоит из трех последовательных шагов: считывание S-выражения; интерпретация S-выражения (EVAL); печать S-выражения.

С точки зрения синтаксиса программа и данные не различаются, поэтому любое S-выражение можно пытаться интерпретировать как программу. Такую попытку осуществляет функция EVAL, которая возвращает полученное при этом значение. Процесс интерпретации сводится к следующему:

если S-выражение является числовым атомом, или атомом T, или NIL, то EVAL возвращает это S-выражение без изменений;

если S-выражение является символьным атомом, то функция EVAL возвращает последнее значение, которое было присвоено этому атому, в противном случае большинство интерпретаторов дает сообщение об ошибке (система XLisp в качестве значения неинициализированного символьного атома возвращает его изображение);

если S-выражение является строкой, то его значением является сама эта строка;

если S-выражение представляет собой список вида  $(f \text{ arg}_1 \dots \text{ arg}_n)$ , то функция EVAL пытается интерпретировать его как выполнение функции с именем  $f$  и аргументами  $\text{arg}_1, \dots, \text{arg}_n$ .

В последнем случае в Лисп-системе должна быть определена функция с именем  $f$ , и EVAL возвращает результат вычисления этой функции на указанном списке параметров. Если же функция с именем  $f$  не определена, то значением S-выражения является список значений элементов данного S-выражения (подобное происходит и в случае, когда первым элементом в списке указано S-выражение, которое не может быть интерпретировано как имя функции).

На базе функций, изначально определенных в Лисп-системе, пользователь может создавать свои функции. В основе определения функций лежит использование  **$\lambda$ -выражений**.

$\lambda$ -выражением называется S-выражение вида:

$(\text{LAMBDA} (\text{arg}_1 \text{ arg}_2 \dots \text{ arg}_N) \text{ expr}_1 \text{ expr}_2 \dots \text{ expr}_K)$ , где

LAMBDA - специальный атом, зарезервированный в языке для определения функций;

$\text{arg}_1 \text{ arg}_2 \dots \text{ arg}_N$  - атомы, используемые для указания формальных аргументов функции, определяемой  $\lambda$ -выражением;

$\text{expr}_1 \text{ expr}_2 \dots \text{ expr}_K$  - любые S-выражения, составляющие "тело" функции.

$\lambda$ -конверсией называется процесс связывания формальных переменных (формальных параметров) функции с фактическими значениями и интерпретация "тела" функции. В качестве результата вычисления  $\lambda$ -выражения, т.е. "тела" функции, возвращается значение последнего S-выражения, составляющего "тело" функции. После выполнения функции ее формальные аргументы освобождаются от связей.

Фрейм — это абстрактный образ для представления стереотипа объекта, понятия или ситуации. Под абстрактным образом понимается некоторая обобщенная и упрощенная модель или структура. Фреймом также называется и формализованная модель для отображения образа.

Различают фреймы-образцы или прототипы, хранящиеся в базе знаний, и фреймы-экземпляры, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных. Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире через:

фреймы-структуры, использующиеся для обозначения объектов и понятий (заем, залог, вексель);

фреймы-роли (менеджер, кассир, клиент);

фреймы-сценарии (банкротство, собрание акционеров, празднование именин);

фреймы-ситуации (тревога, авария, рабочий режим устройства) и др.

Традиционно структура фрейма может быть представлена как список свойств:

(ИМЯ ФРЕЙМА:

(имя 1-го слота: значение 1-го слота),

(имя 2-го слота: значение 2-го слота),

.....

(имя N-го слота: значение N-го слота)).

Существует несколько способов получения слотом значений во фрейме-экземпляре:

по умолчанию от фрейма-образца (Default-значение);

через наследование свойств от фрейма, указанного в слоте;

по формуле, указанной в слоте;

через присоединенную процедуру;

явно из диалога с пользователем;

из базы данных.

Важнейшим свойством теории фреймов является заимствование из теории семантических сетей - так называемое наследование свойств. И во фреймах, и в семантических сетях наследо-

вание происходит по это-связям. Слот связи указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, т. е. переносятся, значения аналогичных слотов.

Основным преимуществом фреймов как модели представления знаний является то, что она отражает концептуальную основу организации памяти человека, а также ее гибкость и наглядность. Специальные языки представления знаний в сетях фреймов FRL (Frame Representation Language).

## 1.6. Семантические сети

Термин "семантическая" означает "смысловая", а сама семантика — это наука, устанавливающая отношения между символами и объектами, которые они обозначают, т. е. наука, определяющая смысл знаков.

**Семантическая сеть** наиболее близка к тому, как представляются знания в текстах на естественном языке. В ее основе лежит идея о том, что вся необходимая информация может быть описана как совокупность троек (  $a r b$  ), где  $a$  и  $b$  - два объекта или понятия, а  $r$  двоичное отношение между ними.

Графически семантическая сеть представляется в виде помеченного ориентированного графа, в котором вершинам соответствуют объекты (понятия), а дугам - их отношения. Дуги помечаются именами соответствующих отношений.

Семантическая сеть является моделью широкого предназначения. Выделяются различные виды семантических сетей:

ситуационные сети, которые описывают временные, пространственные и причинно-следственные (клаузальные) отношения;

целевые сети, используемые в системах планирования и синтеза, которые описывают отношения "цель-средства" и "цель-подцель";

классификационные сети, использующие отношения "род-вид", "класс-подкласс";

функциональные сети, использующие отношения "аргумент-функция" и т.д.

Особенность семантической сети как модели представления знаний, которая может одновременно считаться и ее достоинством

вом, и ее недостатком, заключается в невозможности в явном виде разделить БЗ и механизм логического вывода. Поэтому интерпретация семантической сети осуществляется только с помощью использующих ее процедур.

Основным способом интерпретации семантической сети является способ **сопоставления** частей сетевой структуры. Он основан на построении подсети (подграфа), соответствующей задаваемому вопросу, и сопоставлении ее с общей сетью, имеющейся в БЗ. Запросная подсеть накладывается на имеющийся в базе знаний фрагмент.

Для поиска отношений между концептуальными объектами используется другой способ **перекрестного поиска**. Согласно этому способу ответ на вопрос выводится путем обнаружения в имеющейся сети узла, в котором пересекаются дуги, исходящие из различных узлов запросной подсети.

### **Контрольные вопросы:**

1. При соблюдении каких условий можно говорить о переходе от данных к знаниям.
2. Что такое СУБЗ и каковы её функции.
3. Перечислите основные МПЗ.
4. Что такое Хорновские дизъюнкты.
5. Перечислите последовательность выполнения действий при обратном выводе.
6. Перечислите этапы стратегии построения автоматов.
7. Назовите правила построения синтаксического дерева.
8. Из каких шагов состоит механизм работы Лисп-системы.
9. Назовите самое общее циклическое предложение в Лиспе. Что можно создать с его помощью.
10. Каким образом можно графически представить семантическую сеть.

## Тема 2. ЭКСПЕРТНЫЕ СИСТЕМЫ

Данная глава посвящена вопросам построения ЭС. Даны понятия концепции и назначение ЭС. Предложена структура ЭС.

Рассмотрены технология разработки ЭС: состав и взаимодействие участников построения и эксплуатации, принципы разработки, этапы проектирования экспертной системы.

Описана организация процесса приобретения и формализации знаний.

### 2.1. Назначения и основные свойства ЭС

Существенный прорыв в практических приложениях ИИ произошел в середине 70-х годов, когда на смену поискам универсального алгоритма мышления пришла идея моделировать конкретные знания специалистов-экспертов. Появились первые коммерческие системы, основанные на знаниях, получившие название «экспертные системы» (ЭС). Сформировался новый подход к решению интеллектуальных задач – представление и использование знаний. ЭС используются для решения так называемых неформализованных задач, общим для которых является:

алгоритмическое решение задачи неизвестно или не может быть использовано из-за ограниченности ресурсов ЭВМ;

задача не может быть определена в числовой форме (требуется символьное представление);

цели решения задачи не могут быть выражены в терминах точно определенной целевой функции;

большая размерность пространства решения;

динамически изменяющиеся данные и знания;

неполнота, неоднозначность и противоречивость исходных данных и знаний о решаемой задаче.

ЭС – это программное средство, использующее экспертные знания для обеспечения высокоэффективного решения неформализованных задач в узкой проблемной области.

Основу ЭС составляет БЗ о ПрО, которая накапливается в процессе построения и эксплуатации ЭС. Накопление и органи-

зация знаний – важнейшее свойство всех ЭС. Использование знаний определяет такие свойства ЭС, как (рис. 2.1):



Рис. 2.1. Основные свойства экспертных систем

применение для решения выявленных проблем высококачественного опыта, который представляет собой мышления наиболее квалифицированных экспертов в данной области, что в конечном счете приводит к точным и эффективным результатам;

наличие прогностических возможностей, при которых ЭС выдает ответы не только для конкретной ситуации, но и показывает, как изменяются эти ответы в новых ситуациях и существует возможность получения подробного объяснения каким образом новая ситуация привела к изменениям;

институциональная память – набор знаний, за счет входящей в состав ЭС базы знаний, становится постоянно обновляемым справочником наилучших стратегий и методов, используемых персоналом, таким образом ведущие специалисты уходят, но их опыт остается;

возможность использования ЭС для обучения и тренировки специалистов, по которым можно изучать рекомендуемую политику и методы.

Состав и взаимодействие участников построения и эксплуатации экспертных систем представлены на рис .2.2.

ЭС – это программное средство, использующее знания экспертов, для высокоэффективного решения задач в интересующей пользователя проблемной области. Она является системой, а не просто программой, так как содержит базу знаний, решатель проблем и компоненты поддержки, которые помогают пользователю взаимодействовать с основной программой.



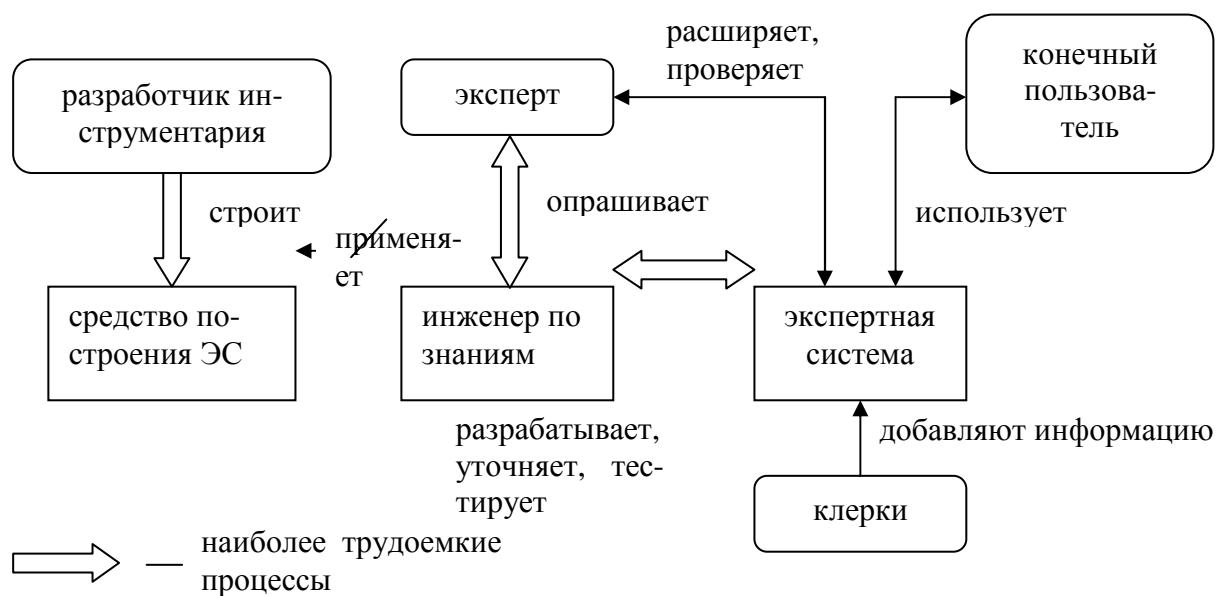


Рис. 2.2. Взаимосвязи основных участников построения и эксплуатации экспертных систем

Эксперт – это человек, способный ясно выразить свои мысли и пользующийся репутацией специалиста, умеющего находить правильные решения проблем в конкретной предметной области. В конечном счете, его подготовка определяет уровень компетенции базы знаний ЭС. Эксперт использует свои приемы и методы, чтобы сделать поиск решения более эффективным, и ЭС моделирует все его стратегии.

Инженер по знаниям – человек, как правило, имеющий познания в информатике и искусственном интеллекте и знающий, как надо строить ЭС. Он опрашивает экспертов, аккумулирует знания, решает, каким образом они должны быть представлены в ЭС, и может помочь программисту в написании программ.

Средство построения ЭС – используется инженером по знаниям или программистом для построения ЭС. Этот инструмент отличается от обычных языков программирования тем, что обеспечивает удобные способы представления сложных высокоуровневых понятий.

Разработчик инструментария – программист, к которому обязательно должен быть знаком с основными структурами представления знаний и механизмами вывода, состоянием отечественного и мирового рынка программных продуктов для разработки ЭС и диалоговых интерфейсов.

Пользователь – человек, который использует уже построенную ЭС. К нему предъявляются самые слабые требования, поскольку его не выбирают. Он является в некотором роде заказчиком системы. Термин пользователь несколько неоднозначен. Обычно он обозначает конечного пользователя. Однако из рис. 2.2 следует, что пользователем может выступать:

разработчик инструментария, отлаживающий средство построения ЭС;

инженер по знаниям, уточняющий существующие в ЭС знания;

эксперт, добавляющий в систему новые знания;

клерк, заносающий в систему текущую информацию.

## 2.2. Особенности построения и организации ЭС

Основой любой ЭС является совокупность знаний, структурированная в целях упрощения процесса принятия решения. Для специалистов в области ИИ термин *знания* означает совокупность сведений о сущностях (объектах, предметах) реального мира, их свойствах и отношениях между ними в определенной *предметной области*. В ЭС они принимают форму *фактов* и *правил*. Факты и правила в ЭС не всегда либо истинны, либо ложные. Иногда существует некоторая степень неуверенности в достоверности факта или точности правила. Если это сомнение выражено явно, то оно называется *коэффициентом доверия*.

Коэффициент доверия – это число, которое означает вероятность или степень уверенности, с которой можно считать данный факт или правило достоверным или справедливым.

Многие правила ЭС являются *эвристиками*, т.е. эмпирическими правилами или упрощениями, которые эффективно ограничивают поиск решения. ЭС используют эвристики, т.к. задачи, которые она решает, трудны, не до конца понятны, не поддаются строгому математическому анализу или алгоритмическому решению. Алгоритмический метод гарантирует корректное или оптимальное решение задачи, тогда как эвристический метод дает приемлемое решение в большинстве случаев.

Знания в ЭС организованы так, чтобы знания о предметной области отделить от других типов знаний системы, таких как об-

щие знания о том, как решать задачи или знания, о том, как взаимодействовать с пользователем. Выделенные знания о проблемной области называют БЗ, в свою очередь, знания о нахождении решений задач называют *механизмом вывода*.

БЗ содержит факты (данные) и правила, использующие эти факты как основу для принятия решений. Механизм вывода содержит:

интерпретатор, определяющий как применять правила для вывода новых знаний на основе информации, хранящейся в БЗ;

диспетчер, устанавливающий порядок применения этих правил.

Обмен же информацией между пользователем и ЭС выполняют программные инструменты *интеллектуального интерфейса*, которые воспринимают сообщения пользователя и преобразуют их в форму представления БЗ и, наоборот, переводят внутреннее представление результата обработки в формат пользователя и выдает сообщение на требуемый носитель. Важнейшим требованием к организации диалога пользователя с ЭС является естественность, которая не означает буквально формулирование потребностей пользователя предложениями естественного языка, хотя это и не исключается в ряде случаев. Важно, чтобы последовательность решения задачи была гибкой, соответствовала представлениям пользователя и велась в профессиональных терминах.

В целом, такие ЭС получили название *статических ЭС* и имеют структуру, аналогичную приведенной на рис. 2.3. Статические ЭС используются в тех приложениях, где можно не учитывать изменения окружающего мира за время решения задачи.

Однако существует более высокий класс приложений, где требуется учитывать динамику изменения окружающего мира за время исполнения приложения. Такие экспертные системы получили название *динамических ЭС* (см. рис. 2.4).

По сравнению со статической ЭС в динамическую вводятся еще два компонента:

подсистема моделирования внешнего мира;

подсистема сопряжения с внешним миром.

Динамические ЭС осуществляют связи с внешним миром через систему контроллеров и датчиков. Кроме того компоненты

БЗ и механизма вывода существенно изменяются, чтобы отразить временную логику происходящих в реальном мире событий.

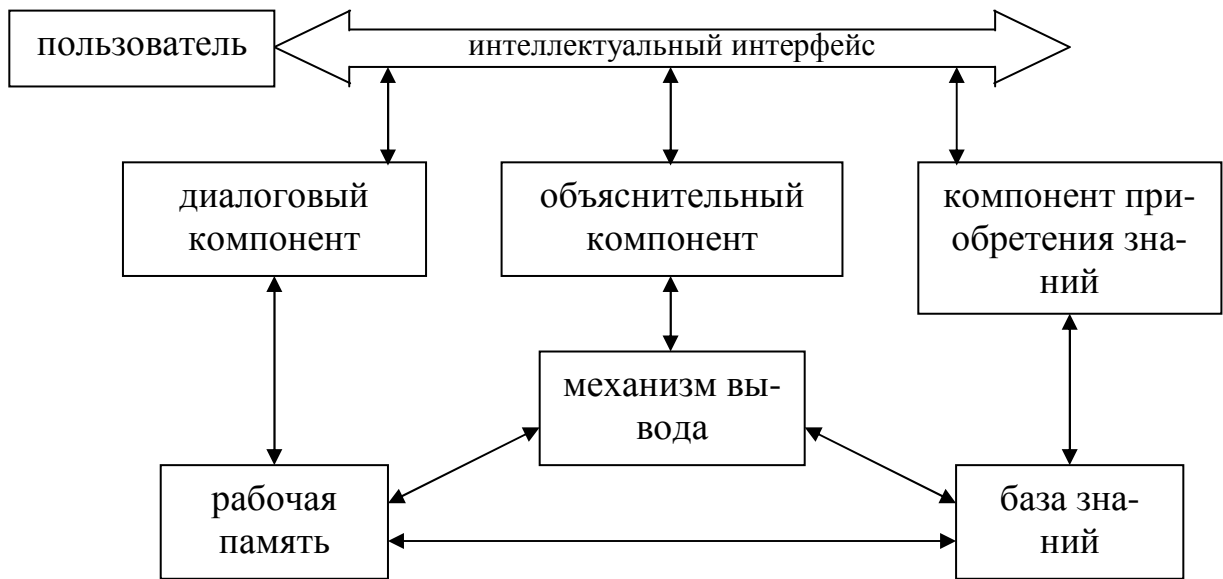


Рис. 2.3. Структура статической ЭС

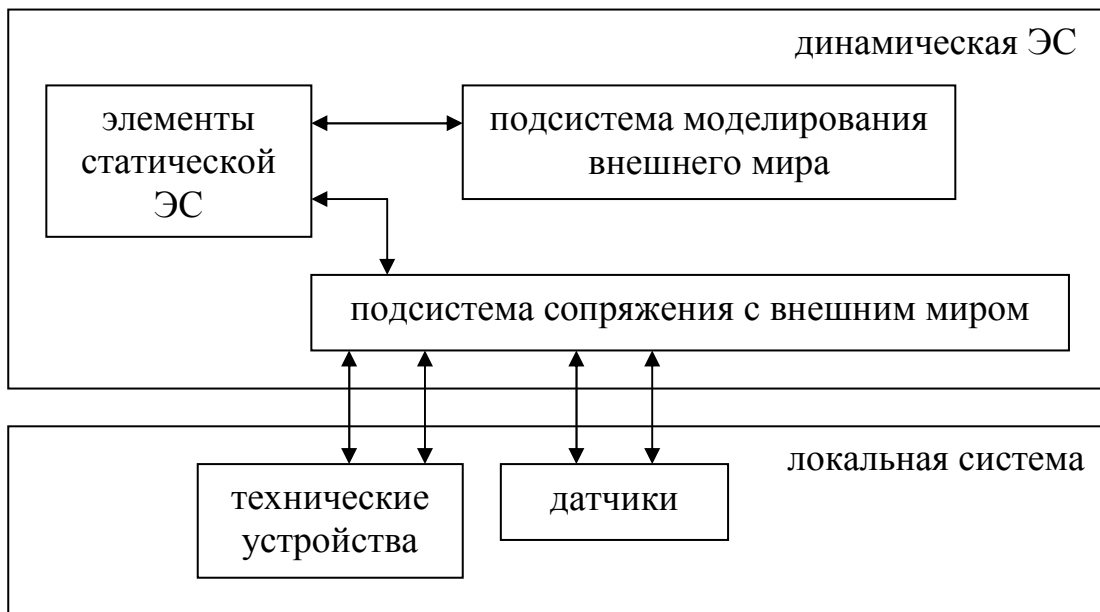


Рис. 2.4. Обобщенная структура динамической ЭС

### 2.3. Преимущества использования экспертных систем

Преимуществами и положительными качествами искусственной компетенции являются:

**постоянство.** Человеческая компетенция ослабевает со временем. Перерыв в деятельности человека-эксперта может серьезно отразиться на его профессиональных качествах;

**легкость передачи.** Передача знаний от одного человека другому – долгий и дорогой процесс. Передача искусственной информации – это простой процесс копирования программы или файла данных;

**устойчивость и воспроизводимость результатов.** Эксперт-человек может принимать в тождественных ситуациях разные решения из-за эмоциональных факторов. Результаты ЭС – стабильны;

**стоимость.** Эксперты, особенно высококвалифицированные обходятся очень дорого. ЭС, наоборот, сравнительно не дороги. Их разработка дорога, но они дешевы в эксплуатации.

Вместе с тем разработка ЭС не позволяет полностью отказаться от эксперта-человека. Однако ЭС может позволить отказаться от услуг высококвалифицированного эксперта, оставив эксперта средней квалификации, используя при этом ЭС для усиления и расширения его профессиональных возможностей.

## **2.4. Основные режимы работы экспертных систем**

В работе ЭС можно выделить два основных режима: режим приобретения знаний и режим решения задачи (режим консультации или режим использования). В режиме *приобретения* знаний общение с ЭС осуществляет эксперт (при помощи инженера по знаниям).

Используя компонент приобретения знаний, эксперт описывает проблемную область в виде совокупности фактов и правил. Другими словами, "наполняет" ЭС знаниями, которые позволяют ей самостоятельно решать задачи из проблемной области.

Отметим, что этому режиму при традиционном подходе к программированию соответствуют этапы: алгоритмизации, программирования и отладки, выполняемые программистом. Таким образом, в отличие от традиционного подхода в случае ЭС разработку программ осуществляет не программист, а эксперт, не владеющий программированием.

В режиме *консультаций* общение с ЭС осуществляет конечный пользователь, которого интересует результат и (или) способ его получения. Необходимо отметить, что в зависимости от назначения ЭС пользователь может:

не быть специалистом в данной предметной области, и в этом случае он обращается к ЭС за результатом, который не умеет получить сам;

быть специалистом, и в этом случае он обращается к ЭС с целью ускорения получения результата, возлагая на ЭС рутинную работу.

Следует отметить, что в отличие от традиционных программ ЭС при решении задачи не только исполняют предписанную алгоритмом последовательность операций, но и сама предварительно формирует её.

Хорошо построенная ЭС имеет возможность самообучаться на решаемых задачах, пополняя автоматически свою БЗ результатами полученных выводов и решений.

## **2.5. Отличия ЭС от традиционных программ**

Особенности ЭС, отличающие их от обычных программ, заключаются в том, что они должны обладать *Компетентностью*, а именно: достигать экспертного уровня решений (т.е. в конкретной предметной области иметь тот же уровень профессионализма, что и эксперты-люди). Быть умелой (т.е. применять знания эффективно и быстро, избегая, как и люди, ненужных вычислений). Иметь адекватную робастность (т.е. способность лишь постепенно снижать качество работы по мере приближения к границам диапазона компетентности или допустимой надёжности данных).

*Возможностью к символьным рассуждениям*, а именно: представлять знания в символьном виде, переформулировать символьные знания.

*Глубиной*, а именно: работать в предметной области, содержащей трудные задачи, использовать сложные правила (т.е. использовать либо сложные конструкции правил, либо большое их количество).

*Самосознанием*, а именно: исследовать свои рассуждения (т.е. проверять их правильность).

*Объяснять* свои действия.

ЭС разрабатываются, чтобы вести себя как эксперты. Они, как правило, дают правильные ответы, но иногда, как и люди, способны ошибаться. ЭС должны иметь потенциальную возможность учиться на своих ошибках.

## 2.6. Классификация ЭС

Класс ЭС сегодня объединяет несколько тысяч различных программных комплексов, которые можно классифицировать по различным критериям. Полезными могут оказаться следующие классификации (см. рис. 2.5).

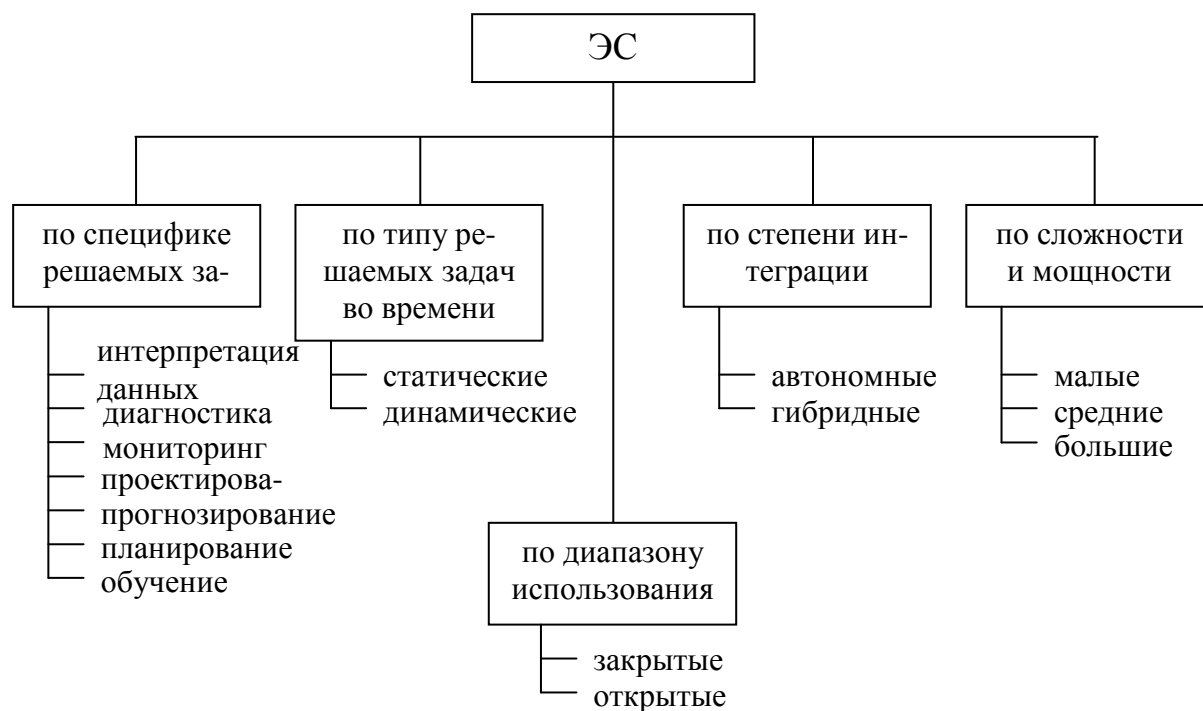


Рис. 2.5. Схема классификации экспертных систем

### **Классификация по специфике решаемых задач.**

**Интерпретация данных.** Это одна из традиционных задач для ЭС. Под интерпретацией понимается определение смысла данных, результаты которого должны быть согласованными и корректными. Обычно предусматривается многовариантный анализ данных: обнаружение и идентификация различных типов оке-

анских судов; определение основных свойств личности по результатам психодиагностического тестирования.

**Диагностика.** Под диагностикой понимается обнаружение неисправности в некоторой системе. Неисправность – это отклонение от нормы. Такая трактовка позволяет с единых теоретических позиций рассматривать и неисправность оборудования в технических системах, и заболевания живых организмов, и всевозможные природные аномалии. Важной спецификой является необходимость понимания функциональной структуры диагностирующей системы: диагностика и терапия сужения коронарных сосудов; диагностика ошибок в аппаратуре и математическом обеспечении ЭВМ.

**Мониторинг.** Основная задача мониторинга – непрерывная интерпретация данных в реальном масштабе времени и сигнализация о выходе тех или иных параметров за допустимые пределы. Главные проблемы – пропуск тревожной ситуации и инверсная задача ложного срабатывания. Сложность этих проблем в размытости симптомов тревожных ситуаций и необходимость учета временного контекста: контроль за работой электростанций, помощь диспетчерам атомного реактора; контроль аварийных датчиков на химическом заводе.

**Проектирование.** Проектирование состоит в подготовке спецификаций на создание объектов с заранее определенными свойствами. Под спецификацией понимается весь набор необходимых документов – чертеж, пояснительная записка и т.д. Основные проблемы здесь – получение четкого структурного описания знаний об объекте и проблема «следа». Для организации эффективного проектирования и, в еще большей степени, перепроектирования необходимо формировать не только сами проектные решения, но и мотивы их принятия. Таким образом, в задачах проектирования тесно связываются два основных процесса, выполняемых в рамках соответствующей ЭС: *процесс вывода решения* и *процесс объяснения*: проектирование БИС; синтез электрических цепей.

**Прогнозирование.** Прогнозирующие системы логически выводят вероятные следствия из заданных ситуаций. В прогнозирующей системе обычно используется параметрическая динамическая модель, в которой значения параметров «подгоняются»



под заданную ситуацию. Выводимые из этой модели следствия составляют основу для прогнозов с вероятностными оценками: предсказание погоды; прогнозы в экономике.

**Планирование.** Под планированием понимается нахождение планов действий, относящихся к объектам, способным выполнять некоторые функции. В таких ЭС используются модели поведения реальных объектов с тем, чтобы логически вывести последствия планируемой деятельности: планирование промышленных заказов; планирование эксперимента.

**Обучение.** Системы обучения диагностируют ошибки при изучении какой-либо дисциплины с помощью ЭВМ и подсказывают правильные решения. Они аккумулируют знания о гипотетическом «ученике» и его характерных ошибках, затем в работе способны диагностировать слабости в знаниях обучаемых и находить соответствующие средства для их ликвидации. Кроме того, они планируют акт общения с учеником в зависимости от успехов ученика с целью передачи знаний.

### **Классификация по типу решаемых задач во времени**

**Статические ЭС** разрабатываются для ПРО, характеризующихся неизменностью входных данных в процессе поиска решения: диагностика неисправностей в автомобиле.

**Динамические ЭС** разрабатываются для проблемных областей, требующих обеспечить возможность изменения входных данных в процессе поиска решения: система обслуживания биржевых операций.

### **Классификация по степени интеграции**

**Автономные ЭС** работают непосредственно в режиме консультаций с пользователем для обеспечения высокоэффективного решения неформализованных задач в узкой проблемной области, не требующих дополнительного привлечения традиционных методов обработки данных (расчеты, моделирование и т.д.), т.е. изолированная ЭС, не рассчитанная на взаимодействие с другими программными системами.

**Гибридные ЭС** представляют собой программный комплекс, агрегирующий стандартные пакеты прикладных программ (например, математическую статистику, линейное программирова-

ние или системы управления базами данных) и средства манипулирования знаниями. Это может быть интегрированная среда для решения сложной задачи с элементами экспертных знаний.

### **Классификация по сложности и мощности**

Здесь мощность определяется множеством используемых правил.

*Малые ЭС* реализовываются на ПК (IBM PC, Macintosh и подобные) как автономные (изолированные) ЭС.

*Средние ЭС* работают в режиме клиент-сервер, используются в большинстве случаев как гибридные (интегрированные) ЭС.

*Большие ЭС* реализовываются на ЭВМ общего назначения и используют БЗ большого объема.

### **Классификация по диапазону использования**

*Закрытые ЭС* ориентированы на использование только в программной среде конкретной ИС.

*Открытые ЭС* ориентированы на использование в разнородном программно-аппаратном окружении и могут быть перенесены на другие платформы без существенных изменений.

## **2.7. Технология разработки экспертных систем**

Состав и взаимодействие участников построения и эксплуатации ЭС. Принципы разработки. Этапы проектирования экспертной системы: идентификация, концептуализация, формализация, реализация, тестирование, опытная эксплуатация. Содержание этапов проектирования. Участники процесса проектирования: эксперты, инженеры по знаниям, конечные пользователи.

**Стадия существования** характеризует степень проработанности и отлаженности ЭС. Выделяют пять стадий: демонстрационный прототип; исследовательский прототип; действующий прототип; промышленная система; коммерческая система.

*Демонстрационным прототипом* называется ЭС, которая решает часть требуемых задач, демонстрируя при этом приемлемость метода инженерии знаний. При наличии развитых инструментальных средств для разработки демонстрационного прототипа требуется в среднем примерно 1-2 месяца (при их отсутст-

вии - 12-18 месяцев). База знаний демонстрационного прототипа содержит обычно 50-100 правил, описывающих проблемную область.

*Исследовательским прототипом* называется ЭС, которая решает все требуемые задачи, но неустойчива в работе и не полностью проверена. На доведении системы до стадии исследовательского прототипа уходит с момента начала работ по созданию ЭС 3-6 месяцев. Исследовательский прототип имеет в базе знаний от 200 до 500 правил.

*Действующий прототип* надежно решает все задачи, но требует чрезмерно много времени и памяти для решения некоторых сложных задач. Для доведения системы до стадии действующего прототипа требуется 6-12 месяцев, а количество правил в базе знаний при этом увеличивается до 500-1000.

После дальнейшего усовершенствования действующего прототипа, связанного с повышением эффективности решения сложных задач (обычно это осуществляется путем переписывания программ с использованием более эффективных инструментальных средств) система достигает стадии *промышленной системы*. База знаний еще более увеличивается (до 1000-1500 правил). На доведение ЭС до стадии промышленной требуется 1-1,5 года.

*Коммерческой системой* называется ЭС, в которой обобщены задачи, решаемые на стадии промышленной системы, и интерфейс пользователя (а также все другие компоненты) развит таким образом, что система становится полностью пригодной не только для собственного использования, но и для продажи различным потребителям. Для доведения системы до стадии коммерческой требуется 1,5-3 года. При этом база знаний достигает объема в 1500-3000 правил и более.

Организация процесса приобретения и формализации знаний. Эксперт и инженер по знаниям: формы и порядок взаимодействия.

### **Принципы разработки**

Разработка ЭС и обычного программного продукта существенно различаются. Опыт разработки ранних ЭС показал, что использование при этом методологии, принятой в традиционном

программировании, либо чрезмерно затягивает процесс создания ЭС, либо вообще приводит к отрицательному результату. Это вызвано неформализованностью задач, решаемых ЭС, и отсутствием завершенной теории и методологии разработки таких систем.

Перед началом работ по разработке ЭС инженер по знаниям должен рассмотреть вопросы о целесообразности данной работы. В общем виде ответ может быть таким: разработка (а также использование) ЭС будет целесообразна, если она возможна, оправдана и методы инженерии знаний соответствуют решаемой задаче. Понятия "возможно", "оправдано", "соответствуют" определяются нижеперечисленными принципами.

Чтобы разработка ЭС была **возможной**, необходимо одновременное выполнение по крайней мере следующих принципов:

наличие экспертов в данной области, которые решают задачу значительно лучше, чем начинающие специалисты (**принцип наличия эксперта**);

сходимость экспертов в оценке предлагаемого решения (иначе нельзя будет оценить качество разработанной ЭС) (**принцип сходимости экспертов**);

вербализуемость методов экспертов, которые должны уметь выразить их на естественном языке и объяснить (в противном случае трудно рассчитывать на то, что знания экспертов будут "извлечены" и вложены в ЭС) (**принцип вербализуемости методов**);

задача, решаемая с помощью ЭС, должна быть не слишком трудной и решаться не более нескольких часов (**принцип ограниченности во времени**);

решение не должно в значительной степени использовать "здоровый смысл", т.е. широкий спектр общих сведений о мире и о способе его функционирования, которые использует любой нормальный человек, так как подобные знания пока не удастся в достаточном количестве вложить в СИИ (**принцип ограниченности здравого смысла**).

Использование ЭС в данном приложении может быть возможно, но не оправдано. Применение ЭС может быть **оправдано** одним из следующих факторов:

решение задачи принесет значительный эффект (**принцип значительного эффекта**);

использование человека-эксперта для решения задачи по какой-либо причине невозможно либо из-за недостаточного количества экспертов, либо из-за необходимости выполнять экспертизу одновременно в различных местах (**принцип отсутствия эксперта**);

при необходимости решать задачу в окружении, враждебном для человека (**принцип враждебной среды**).

Приложение **соответствует** методам ЭС, если решаемая задача будет иметь:

естественное решение посредством манипулирования с символами, т.е. с помощью символьных рассуждений, а не с числами, как принято в математических методах и в традиционных программах (**принцип символьных рассуждений**);

эвристическую, а не алгоритмическую природу, т.е. ее решение должно сводиться к применению эвристических правил; если задача гарантированно решается при соблюдении заданных ограничений с помощью некоторых формальных процедур, то она не подходит для применения ЭС (**принцип эвристического решения**);

достаточную сложность, чтобы оправдать затраты на разработку ЭС, однако не чрезмерную сложность (**принцип достаточной сложности решения**); если решение задачи занимает у эксперта несколько дней или даже недель, то задача перестает отвечать принципу ограниченности во времени (как указано выше) и, следовательно, разработка для нее ЭС считается невозможной;

практическую значимость и достаточную ограниченность, чтобы решаться методами инженерии знаний (**принцип практической значимости**).

При разработке ЭС используется концепция "быстрого прототипа". Она заключается в том, что разработчики не пытаются сразу создать готовый конечный продукт. На начальных этапах они создают прототип (прототипы) ЭС, который должен удовлетворять двум противоречивым требованиям. С одной стороны, он должен решать типовые задачи конкретного приложения. С другой стороны, время и трудоемкость разработки прототипа должны быть незначительными, чтобы можно было максимально запараллелить процесс накопления и отладки знаний, осуществляемый экспертом, с процессом выбора или разработки про-

граммных средств, осуществляемым инженером по знаниям и программистом. Для удовлетворения указанных требований при создании прототипа, как правило, используются разнообразные инструментальные средства.

Одной из основных задач прототипа является демонстрация пригодности методов инженерии знаний для данного приложения. В случае успеха эксперт с помощью инженера по знаниям расширяет знания прототипа о предметной области. При неудаче разрабатывается новый прототип или делается вывод о непригодности методов инженерии знаний для данного приложения.

По мере увеличения знаний прототип достигает такого состояния, когда он успешно решает все задачи данного приложения. В этом случае осуществляется преобразование прототипа ЭС в конечный продукт, которое обычно проводится путем репрограммирования ЭС на языках низкого уровня, обеспечивающих повышение быстродействия ЭС и уменьшение требуемой памяти.

Технология разработки ЭС включает в себя шесть этапов (см. рис. 2.6): этапы идентификации, концептуализации, формализации, выполнения, тестирования, опытной эксплуатации. Рассмотрим более подробно последовательности действий, которые необходимо выполнить на каждом из этапов.

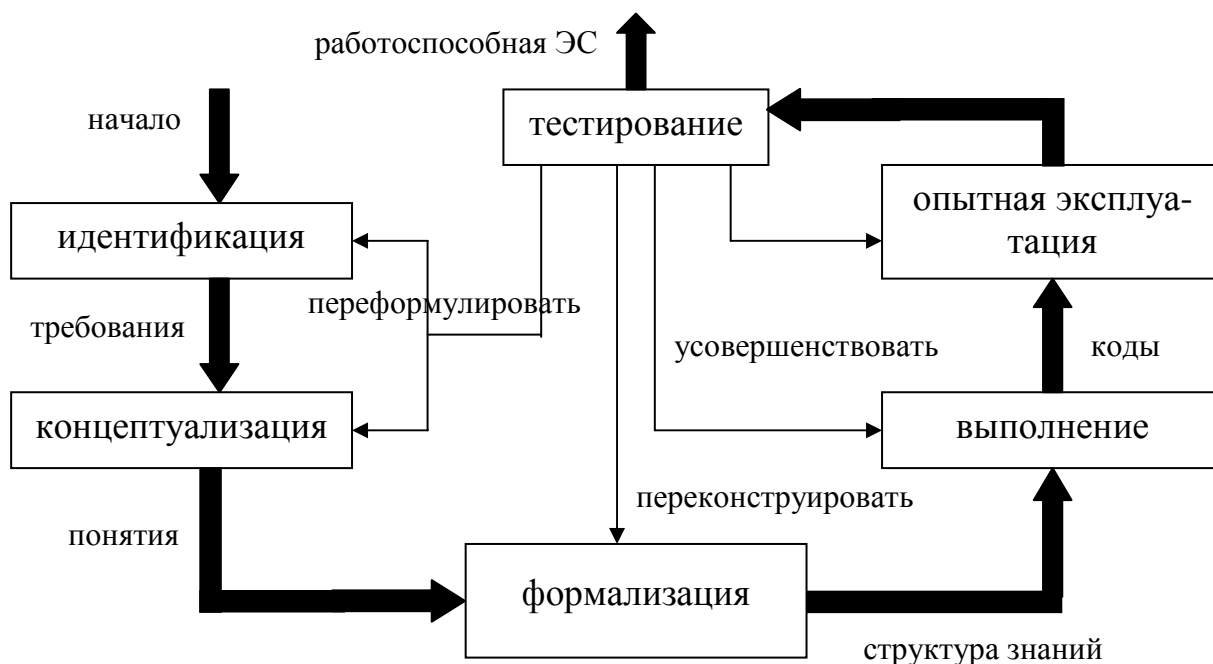


Рис. 2.6. Этапы создания ЭС

На этапе *идентификации* необходимо выполнить следующие действия:

- определить задачи, подлежащие решению и цели разработки;

- определить экспертов и тип пользователей.

На этапе идентификации определяются участники процесса проектирования и их роли, идентифицируется задача (задачи), определяются ресурсы и цели построения системы.

Определение участников и их ролей сводится к определению количества экспертов и инженеров по знаниям, а также форм их взаимоотношения. Обычно в разработке ЭС участвует не менее трех-четырех человек: один эксперт, один или два инженера по знаниям, один программист. К процессу разработки ЭС могут привлекаться и другие участники. Например, инженер по знаниям может привлекать других экспертов, чтобы убедиться в правильности понимания основного эксперта.

Главными участниками на этом этапе являются эксперт и инженер по знаниям. Взаимодействие их между собой происходит в следующих формах:

- эксперт выступает в роли информатора, а инженер по знаниям в роли получателя информации;

- эксперт выполняет роль учителя, а инженер выполняет роль ученика.

Считается, что вторая форма (учитель-ученик) больше соответствует методологии проектирования ЭС.

После выбора участников инженер по знаниям и основной эксперт начинают идентификацию задачи. Идентификация задачи заключается в составлении неформального (вербального) описания решаемой задачи. В этом описании указываются:

- общие характеристики задачи;

- подзадачи, выделяемые внутри данной задачи;

- ключевые понятия (объекты), характеристики и отношения;

- входные и выходные данные;

- предположительный вид решения;

- знания, релевантные решаемой задаче;

- примеры (тесты) решения задачи.

Цель идентификации задачи - дать общую характеристику задачи и структуры поддерживающих ее знаний и таким образом обеспечить начальный импульс для развития базы знаний.

В ходе идентификации задачи необходимо ответить на следующие вопросы (примерный состав):

- какой класс задач будет решать данная ЭС;
- как эти задачи могут быть охарактеризованы или определены;
- на какие подзадачи разбивается каждая задача;
- какие данные они используют;
- каковы основные понятия и взаимоотношения, используемые при формулировании и решении задачи;
- какой вид имеет решение и какие знания используются в нем;
- какие аспекты опыта эксперта существенны при решении задач;
- какие возможны ситуации, препятствующие решению;
- как эти препятствия влияют на ЭС.

При идентификации задачи эксперт и инженер по знаниям работают в тесном контакте. Эксперт дает начальное неформальное описание задачи, которое используется инженером по знаниям для уточнения терминов и ключевых понятий. Затем эксперт дает подробное описание типовых задач, объясняет, как решать эти задачи, какие соображения лежат в основе решений. После нескольких циклов обсуждений инженер по знаниям и эксперт достигают окончательного неформального описания задачи.

- При проектировании ЭС типичными ресурсами являются:
- источники знаний;
  - время разработки;
  - вычислительные средства (возможности ЭВМ и программного инструментального средства);
  - объем финансирования.

Для достижения успеха при построении ЭС эксперт и инженер по знаниям должны использовать все доступные им источники знаний. Для эксперта таковыми являются его предшествующий опыт, книги, конкретные примеры задач и использованных решений. Для инженера по знаниям источниками являются опыт в решении аналогичных задач, существующие методы решения и



представления решений, программное инструментальное средство.

При определении времени разработки необходимо иметь в виду, что при трудоемкости 5 чел.-лет сроки разработки и внедрения ЭС составляют (за редким исключением) не менее года.

Если объем финансирования оказывается недостаточным, то предпочтение может быть отдано не разработке оригинальной новой системы, а модернизации существующей.

Определение целей заключается в формулировании в явном виде целей построения ЭС. При этом необходимо отличать цели, ради которых создается ЭС, от задач, которые она должна решать. Примерами возможных целей являются:

- формализация неформальных знаний экспертов;
- улучшение качества решений, принимаемых экспертом;
- автоматизация рутинных аспектов работы эксперта (пользователя);
- тиражирование знаний эксперта.

Выявленные цели проектирования ЭС образуют дополнительные ограничения, которые необходимо учитывать при выборе подхода к решению задачи.

## **2.8. Приобретение и формализация знаний**

*Приобретением знаний* начинается выявление знаний из источников и преобразование их в нужную форму, а также перенос в базу знаний ИС. Источниками знаний могут быть книги, архивные документы, содержимое других баз знаний и т.п., т.е. некоторые *объективизированные знания*, переведенные в форму, которая делает их доступными для потребителя. Другим типом знаний являются *экспертные знания*, которые имеются у специалистов, но не зафиксированы во внешних по отношению к нему хранилищах. Экспертные знания являются *субъективными*. Еще одним видом субъективных знаний являются *эмпирические знания*. Такие знания могут добываться ИС путем наблюдения за окружающей средой (если у ИС есть средства наблюдения).

Ввод в базу знаний объективизированных знаний не представляет особой проблемы, выявление и ввод субъективных и особенно экспертных знаний достаточно трудны. Чтобы разрабо-

тать методологию приобретения субъективных знаний, получаемых от эксперта, надо четко различать две формы репрезентации знаний. Одна форма связана с тем, как и в каких моделях хранятся эти знания у человека-эксперта. При этом эксперт не всегда осознает полностью, как репрезентированы у него знания. Другая форма связана с тем, как инженер по знаниям, проектирующий ИС, собирается их описывать и представлять. От степени согласованности этих двух форм репрезентации между собой зависит эффективность работы инженера по знаниям.

В когнитивной психологии изучаются формы репрезентации знаний (*когнитивные структуры знаний*) характерные для человека. Примерами могут служить: представление класса понятий через его элементы (например, понятие «птица» репрезентируется рядом «чайка, воробей, скворец, ... »); представление понятий класса с помощью базового прототипа, отражающего наиболее типичные свойства объектов класса (например, понятие «птица» репрезентируется прототипом «нечто с крыльями, клювом, летает .. »); представление с помощью признаков (для понятия «птица», например, наличие крыльев, клюва, двух лап, перьев ).

Кроме понятий репрезентируются и отношения между ними. Как правило, отношения между понятиями определяются процедурным способом, а отношения между составляющими понятий (определяющими структуру понятия) — декларативным способом. Наличие двух видов описаний заставляет в моделях представления знаний одновременно иметь оба компонента, например семантическую сеть и продукционную систему, как это представлено в когнитивной модели.

При приобретении знаний важную роль играют так называемое *поле знаний*, в котором содержатся основные понятия, используемые при описании предметной области, и свойства всех отношений, используемых для установления связей между понятиями. Поле знаний связано с концептуальной моделью проблемной области, в которой еще не учтены ограничения, которые неизбежно возникают при формальном представлении знаний в базе знаний. Переход от описания некоторой области в поле знаний к описанию в базе знаний аналогичен переходу от концептуальной модели базы данных к ее логической схеме, когда уже зафиксирована система управления базой данных. Важно отметить, что

переход непосредственно к формальным представлениям в базе знаний без этапа концептуального описания в поле знаний приводит к многочисленным ошибкам что замедляет процесс формирования базы знаний ИС.

Возможны три режима взаимодействия инженера по знаниям с экспертом-специалистом:

протокольный анализ,

интервью;

игровая имитация профессиональной деятельности.

### **Режимы взаимодействия эксперта и инженера по знаниям.**

Протокольный анализ заключается в фиксации (например, путем записи на магнитную ленту) «мыслей вслух» эксперта во время решения проблемы и в последующем анализе полученной информации. В режиме интервью инженер по знаниям ведет с экспертом активный диалог, направляя его в нужную сторону. При игровой имитации эксперт помещается в ситуации, похожие на те, в которых протекает его профессиональная деятельность. Наблюдая за его действиями в различных ситуациях, инженер по знаниям, формирует свои соображения об экспертных знаниях, которые впоследствии могут быть уточнены с экспертом в режиме интервью. Принципы игровой имитации нашли применение в разнообразных деловых играх, специальных тренажерах.

Каждый из упомянутых способов извлечения знаний имеет свои преимущества и недостатки. Так, при анализе протоколов инженеру по знаниям нелегко отделить понятия, важные для включения в словарь предметной области, от тех, которые при «мыслях вслух» появляются случайно. Кроме того, в протоколах обнаруживаются пробелы, когда рассуждение эксперта как бы прерывается и продолжается уже на основе пропущенных шагов вывода. Заполнение подобных лакун возможно лишь в режиме интервью. Таким образом, во всех трех подходах к извлечению знаний из экспертов необходим этап интервью, что делает его одним из важнейших методов приобретения знания.

Существует не менее двух десятков стратегий интервьюирования. Наиболее известны три:

разбиение на ступени;

репертуарная решетка;  
подтверждение сходства.

При разбиении на ступени эксперту предлагается назвать наиболее важные, по его мнению, понятия предметной области и указать между ними отношения структуризации, т.е. отношения типа «род—вид», «элемент—класс», «целое-часть» и т. п. Эти понятия используются на следующем шаге опроса как базовые. Стратегия нацелена на создание иерархии понятий предметной области, выделение в понятиях тесно связанных между собой групп — *таксонов* (кластеров).

Стратегия репертуарной решетки направлена на выявление характеристических свойств понятий, позволяющих отделять одни понятия от других. Методика состоит в предъявлении эксперту троек понятий с предложением назвать признаки для каждой двух понятий, которые отделяли бы их от третьего. Так как каждое понятие входит в несколько троек, то на основании такой процедуры происходит уточнение объемов понятий и формируются «симптокомплексы» понятий, с помощью которых эти понятия могут идентифицироваться в базе знаний.

Стратегия подтверждения сходства состоит в том, что эксперту предлагается установить принадлежность каждой пары понятий из предметной области к некоторому отношению сходства (толерантности). Для этого эксперту задается последовательность достаточно простых вопросов, цель которых заключается в уточнении того понимания сходства, которое вкладывает эксперт в утверждение о сходстве двух понятий предметной области.

Процесс взаимодействия инженера по знаниям (аналитика) с экспертом-специалистом включает три основных этапа.

1. Подготовительный этап. Для успеха общения оба участника должны тщательно подготовиться к диалогу или игре. Желательно, чтобы эксперт был не только компетентным специалистом, но и заинтересованным (морально или материально) лицом в достижении конечной цели—построении ИС. Он должен быть доброжелателен к аналитику и уметь объяснять свои знания (наилучший случай, когда эксперт имеет опыт преподавательской работы).

Аналитику необходимо: глубоко познакомиться со специальной литературой по предметной области, чтобы не задавать

очень «глупых» вопросов (просто «глупые» вопросы бывают чрезвычайно полезны), а также увеличить количество «пакетов ожиданий», уметь слушать и грамотно задавать вопросы, настроиться на роль «ученика», а не «экзаменатора»; разбираться в моделях когнитивной психологии, а также в моделях представления знаний, чтобы из знаний эксперта выделять четкие структуры.

В любой совместной деятельности большое значение имеют психологические качества исследователей, такие как личность, манера поведения, стиль научного мышления. Существуют различные классификации научных работников. В качестве примера приведем следующую:

инициатор — быстро реагирует на перспективные проблемы, т.е. один из первых ощущает необходимость решения проблемы с элементами неопределенности;

диагност — способен к быстрой оценке сильных и слабых сторон решения задачи;

эрудит — наделен исключительной памятью, отличается повышенным вниманием к деталям и стремлением к упорядоченности;

ремесленник — способен воплощать в жизнь плохо оформленные идеи других;

эстет — стремится исследовать проблемы, приводящие к изящным решениям, не склонен к кропотливому труду;

методолог — заинтересован методологическими аспектами исследований;

независимый—стремится к индивидуальному решению проблем;

фанатик—самоотверженно увлечен своей научной проблемой, того же требует и от окружающих.

Принадлежность научного работника к тому или иному типу определяется с помощью косвенных методик (тестов личности, интеллекта, когнитивных стилей, проектных методик).

Для роли эксперта наиболее предпочтительны инициатор, эрудит, диагност и ремесленник (в паре с аналитиком-эрудитом), а для роли аналитика—диагност, методолог, эрудит, инициатор. При этом наилучшее сочетание дают сочетания разных типов. Благодаря различиям в подходах к решению задачи, в точках зрения, стиле мышления, восприятия, памяти и т. п. участники в та-

кой паре с разных сторон подходят к поставленной цели, в результате увеличивается общее количество гипотез, идей, альтернативных вариантов, а следовательно, обогащается поле знания. Однако не все сочетания даже из приемлемых типов улучшают взаимодействие, а некоторые типы (например, фанатик, эстет, независимый, ремесленник) часто слабо приспособлены для творческого взаимодействия, что приводит к возникновению скрытых и явных конфликтов, которые усложняют процесс продуктивного общения.

Важное значение имеет также лидерство в паре. В ходе любого диалога одна сторона обычно занимает позицию ведущей, чаще эту роль берет интервьюер, т.е. аналитик. Роль лидера в диалоге позволяет аналитику направлять и систематизировать процесс создания поля знания, не давая эксперту «размыться» или излишне детализовать процесс. С другой стороны, догматизм и настойчивость могут привести к неадекватному полю. Имеет место также эффект «фасада», т.е. желание эксперта не ударить «в грязь лицом» перед аналитиком, и отсюда генерирование неподтвержденных гипотез.

2. Установление «общего кода». Для создания лингвистического альянса взаимодействия участники взаимодействия должны пытаться сократить «расстояние» между объектом (т. е. исследуемой предметной областью) и аналитиком. Необходимо определить главные понятия, т. е. выработать словарную основу базы знаний; уровень детализации; взаимосвязи между понятиями.

3. Гносеологический этап. На этом этапе происходит выяснение закономерностей, присущих предметной области, условий достоверности и истинности утверждений, структурирование за счет введения отношений и т. п. Этот этап является определяющим во взаимодействии аналитика и эксперта. В процессе анализа игры или диалога вербализуется и формализуется знание эксперта и зачастую для него самого порождается новое знание. Репрезентация внешнего мира в его памяти получает материальное воплощение в форме поля знаний.

В процессе извлечения знаний сначала желательно получить от эксперта поверхностные знания (такие, например, как репрезентация признаков), постепенно переходя к глубинным структу-

рам и более абстрактным понятиям (таким, например, как прототипы).

При формировании поля знаний учитываются особенности эмпирического знания: модальность, противоречивость, неполнота и т. д.

Аналитик должен за частным всегда видеть общее, т. е. строить цепочки «факт — обобщенный факт — эмпирический закон — теоретический закон». Центральное звено цепочки — формализация эмпирики. При этом иногда основным на этапе формализации становится не извлечение «слепых» непонятных связей, а понимание внутренней структурной связи понятий предметной области. Искусство аналитика состоит в стремлении к созданию ясной и понятной модели проблемной области.

Следует также учитывать, что эксперты в проблемной области не всегда опираются на логические рассуждения. В их представлениях о проблемной области и методах решения задач, характерных для нее, широкое применение находят ассоциативные рассуждения и рассуждения правдоподобия. Опишем примерную методику работы с экспертом по формированию поля знаний.

### **Подготовительный этап**

1. Четкое определение задач проектируемой системы (сужение поля знаний):

определение, что на входе и выходе; определение режима работ — консультации, обучение и др.

2. Выбор экспертов: определение количества экспертов; выбор уровня компетентности (не всегда хорошо выбирать самый высокий уровень сразу); определение способов и возможности заинтересовать экспертов в работе; тестирование экспертов.

3. Знакомство аналитика со специальной литературой в предметной области

4. Знакомство аналитика и экспертов (в дальнейшем для простоты будем считать, что эксперт один).

5. Знакомство эксперта с популярной литературой по искусственному интеллекту (желательно, но необязательно).

6. Попытка аналитика создать поле знаний первого приближения априорным знаниям из литературы (прототип поля знаний).

### **Основной этап**

1. «Накачка» поля знаний:

а) в зависимости от предметной области выбор способа интервьюирования;

б) протоколирование мыслей вслух или запись на магнитофон рассуждений эксперта (аналитик по возможности не должен пока вмешиваться в рассуждения).

2. «Домашняя работа». Попытка аналитика выделить некоторые причинно-следственные связи в рассуждениях эксперта; построение словаря предметной области (возможно, на карточках) и подготовка вопросов к эксперту.

3. «Подкачка» поля зрения. Обсуждение с экспертом прототипа поля знаний и домашней работы, а также ответы на вопросы аналитика.

4. Формализация концептуальной модели.

5. Построение поля знаний второго приближения.

6. При необходимости повторение пп. 1б, 2, 3, 4 и 5.

На этапе *концептуализации*:

проводится содержательный анализ предметной области;

выделяются основные понятия и их взаимосвязи;

определяются методы решения задач.

На этом этапе инженер по знаниям и эксперт объединяют ключевые понятия, отношения, упомянутые на этапе идентификации и характеристики, необходимые для описания процесса решения задачи. В ходе этого этапа следует ответить на следующие вопросы: какие имеются типы данных; что является исходными данными и что должно быть выведено; какие стратегии и гипотезы используются; каковы виды взаимосвязей (типы отношений) между объектами предметной области (иерархия, причина-следствие, часть-целое и т.п.); какие процессы участвуют в решении задачи; каков состав знаний, используемых в ходе решения задачи, и состав знаний, используемых для объяснения решения.



Для определения перечисленных характеристик задачи инженеру по знаниям целесообразно составить детальный протокол действий и рассуждений эксперта в процессе решения хотя бы одной конкретной задачи. Протокол обеспечивает инженера по знаниям словарем терминов и некоторым приблизительным представлением о тех стратегиях, которые использует эксперт. Кроме того, он помогает ответить на многие другие вопросы, возникающие в ходе разработки.

На этом этапе инженер по знаниям рассматривает вопросы, относящиеся к представлению знаний и методам решения, но говорить о выборе конкретных способов и методов здесь еще рано. Необходимо обеспечить достаточность введенных ключевых понятий и отношений для описания всех имеющихся примеров решения рассматриваемой проблемы. Не следует производить окончательный выбор представления знаний и инструментария, хотя уже на этом этапе инженер по знаниям может попытаться применить имеющиеся у него идеи для создания прототипа некоторой подзадачи.

На этапе *формализации*:

выбираются программные средства разработки ЭС;  
определяются способы представления всех видов знаний;  
формализуются основные понятия.

На данном этапе все ключевые понятия и отношения, введенные на этапе концептуализации, выражаются на некотором формальном языке, выбранном инженером по знаниям. Инженер по знаниям на этом этапе играет главную роль. Он определяет, подходят ли имеющиеся инструментальные средства для решения рассматриваемой проблемы или необходимы оригинальные разработки.

Выходом данного этапа является описание того, как рассматриваемая задача может быть представлена в выбранной или разработанной формальной модели.

В процессе формализации выделяются: пространство поиска (гипотез), модель процесса решения задачи, свойства данных рассматриваемой задачи.

Для определения структуры пространства поиска необходимо формализовать концепции (объекты, их характеристики и зна-

чения) и определить, как они могут связываться между собой при образовании гипотез, используемых для направления поиска. При этом следует ответить на следующие вопросы: являются понятия примитивными или имеют внутреннюю структуру; необходимо ли представлять причинные и пространственно-временные отношения между понятиями и должны ли они быть представлены явно; необходима ли иерархия гипотез; относится коэффициент определенности (уверенности) только к окончательным гипотезам или он необходим для промежуточных гипотез; следует ли использовать различные уровни абстракции.

Построение модели процесса решения задачи является важным шагом на этапе формализации, так как наличие модели позволяет генерировать решение. Модели могут быть поведенческими и математическими. Если эксперт использует при рассуждении или обосновании решения хотя бы упрощенную поведенческую модель, то она позволяет вырабатывать важные понятия и отношения. Если часть концептуальной структуры процесса решения описывается математической моделью (аналитической или статистической), то она может быть непосредственно включена в ЭС как для формирования решения, так и для объяснения причинных отношений, обнаруживаемых в базе знаний.

Очень важным является выявление природы данных. Для этого следует ответить на следующие вопросы: являются ли данные редкими, недостаточными, избыточными; имеется ли неопределенность данных; зависит ли логическая интерпретация данных от порядка их появления во времени; какова стоимость приобретения данных; как приобретаются или извлекаются данные и какие вопросы при этом задаются; какие необходимые характеристики объектов могут быть извлечены из входного сообщения; являются ли данные полными и непротиворечивыми, а также надежными (ненадежными), однозначными (многозначными) и т.д.

Результатом этапа формализации является частичная спецификация для построения прототипа базы знаний. Спецификация содержит организующую схему, явное представление основных концепций и связей задачи. На этом же этапе выбираются языки или инструментальные средства.

На этапе *выполнения* (наиболее важном и трудоёмком) осуществляется наполнение экспертом БЗ, при котором процесс приобретения знаний разделяют:

на "извлечение" знаний из эксперта;

на организацию знаний, обеспечивающую эффективную работу ЭС

на представление знаний в виде, понятном для ЭС.

Процесс приобретения знаний осуществляется инженером по знаниям на основе деятельности эксперта (см. табл. 1).

Этот этап включает перевод формализованных знаний в схему представления, определяемую выбранным языком. В результате появляется рабочая программа (прототип). Дальнейшее совершенствование схемы представления знаний посредством рабочей программы, осуществляемое по результатам этапов тестирования и опытной эксплуатации, приводит к появлению новых прототипов и, в конечном итоге, созданию конечного продукта, пригодного для промышленного использования.

Разработка первой версии прототипа ЭС начинается, как только становится хорошо понят первый тестовый пример.

Прототипная БЗ создается с помощью различных средств: текстовых редакторов, интеллектуальных редакторов, программ для приобретения знаний. Если существующие программные средства не подходят, то инженер по знаниям и программист должны их разработать.

Не следует откладывать процесс приобретения знаний до полного завершения программирования. Первый прототип ЭС - демонстрационный должен появиться через несколько месяцев (обычно два - три), а не через годы после начала работы. В демонстрационном прототипе должна реализоваться простейшая процедура вывода. Основная цель разработки демонстрационного прототипа - получить решение задачи, не заботясь об эффективности.

После разработки первого прототипа круг задач, решаемых системой, расширяется. Развитие версии первого прототипа осуществляется путем добавления "дружественного" интерфейса, средств исследования цепочек выводов, базы знаний в целом, средств для сбора замечаний пользователей, средств для хранения библиотеки задач, решаемых системой. Расширенная версия

первого прототипа может рассматриваться как исследовательский прототип.

Выполнение экспериментов с исследовательским прототипом, анализ пожеланий и замечаний служат отправной точкой для создания действующего прототипа ЭС. Этот процесс - итеративный и может продолжаться от нескольких месяцев до нескольких лет в зависимости от сложности проблемной области, гибкости выбранного представления и степени соответствия управляющего механизма решаемым задачам.

Развитие действующего прототипа осуществляется в направлении повышения эффективности решения задач, для чего производится:

- анализ функционирования системы при значительном расширении базы знаний;

- исследование возможностей системы в решении более широкого круга задач и принятие мер для обеспечения таких возможностей;

- разработка системы ввода-вывода, осуществляющей анализ или синтез предложений ограниченного естественного языка, что позволяет пользователю взаимодействовать с ЭС в форме, близкой к форме стандартных учебников для данной области.

Если система, в которой решены перечисленные выше задачи, успешно прошла этап тестирования, то она может классифицироваться как промышленная система.

Следующий этап жизненного цикла ЭС – это *опытная эксплуатация* и *тестирование*. На этапе опытной эксплуатации проверяется пригодность ЭС для конечных пользователей. По результатам данной проверки может потребоваться разработка новых специализированных версий, учитывающих особенности проблемных областей. В свою очередь, на этапе тестирования эксперт и инженер по знаниям с использованием диалоговых и объяснительных средств проверяют компетентность ЭС. Процесс тестирования продолжается до тех пор, пока эксперт не решит, что система достигла требуемого уровня компетентности.

Этап тестирования. Как только ЭС оказывается в состоянии обработать от начала до конца несколько примеров, необходимо

начинать ее проверку на более широком круге примеров, чтобы определить недостатки базы знаний и процедур вывода. Иными словами, следует проводить тестирование ЭС, задачей которого является оценка выбранного способа представления знаний и ЭС в целом. Подбор тестовых примеров возлагается на инженера по знаниям.

Основными источниками неудач в работе системы обычно являются: тестовые примеры; ввод-вывод данных; правила вывода; управляющие стратегии.

Наиболее очевидной причиной неудачной работы ЭС являются недостаточно показательные тестовые примеры. В худшем случае они могут оказаться вообще вне проблемной области, на которую рассчитана система. Однако часто бывает, что тестовые примеры находятся в рамках рассматриваемой проблемной области, но являются однородными и не позволяют охватить все нюансы. В этом случае целесообразно осуществить классификацию тестовых примеров по подпроблемам с выделением стандартных случаев, границ трудных ситуаций и т.д.

Неудачи системы, определяемые вводом-выводом данных, могут быть вызваны недостаточной проработанностью методов приобретения знаний в ходе диалога с экспертом, а также выбранных способов представления вводимых и выводимых данных. Если в ходе диалога с экспертом задавались неправильные (некорректные) вопросы или собрана неполная информация, то методы приобретения знаний могут не давать нужных результатов. Ошибки при вводе могут возникать из-за неудобного для пользователя входного языка. В ряде случаев бывает необходим ввод не только в печатной, но и в графической или звуковой форме. Выходные сообщения могут быть непонятными пользователю (эксперту) либо по причине своего чрезвычайно большого или недостаточно малого количества, либо ввиду принятого неудобного способа упорядочивания, либо из-за использования не подходящей для пользователя лексики и (или) уровня абстракции.

Типичным источником ошибок в рассуждениях являются правила вывода. Основная причина этого - отсутствие учета зависимостей между правилами, так как правила редко бывают неза-

висимыми друг от друга. Среди других причин ошибок можно отметить ошибочность, противоречивость и неполноту правил.

Весьма часто к ошибкам в работе ЭС приводят управляющие стратегии. Последовательность, в которой данные рассматриваются ЭС, не только влияет на эффективность работы системы, но и может приводить к изменению конечного результата. Кроме того, недостатки в управляющих стратегиях могут привести к чрезмерно сложным заключениям и объяснениям ЭС.

Критерии, по которым оценивается ЭС, зависят от того, с чьей точки зрения дается оценка. Если, например, тестируется исследовательский прототип ЭС, то оценка дается с точки зрения эксперта, для которого важна в первую очередь полнота и безошибочность правил вывода. При тестировании промышленной системы оценка производится с точки зрения инженера по знаниям, которого интересует эффективность работы ЭС. При тестировании ЭС после опытной эксплуатации оценка дается с точки зрения пользователя, которого в первую очередь интересуют удобство работы и получение практической пользы.

Этап опытной эксплуатации. На данном этапе проверяется пригодность ЭС для конечного пользователя. К этому этапу переходят лишь после того, как система, по мнению эксперта, успешно решает практически все требуемые задачи. Иначе ошибки в решениях будут создавать у пользователя отрицательное представление о ЭС.

Пригодность ЭС для пользователя определяется и удобством работы с ней, и ее полезностью. Под удобством работы ЭС понимается естественность взаимодействия с ней, гибкость системы и ее устойчивость к ошибкам. Под полезностью ЭС понимается ее способность в ходе диалога определять потребности пользователя, выявлять и устранять причины неудач в работе, а также удовлетворять потребности пользователя.

Работу ЭС на этапе опытной эксплуатации следует организовать на рабочем месте пользователя.

По результатам эксплуатации может потребоваться не только модификация программ и данных, но и изменение устройств ввода-вывода из-за их неприемлемости для пользователя. По результатам этого же этапа принимается решение о переносе ЭС на другие ЭВМ.

После успешного завершения этапа опытной эксплуатации ЭС классифицируется как коммерческая, а процесс разработки ЭС заданного назначения может считаться законченным.

Приемы извлечения знаний:

Наблюдение - инженер наблюдает, не вмешиваясь, за тем, как эксперт решает реальную задачу;

обсуждение задачи - инженер на предоставленном множестве задач неформально обсуждает с экспертом данные, знания и процедуры решения;

описание задачи - эксперт описывает решение задач для типичных запросов;

анализ решения - эксперт комментирует получаемые результаты решения задачи, детализируя ход рассуждений;

проверка системы - эксперт предлагает инженеру перечень задач для решения (от простых до сложных), которые решаются разработанной системой;

исследование системы - эксперт исследует и критикует структуру БЗ и работу механизма вывода;

оценка системы - инженер предлагает новым экспертам оценить решения разработанной системы

На всех этапах разработки инженер по знаниям играет активную роль, а эксперт – пассивную. По мере развития самообучающихся свойств ЭС роль инженера по знаниям уменьшается, а активное поведение пользователя-эксперта, заинтересованного в эффективной работе ЭС, возрастает.

Процесс создания ЭС не сводится к строгой последовательности этих этапов, так как в ходе разработки приходится неоднократно возвращаться на более ранние этапы и пересматривать принятые там решения.

### **Контрольные вопросы:**

1. Что такое экспертная система.
2. Классифицируйте ЭС по сложности и мощности.
3. Приведите примеры возможных целей ЭС.
4. Перечислите основные этапы жизненного цикла ЭС.
5. Опишите роли эксперта и инженера по знаниям.
6. Перечислите основные приемы извлечения знаний.

### Тема 3. НЕЧЕТКИЕ ЗНАНИЯ

Глава посвящена понятию нечеткости знаний и способу их обработка с использованием коэффициентов уверенности. Отдельные вопросы касаются неопределенности в ЭС.

#### 3.1. Понятие нечеткости знаний

Во всех предметных областях, где ИИС будут эффективным средством построения информационных систем нового поколения, существуют плохо (некорректно) поставленные задачи. К ним относятся проблемы, строгое описание которых либо невозможно, либо чересчур сложно. Точные знания об этих проблемах или вообще нельзя получить, или нельзя получить сразу, и приходится последовательно приближаться к более или менее полному набору знаний. Т.е. знания чаще всего нечетки.

Смысл термина *нечеткость* до настоящего времени не определен, и в разных источниках встречаются разные его трактовки. Невозможно привести однозначное определение термина *нечеткость*, но можно объединить разные виды нечеткостей в некоторые группы следующим образом: недетерминированность выводов, многозначность, ненадежность, неполнота, неточность.

**Недетерминированность выводов.** Практически во всех системах искусственного интеллекта знания накапливаются фрагментарно, и нельзя априори определить цепочку логических выводов, в которых они будут использоваться. Последовательность действий при поиске решения заранее не может быть определена, и необходимо методом проб и ошибок выбрать некую цепочку выводов, а в случае неуспеха организовать перебор с возвратами для поиска другой цепочки и т.д.

**Многозначность.** О *многозначности* знаний говорят тогда, когда один и тот же элемент знаний (понятие, символ, звук, изображение и т.п.) может быть интерпретирован по-разному. Многозначность интерпретации - обычное явление при понимании естественных языков и распознавании изображений и речи.

**Ненадежными** являются те знания, представить которые двумя значениями - истина или ложь - невозможно или трудно. В



современной физике и технике такую ненадежность представляют вероятностью, подчиняющейся законам Байеса. Но при обработке знаний было бы нелогично иметь дело со степенью надежности, приписанной знаниям изначально, и поэтому применяются специфические методы работы с ненадежными знаниями.

**Неполнота знаний.** Полностью описать окружающий мир чрезвычайно сложно. Содержимое базы знаний по любой предметной области является *неполным*, поскольку можно (хотя и трудно) перечислить все верные знания в данной области, но невозможно перечислить и разумно определить неверные знания. Поэтому целесообразно в БЗ определять только заведомо верные знания, а любые утверждения, которые не определены относить к ложным. Это называется *гипотезой закрытого мира*.

Одним из методов изучения знаний без уточнения их границ является *теория нечетких множеств*, которая была предложена Заде в 1965 году и продолжает развиваться.

**Обработка нечетких знаний с использованием коэффициентов уверенности**

Результат любой операции над переменной с нечетким значением является нечетким, т.е. и в его значении можно быть уверенным лишь с некоторой степенью (коэффициентом). Определение этого коэффициента именуется *исчислением коэффициента уверенности* и базируется на нечеткой логике.

Коэффициент уверенности (КУ) есть целое число в диапазоне от 0 до 100, которое придается значению переменной при рассмотрении вопроса о его достоверности и указывает степень уверенности в том, что переменная имеет именно это значение.

**Обработка нечетких множеств**

Если значением переменной является *нечеткое множество*, т.е. если переменная одновременно может иметь несколько значений, каждое со своим значением КУ, такая переменная носит название *нечеткой переменной*.

### 3.2. Неопределенность в экспертных системах

**Неопределенности в ЭС и проблемы порождаемые ими**

В жизни часто приходится оценивать гипотезы, для которых имеется неполная или недостаточная информация. Иногда трудно

сделать точные оценки, но, не смотря на неопределенность, мы принимаем разумные решения. Чтобы ЭС были полезными, они тоже должны уметь это делать. Неопределенность в ЭС она может быть двух типов: неопределенность в *истинности самой посылки*; неопределенность *самого правила*.

Существуют четыре важные проблемы, которые возникают при создании ЭС с неопределенными знаниями.

Как количественно выразить степень определенности при установлении истинности (ложности) некоторой части данных?

Как выразить степень поддержки заключения конкретной посылкой?

Как использовать совместно две (или более) посылки, независимо влияющие на заключение?

Как быть в ситуации, когда нужно обсудить цепочку вывода для подтверждения заключения в условиях неопределенности?

Прежде всего, рассмотрим возможности использования теории вероятности при вводе в условиях неопределенности.

### **Теория субъективных вероятностей**

Основное понятие вероятности настолько естественно, что оно играет значительную роль в повседневной жизни. Разговоры, касающиеся вероятности дождя или хорошего урожая в огороде, часто встречаются в нашей жизни. Понятие вероятности было разработано несколько столетий назад. Но уже тысячи лет человек использует такие слова, как «может быть», «шанс», «удача» или иные их эквиваленты в разговорном языке.

Однако математическая теория вероятностей была сформулирована относительно недавно (около 1660 года). Вероятность события классически определяется как отношение случаев, в которых данное событие происходит, к общему числу наблюдений.

Однако возможны и другие определения. В настоящее время существует несколько интерпретаций теории вероятностей. Рассмотрим три наиболее доминирующих взгляда.

**Объективистский взгляд.** Заключается в том, что рассматривает вероятность отношения исходов ко всем наблюдениям в течении длительного времени. Другими словами этот подход основан на законе больших чисел, гарантирующим то, что при наличии достаточно большого количества наблюдений частота ис-

ходов, интересующего события будет стремиться к объективной вероятности.

***Персонафицированный, субъективистский или основанный на суждениях взгляд.*** Заключается в том, что вероятностная мера рассматривается как степень доверия того, как отдельная личность судит об истинности некоторого высказывания. Этот взгляд постулирует, что данная личность имеет в некотором смысле отношение к этому событию. Но это не отрицает возможности того, что две приемлемые личности могут иметь различные степени доверия для одного и того же суждения. Термин «байесовский» часто используется как синоним субъективной вероятности.

***Необходимый или логический.*** Характеризуется тем, что вероятностная мера расширяется на множество утверждений, имеющих логическую связь такую, что истинность одного из них может выводиться из другого. Другими словами, вероятность измеряет степень доказуемости логически выверенного заключения. Такой взгляд можно рассматривать как расширение обычной логики.

Эти вероятностные интерпретации используют и различные схемы вывода. Однако существует всего две школы вероятностных расчетов: школа Паскаля (или общепринятая), школа Бэкона (или индуктивная). Расчеты по Паскалю используют байесовские правила для проверки и обработки мер доверия. Вычисления по Бэкону используют правила логики для доказательства или опровержения гипотез. Таким образом, общепринятые вероятности (по Паскалю) не могут быть получены из индуктивных вероятностей (по Бэкону) и, наоборот. Объективистский и субъективистский взгляды используют расчеты по Паскалю. Те, кто поддерживают логические выводы, используют расчеты по Бэкону.

Существуют ЭС, построенные на обоих из этих направлений. Однако в ЭС БЗ накапливают человеческие знания, поэтому для представления знаний экспертов с учетом вероятностей наиболее подходящим является интерпретация на основе субъективных мер доверия. В результате чего и большинство современных ЭС, использующих теорию вероятностей, являются "байесовскими".

## **Теорема Байеса как основа управления неопределенностью**

Если  $A$  и  $B$  являются непересекающимися множествами, то объединение множеств соответствует сумме вероятностей, а пересечение – произведению вероятностей:

$$p(A \cup B) = p(A) + p(B) \quad \text{и} \quad p(A \cap B) = p(A) * p(B).$$

Без предположения независимости эта связь является неточной и формулы должны содержать дополнительные члены включения и исключения (так, например,  $p(A \cup B) = p(A) + p(B) - p(A \cap B)$ ).

Продолжая теоретико-множественное обозначение,  $B$  можно записать как:

$$B = (B \cap A) \cup (B \cap \bar{A}).$$

Т.к. это объединение явно непересекающееся, то:

$$\begin{aligned} p(B) &= p((B \cap A) \cup (B \cap \bar{A})) = p(B \cap A) + p(B \cap \bar{A}) = \\ &= p(B|A) \times p(A) + p(B|\bar{A}) \times p(\bar{A}). \end{aligned}$$

Если вернуться к обозначению событий, а не множеств, то последнее равенство может быть подставлено в правило Байеса:

$$p(A|B) = \frac{p(B|A) \times p(A)}{p(B|A) \times p(A) + p(B|\bar{A}) \times p(\bar{A})}.$$

Это равенство является основой для использования теории вероятности в управлении неопределенностью. Оно обеспечивает путь для получения условной вероятности события  $B$  при условии  $A$ . Это соотношение позволяет ЭС управлять неопределенностью и «делать вывод вперед и назад».

### **Логический вывод на основе субъективной вероятности.**

Простейший логический вывод.

Рассмотрим случай, когда все правила в ЭС отражаются в форме:

*Если  $\langle H \text{ является истинной} \rangle$  То  $\langle E \text{ будет наблюдаться с вероятностью } p \rangle$ .*

Очевидно, если  $H$  произошло, то это правило говорит о том, что событие  $E$  происходит с вероятностью  $p$ . Но что будет, если состояние  $H$  неизвестно, а  $E$  произошло? Использование теоремы Байеса позволяет вычислить вероятность того, что  $H$  истинно. Замена « $A$ » и « $B$ » на « $H$ » и « $E$ » не существенна для формулы Бай-

еса, но с её помощью мы можем покинуть общую теорию вероятности и перейти к анализу вероятностных вычислений в ЭС. В этом контексте:

$H$  – событие, заключающееся в том, что данная гипотеза верна;

$E$  – событие, заключающееся в том, что наступило определённое доказательство (свидетельство), которое может подтвердить правильность указанной гипотезы.

Переписывая формулу Байеса в терминах гипотез и свидетельств, получим:

$$p(H|E) = \frac{p(E|H) \times p(H)}{p(E|H) \times p(H) + p(E|\neg H) \times p(\neg H)}.$$

Это равенство устанавливает связь гипотезы со свидетельством и, в то же время, наблюдаемого свидетельства с пока ещё не подтверждённой гипотезой. Эта интерпретация предполагает также определение априорной вероятности гипотезы  $p(H)$ , назначаемой  $H$  до наблюдения или получения некоторого факта.

В ЭС вероятности, требуемые для решения некоторой проблемы, обеспечиваются экспертами и запоминается в БЗ. Эти вероятности включают:

априорные вероятности всех возможных гипотез  $p(H)$ ;

условные вероятности возникновения свидетельств при условии существования каждой из гипотез  $p(E/H)$ .

Так, например, в медицинской диагностике эксперт должен задать априорные вероятности всех возможных болезней в некоторой медицинской области. Кроме того, должны быть определены условные вероятности проявления тех или иных симптомов при каждой из болезней. Условные вероятности должны быть получены для всех симптомов и болезней, предполагая, что все симптомы независимы в рамках одной болезни.

Два события  $E_1$  и  $E_2$  являются условно независимыми, если их совместная вероятность при условии некоторой гипотезы  $H$  равна произведению условных вероятностей этих событий при условии  $H$ :

$$p(E_1 E_2|H) = p(E_1|H) \cdot p(E_2|H).$$

Вероятность  $p(H_i|E_j \dots E_k)$  является апостериорной вероятностью гипотез  $H_i$  по наблюдениям  $(E_j \dots E_k)$ , т.е. дает сравнительное

ранжирование всех возможных гипотез. Результатом вывода ЭС является выбор гипотезы с наибольшей вероятностью.

Однако, приведённая выше формула Байеса ограничена в том, что каждое свидетельство влияет только на одну гипотезу. Можно обобщить это выражение на случай множественных гипотез ( $H_1 \dots H_m$ ) и множественных свидетельств ( $E_1 \dots E_n$ ). Вероятности каждой из гипотез при условии возникновения некоторого конкретного свидетельства  $E$  можно определить из выражения:

$$p(H_i | E) = \frac{p(E | H_i) \times p(H_i)}{\sum_{k=1}^m p(E | H_k) \times p(H_k)}, \quad i = \overline{1, m};$$

а в случае множественных свидетельств:

$$p(H_i | E_1 E_2 \dots E_n) = \frac{p(E_1 E_2 \dots E_n | H_i) \times p(H_i)}{\sum_{k=1}^m p(E_1 E_2 \dots E_n | H_k) \times p(H_k)}, \quad i = \overline{1, m}.$$

К сожалению данное выражение имеет ряд недостатков. Так, знаменатель требует от нас знания условных вероятностей всех возможных комбинаций свидетельств и гипотез, что делает правило Байеса малоприменимым для ряда приложений. Однако в тех случаях, когда возможно предположить условную независимость свидетельств, правило Байеса можно привести к более простому виду:

$$p(H_i | E_1 E_2 \dots E_n) = \frac{p(E_1 | H_i) \times p(E_2 | H_i) \times \dots \times p(E_n | H_i) \times p(H_i)}{\sum_{k=1}^m p(E_1 | H_k) \times p(E_2 | H_k) \times \dots \times p(E_n | H_k) \times p(H_k)},$$

$$i = \overline{1, m}.$$

Вместе с тем предположения о независимости событий в ряде случаев подавляют точность суждений и свидетельств в ЭС.

### Контрольные вопросы:

1. Поясните, что означает недетерминированность выводов.
2. Роль нечеткости в ЭС.
3. Основные типы нечеткости в ЭС.
4. Сформулируйте формулу Байеса.

## Тема 4. НЕЙРОННЫЕ СЕТИ

Вводятся понятие нейронной сети, принципы ее организации, типы и процесс функционирования.

### 4.1. Понятие нейронной сети

*Нейронные вычисления* – это теория разработки и исследования систем обработки информации, использующих механизмы восприятия и переработки информации естественных информационных систем – мозга человека и животных.

Исследования нейрофизиологов показали, что основным функциональным элементом головного мозга является нейрон. Мозг состоит из приблизительно  $10^{11}$  нейронов. Нейрон - специализированная клетка, которая способна к передаче, обработке и хранению информации, необходимой для реализации физиологических функций организма. Биологический нейрон состоит из тела и входных и выходных каналов - дендритов и аксонов.

Для хранения информации в нейронах имеются так называемые синапсы. Эти образования представляют собой в большинстве случаев однонаправленные «ворота» между аксонами пресинаптических нейронов и дендритами постсинаптического нейрона. Их основное назначение заключается в накоплении и хранении информации в течение всего жизненного цикла организма. Основой всех функций *памяти* мозга является именно механизм деятельности синапсов, количество которых в головном мозге человека составляет около  $10^{14}$ – $10^{15}$ .

Еще одна особенность биологических нейронных сетей — естественный параллелизм и коллективизм их работы.

*Обучение* естественных нейронных систем происходит путем изменения силы межнейронных связей.

Возможности биологических нейронных сетей обусловлены следующими их свойствами:

- 1) Параллельность обработки информации: параллельное функционирование групп нейронов, и распределенную организацию памяти.

2) Способность к *ассоциированию, классификации, обобщению, абстрагированию* и множество других.

3) Способность к *самоорганизации*. В процессе работы биологические нейронные сети самостоятельно, под воздействием внешней среды, обучаются решению разнообразных задач.

4) *Надежность*. Биологические нейронные сети обладают фантастической надежностью: выход из строя даже 10% нейронов в нервной системе не прерывает ее работы.

Нейронные вычисления можно рассматривать как перспективную альтернативу программируемым вычислениям.

### **Задачи, решаемые с помощью нейронных вычислений**

*Классификация образов*. Задача состоит в указании принадлежности входного образа (например, речевого сигнала или рукописного символа), представленного вектором признаков, одному или нескольким предварительно определенным классам. Для решения задачи используется обучение *с учителем* – сети предьявляется обучающая выборка, состоящая из пар (*входной образ, метка класса*), и применяются определенные правила для того, чтобы научить сеть определять метку класса для входного образа, который на этапе функционирования может и не входить в обучающую выборку.

*Кластеризация/категоризация*. При решении задачи кластеризации, (классификация образов *без учителя*), отсутствует обучающая выборка с метками классов. Алгоритм кластеризации основан на подобии образов и размещает близкие образы в один кластер.

*Аппроксимация функций*. Необходима при решении многочисленных инженерных и научных задач моделирования.

*Предсказание/прогноз*. Пусть заданы значения дискретных отсчетов в последовательные моменты времени. Задача состоит в предсказании значения в некоторый будущий момент времени.

*Оптимизация*. Задачей алгоритма оптимизации является отыскание такого решения, которое удовлетворяет системе ограничений и максимизирует или минимизирует целевую функцию.

*Память, адресуемая по содержанию*. В модели вычислений фон Неймана обращение к памяти доступно только с помощью адреса, который не зависит от содержания памяти. Ассоциатив-



ная память, или память, адресуемая по содержанию, доступна по указанию заданного содержания. Содержимое памяти может быть вызвано даже по частичным данным или искаженному содержанию.

## 4.2. Принципы организации и функционирования ИНС

*Нейронная сеть* – это параллельная распределенная структура обработки информации в виде направленного графа с учетом следующих определений и ограничений:

1) вершинами графа являются обрабатывающие элементы; обрабатывающий элемент ИНС - *искусственные нейроны* (ИН);

2) дуги графа - *связи*; каждая функционирует как однонаправленный канал непрерывной передачи сигнала; математический тип данных, передаваемых по связям, может быть любым;

3) каждый обрабатывающий элемент может принимать любое количество входных связей (входов);

4) каждый обрабатывающий элемент может иметь любое количество выходных связей (выходов), но по ним должен передаваться один и тот же сигнал (выходной сигнал);

5) обрабатывающий элемент имеет локальную память; содержимое локальной памяти характеризует связь и называется *весовым коэффициентом* связи; выходной сигнал обрабатывающего элемента определяется как значение функции, называемой *функцией активации*, аргументами которой являются весовые коэффициенты и входные сигналы;

б) входные сигналы нейронной сети поступают по входным связям, начинающимся за ее пределами, а выходные сигналы – оканчиваются за пределами сети.

Как правило, во всех известных ИНС множество ИН разделено на подмножества, которые называются *слоями*. ИН могут быть связаны как с ИН своего слоя, так и с ИН других слоев.

Многие модели ИНС включают в себя слой, называемый входным. Каждый ИН входного слоя принимает только внешние по отношению к ИНС входные сигналы и, как правило, его функция заключается только в передаче сигналов на следующий слой сети.

Функцией нейрона является также реализация процесса обучения сети, который заключается в изменении значений весовых коэффициентов в соответствии с изменяющимися условиями функционирования.

Схема искусственного нейрона представлена на рис. 4.1.

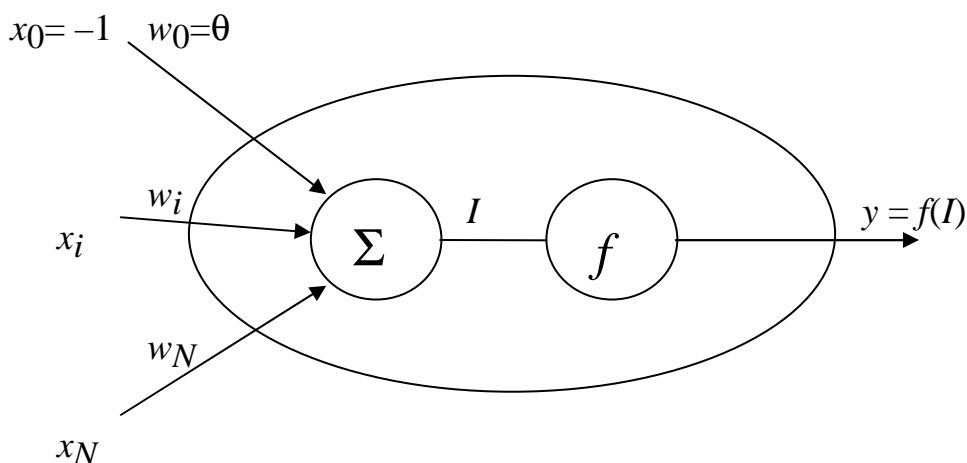


Рис. 4.1. Искусственный нейрон

Каждый из  $N + 1$  входных сигналов  $x_i$  умножается на соответствующий весовой коэффициент  $w_i$  и поступает на суммирующий блок, обозначенный  $\Sigma$ . Множество весовых коэффициентов входных связей нейрона обозначается  $W$  и представляет собой вектор-столбец  $W = [w_0, w_1, \dots, w_N]^T$ .

Суммирующий блок, соответствующий телу биологического элемента, складывает взвешенные входы алгебраически:

$$I = \sum_{i=0}^N w_i x_i = W^T X.$$

Значение  $I$  называется *внутренним выходом* ИН, и преобразуется функцией активации  $f$ , в результате чего получается выходной сигнал ИН:  $y = f(I)$ .

Значения параметров  $x_0$  и  $w_0$  фиксированы. Весовой коэффициент  $w_0$  принимает значение  $\theta$ , которое называется *пороговым значением* функции активации.

Существует много разновидностей функций активации, но наиболее удобна в использовании нелинейная зависимость, называемая логистической функцией и определяемая как:

$$f(x) = \frac{1}{1 + e^{-ax}},$$

где  $a$  – параметр, определяющий свойства функции.

### Классификация законов и способов обучения

Существует два основных класса методов обучения ИНС:

1. Обучение с учителем;
2. Самообучение.

### Архитектуры искусственных нейронных сетей

На сегодняшний день можно выделить четыре основные разновидности архитектуры ИНС.

1. *Однослойные прямонаправленные сети.*  $K$ -слоистой называется ИНС, состоящая из  $k$  групп нейронов, разделенных по слоям. Если сигналы в сети распространяются только по направлению из начала в конец, то такая ИНС называется *прямонаправленной*. На рис. 4.2 однослойная прямонаправленная ИНС.

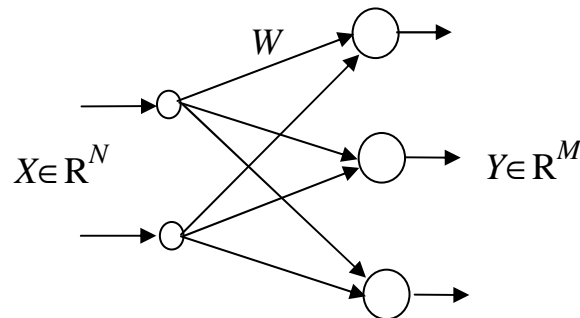


Рис. 4.2. Прямонаправленная сеть с одним слоем нейронов

Она включает в себя слой входных нейронов и слой выходных. Нейроны входного слоя просто передают сигналы на выходной слой, не преобразуя их. В выходном происходит преобразование сигналов и формирование выхода сети. Матрица  $W$  имеет размерность  $M \times N$ . Сеть функционирует по правилу:  $Y = f(WX)$ ;

$$y_i = f\left(\sum_{j=1}^N w_{ij} x_j\right), \quad \text{где } i = 1, \dots, M.$$

2. *Многослойные прямонаправленные сети.* Они характеризуются наличием одного или нескольких промежуточных (скрытых) слоев, осуществляющих преобразование информации. Ней-

роны скрытого слоя называются скрытыми. В скрытых нейронах последовательно, слой за слоем, происходит нелинейное преобразование сигналов. Сигналы с последнего скрытого слоя поступают на нейроны выходного слоя, которые формируют выход сети. Структура двухслойной сети представлена на рис. 4.3.

Сеть функционирует по правилу:  $Z = f(VX)$ ;  
 $Y = f(WZ) = f(W f(VX))$ ;  $y_i = f(\sum_{l=1}^L w_{il} f(\sum_{j=1}^N v_{lj} x_j))$ , где  $i = 1, \dots, M$ .

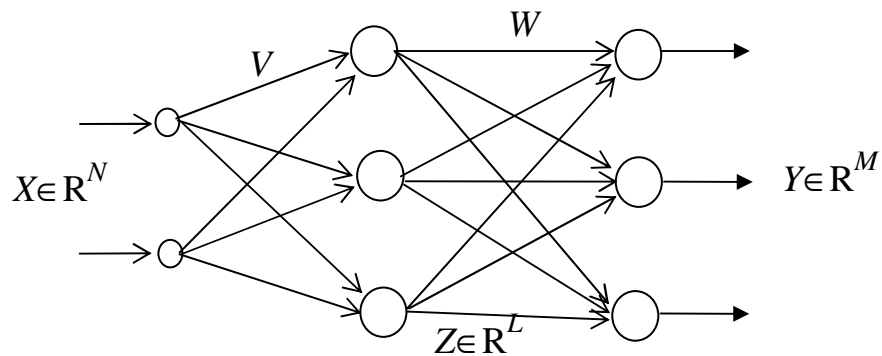


Рис. 4.3. Двухслойная сеть

3. *Рекуррентные сети.* Этот тип ИНС отличается существованием обратных связей и элементов временной задержки сигнала. Подробно рассмотрим, как функционирует обратная связь.

На рис. 4.4 представлена структура простейшей системы с обратной связью. Здесь входной сигнал  $X(t)$ , внутренний сигнал  $X'(t)$  и выходной сигнал  $y(t)$  – функции переменной  $t$  – дискретных отсчетов времени.

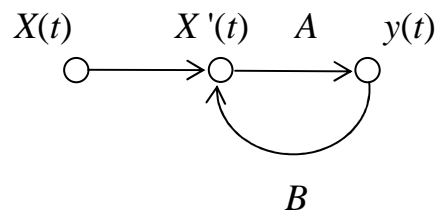


Рис. 4.4. Рекуррентная сеть

Если предположить, что прямой канал передачи данных характеризуется некоторым оператором  $A$ , а обратный –  $B$ , то  $y(t) = A[X'(t)]$ , а  $X'(t) = X(t) + B[y(t)]$ . Тогда,

$$y(t) = \frac{A}{1 - AB} [X(t)].$$

Наиболее простым случаем рекуррентной сети является один слой нейронов, охваченный обратными связями. При этом каждый нейрон получает выходные сигналы всех остальных нейронов с некоторой временной задержкой.

4. *Полносвязные сети.* Характерным признаком ИНС этого типа является наличие связей между всеми нейронами. Полносвязные сети применяются при решении задач классификации и распознавания образов.

### **Простой и однослойный персептроны**

*Персептрон* – это ИНС, относящаяся к классу однослойных прямонаправленных сетей, используемая для классификации линейно разделимых образов. Образы представлены векторами признаков в  $N$ -мерном пространстве. *Простой персептрон* – это элементарная сеть, состоящая из единственного персептронного нейрона. *Персептронный нейрон* выполняет:

1) вычисление внутреннего выхода как линейной комбинации входов и весовых коэффициентов, соответствующих этим входам;  $I = \sum_{i=1}^N w_i x_i - \theta$ , где  $N$  – количество входов;

2) вычисление выходного сигнала с помощью пороговой функции активации  $y = \text{sign}(I)$  или  $y = 0.5(\text{sign}(I) + 1)$ .

Персептрон, состоящий из нескольких элементов, изображен на рис. 4.5. Персептрон из  $M$  нейронов способен разделять  $2^M$  классов образов. На его вход поступает вектор входных сигналов  $X = [x_1, \dots, x_N]$ , а на выходе получается вектор выходных сигналов  $Y = [y_1, \dots, y_M]$ . Матрица  $W = \{w_{ji} \mid i = 1, 2, \dots, N, j = 1, 2, \dots, M\}$ , где  $w_{ji}$  весовой коэффициент  $i$ -го входа  $j$ -го персептронного нейрона, называется весовой матрицей сети, которая определяет весовые коэффициенты входов каждого персептронного нейрона.

### **Классификация линейно разделимых образов**

Целью решения задачи классификации является разделение набора образов  $x_1, x_2, \dots, x_p$  на два класса  $S_1$  и  $S_2$ . Она предполагает наличие сети с определенными параметрами, которая при

предъявлении образа  $x_i$ , относящегося к классу  $S1$ , выдает на выходе 1, а если образ относится к классу  $S2$  – выдает  $-1$  (или 0).

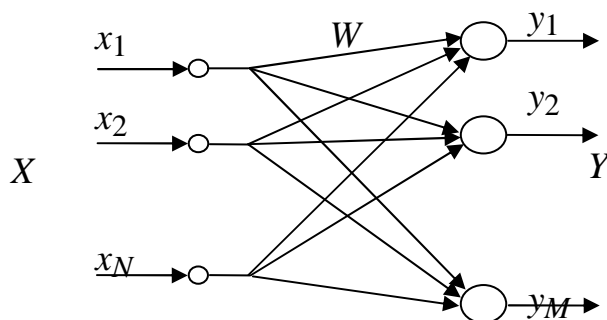


Рис. 4.5. Персептрон, состоящий из  $M$  нейронов с  $N$  входами

В общем случае задача решается в многомерном пространстве образов. Образом является точка, координаты которой могут представлять собой коды атрибутов или свойств объектов, подлежащих классификации. На рис. 4.6 представлена графическая интерпретация решения задачи для двухмерного пространства.

В этом случае два класса объектов могут быть разделены прямой, причем точки, расположенные ниже прямой, соответствуют объектам класса  $S1$ , а точки, расположенные выше – объектам класса  $S2$ .

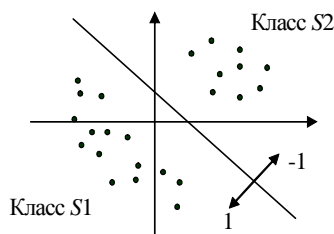


Рис. 4.6. Графическая интерпретация решения задачи для двухмерного пространства

В многомерном варианте два класса могут быть разделены простой линейной гиперплоскостью.

Образы, которые могут быть разделены гиперплоскостью в  $N$ -мерном пространстве признаков так, чтобы все образы, принадлежащие одному классу, находились с одной стороны гиперплоскости, а образы, принадлежащие другому классу, с другой стороны гиперплоскости, называются **линейно разделимыми**.

Очевидно, что требование линейной разделимости резко ограничивает возможности простых персептронов. Это ограничение можно преодолеть, добавив в сеть дополнительные слои.

### **Обучение персептрона**

Закон обучения персептрона относится к классу законов обучения «с учителем». Персептрон обучают, подавая на его вход в некоторый момент времени очередной образ из обучающего множества и значение желаемого выхода и подстраивая веса, пока для всех образов не будет достигнут желаемый выход.

#### Алгоритм обучения:

1. Инициализация. Начальные значения весовых коэффициентов обнуляются или случайно определяются как малые величины.

2. Активация. В момент времени  $t$  на вход сети подается входной обучающий вектор  $X(t)$  и значение желаемого выхода  $d(t)$ .

3. Вычисление действительного выхода. Вычисляется внутренний выход  $I$  каждого нейрона и значение пороговой функции активации каждого нейрона.

4. Модификация весовых коэффициентов. Каждый вес модифицируется в соответствии с правилом обучения.

5. Приращение времени на единицу и повторение шагов со второго по пятый до тех пор, пока разница между желаемым и действительным выходами не станет достаточно малой.

### **4.3. Рекуррентные ассоциативные сети**

Рекуррентные ассоциативные сети являются подмножеством ассоциативных, предназначенных для решения широкого круга задач ассоциативной памяти. Используются в основном для решения задач автоассоциативной памяти - для восстановления изображений по их части или распознавания изображений, искаженных шумом. Сеть хранит в памяти множество образов, закодированных двоичными векторами в пространстве признаков, и выдает один из этих образов в ответ на внешнее воздействие.

Эта способность обусловлена особенностью функционирования рекуррентных ассоциативных сетей. Все рекуррентные се-

ти берут начало в некотором начальном состоянии и затем в процессе функционирования сходятся к одному из множества стабильных состояний, соответствующих минимуму некоторой характеристической функции сети, называемой также энергетической.

### Энергетическая функция рекуррентной сети

Рассмотрим ИНС, содержащую  $N$  ИН, полностью связанных между собой симметричными связями. Пусть состояние нейрона  $i$  в момент времени  $t$  определяется выходным сигналом  $s_i(t)$ . Для такой сети можно определить энергетическую функцию:

$$E = - \sum_{i=1}^N \sum_{j=1}^N w_{ij} s_i s_j + 2 \sum_{i=1}^N \theta_i s_i .$$

Энергетическая функция всегда убывает при изменении состояния нейрона. Можно показать также, что  $E$  имеет *глобальный* минимум, однако, не всегда можно достичь его. Тем не менее, в теории рекуррентных сетей доказано, что сеть всегда приходит к некоторому *локальному* минимуму за конечное время.

### Сеть Хопфилда

Наиболее известной рекуррентной автоассоциативной сетью является сеть Хопфилда. Это однослойная полносвязная рекуррентная сеть, элементами которой являются искусственные нейроны с логистической функцией активации.

Для сети из  $N$  нейронов состояние сети определяется вектором  $S = [s_1, s_2, \dots, s_N]^T$ , где  $s_i = \pm 1$  – переменные состояния сети.

Предполагается, что наклон логистической функции  $f_j(\cdot)$  в области начала координат бесконечно велик, т.е.  $f_j(\cdot)$  приближается к пороговой функции. Это можно обеспечить выбором достаточно большого значения параметра  $a$ .

Искусственный нейрон  $j$  сети Хопфилда обновляет свое состояние следующим образом.

Вычисляется:

- 1) внутренний выход нейрона  $I_j = \sum_{i=1}^N w_{ji} s_i - \theta_j$ ;

- 2) новое значение активации для нейрона  $j$ :



$$s_j(t+1) = f(I_j) = \begin{cases} 1, & \text{если } I_j > 0 \\ s_j(t), & \text{если } I_j = 0 \\ -1, & \text{если } I_j < 0 \end{cases}$$

В начальный момент времени состояния нейронов  $s_j(0)$  принимают значения внешних входов.

Матрица весовых коэффициентов формируется в процессе обучения с помощью правила Хебба следующим образом. Пусть необходимо хранить в памяти сети набор из  $p$   $N$ -мерных векторов  $\{\xi_\mu \mid \mu = 1, 2, \dots, p\}$ . Эти  $p$  векторов составляют *общую память* сети.

Пусть  $\xi_{\mu,i}$  –  $i$ -й элемент общей памяти. Элементы весовой матрицы определяют как  $w_{ji} = \frac{1}{N} \sum_{\mu=1}^p \xi_{\mu,j} \xi_{\mu,i}$ .

Обычно  $w_{ii} = 0$ , т.е. нейроны сети не связаны сами с собой.

Тогда  $W = \frac{1}{N} \sum_{\mu=1}^p \xi_\mu \xi_\mu^T - \frac{p}{N} I$ ,  $I$  – единичная матрица. Процесс обу-

чения называется также фазой запоминания.

Функционирование сети называют фазой восстановления. На фазе восстановления сети предьявляется образ, представляющий собой зашумленный или неполный вариант образа из общей памяти сети. В результате функционирования сеть приходит к некоторому стабильному состоянию, которое является выходом. Состояние стабильности характеризуется свойством  $s_i(t+1) = f(Ws_i(t))$  для всех нейронов.

### **Двунаправленная ассоциативная память**

Двунаправленная ассоциативная память (ДАП) является модификацией сети Хопфилда и предназначена для решения задач гетероассоциативной памяти. Это рекуррентная и двухслойная сеть. ДАП способна к обработке неполных и зашумленных данных. На рис. 4.7 приведена базовая конфигурация ДАП. Эта сеть ассоциирует образы  $A$  и  $B$ . Образ  $A$  подается на вход сети, а  $B$  получается на выходе.

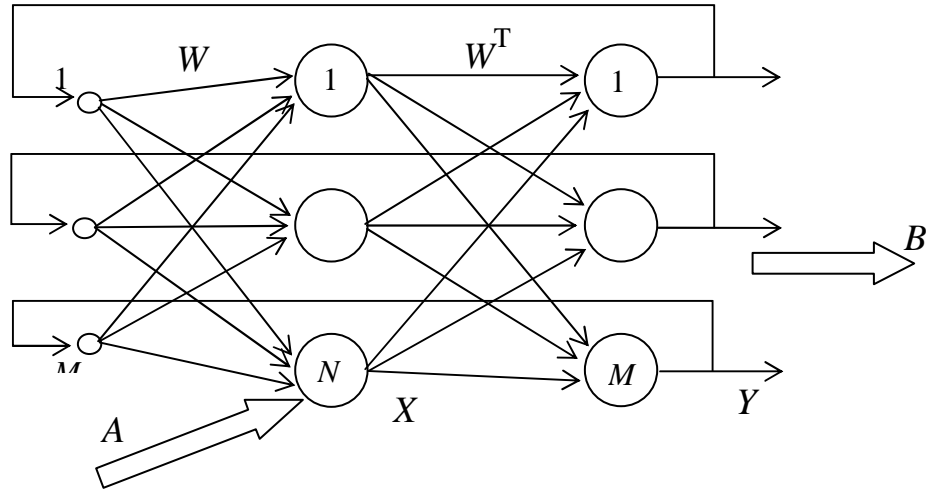


Рис. 4.7. Базовая конфигурация ДАП

ДАП состоит из двух слоев нейронов, первый из  $N$  элементов имеет выходной сигнал  $X \in \{-1, 1\}^N$ , а  $M$  нейронов второго – выходной сигнал  $Y \in \{-1, 1\}^M$ . Весовые коэффициенты связей между вторым слоем, с выходами  $X$ , и первым слоем, с выходами  $Y$ , определяются матрицей  $W = \{w_{ji}\}, i = 1, 2, \dots, N; j = 1, 2, \dots, M$ .

Весовая матрица ДАП формируется обучением с помощью правила Хебба. Пусть общая память сети  $\xi$  содержит  $p$  пар ассоциируемых друг с другом образов, представленных векторами в пространстве признаков:  $\xi = \{(A_1, B_1), (A_2, B_2), \dots, (A_p, B_p)\}$ .

$$\text{Тогда } W = \sum_{\mu=1}^p B_{\mu} A_{\mu}^T.$$

Смена состояний нейронов первого и второго слоев происходит следующим образом:

$$x_i^{new} = \begin{cases} 1, & \text{если } v_i^{(1)} > S_i; \\ x_i^{old}, & \text{если } v_i^{(1)} = S_i; \\ -1, & \text{если } v_i^{(1)} < S_i; \end{cases} \quad y_j^{new} = \begin{cases} 1, & \text{если } v_j^{(2)} > T_j; \\ y_j^{old}, & \text{если } v_j^{(2)} = T_j; \\ -1, & \text{если } v_j^{(2)} < T_j. \end{cases}$$

где  $v_i^{(1)}$  и  $v_j^{(2)}$  – внутренние выходы нейронов первого и второго слоев соответственно, а  $S_i$  и  $T_j$  – пороговые значения функции активации для первого и второго слоев.

Внутренние выходы вычисляются следующим образом:

$$v_i^{(1)} = \sum_{j=1}^M w_{ij} y_j; \quad v_j^{(2)} = \sum_{i=1}^N w_{ji} x_i.$$

Таким образом,  $X^{new} = f(W^T Y^{old})$  и  $Y^{new} = f(WX^{old})$

Все нейроны слоя с выходным сигналом  $X$  обновляют свои состояния одновременно в один момент времени, все нейроны второго – также одновременно в следующий момент времени. Процесс продолжается, пока  $X$  и  $Y$  не перестанут изменяться.

### Машина Больцмана

Это модификация сети Хопфилда, в которой используется так называемое стохастическое функционирование, в результате чего сеть приобретает возможность в большем количестве случаев достигать глобального экстремума энергетической функции находить правильное решение. Машина Больцмана и сеть Хопфилда имеют следующие *общие характеристики*:

- предназначена для решения задач ассоциативной памяти;
- однослойная, рекуррентная;
- состояния принимают значения  $\pm 1$ ;
- весовая матрица межнейронных связей симметричная;
- нейроны не имеют обратных связей к самим себе, т.е. значения диагональных элементов весовой матрицы равны нулю;
- нейроны обновляют свое состояние по одному в единицу времени и выбираются случайно.

Они также имеют важное *отличие*: алгоритм обновления состояний нейронов машины Больцмана является вероятностным, а не детерминированным, как в сети Хопфилда, а сами нейроны называются *стохастическими*.

Стохастический нейрон  $j$  изменяет свое состояние  $s_j$  в зависимости от значения его внутреннего выхода  $v_j$  с вероятностью  $P(v_j)$ . Тогда состояние  $j$ -го нейрона определяется следующим об-

$$s_j = \begin{cases} 1 & \text{с вероятностью } P(v_j); \\ -1 & \text{с вероятностью } 1 - P(v_j). \end{cases}$$

разом:

Вероятность изменения состояния нейрона определяется с помощью функции

$$P(v) = \frac{1}{1 + \exp(-2v/T)},$$

где  $T$  – параметр, определяющий характер функционирования нейрона. Если  $T \rightarrow 0$ , то функция становится пороговой и правило активации стохастического нейрона сводится к правилу функционирования нейрона сети Хопфилда. Чем больше  $T$ , тем сильнее вероятностный характер функционирования стохастического нейрона. При  $T \rightarrow \infty$  вероятность смены состояния составляет  $1/2$ .

Функционирование нейронов можно объяснить и в терминах изменения энергии сети  $\Delta E$ .

Вероятность смены состояния  $j$ -го нейрона на некотором шаге процесса функционирования сети от состояния  $s_j$  к состоянию  $-s_j$  равна  $P(s_j \rightarrow -s_j) = \frac{1}{1 + \exp(-\Delta E_j / T)}$ ,

где  $\Delta E_j$  – изменение энергии в результате смены состояния.

Алгоритм функционирования машины Больцмана:

1. Вычисление весовых коэффициентов в соответствии с правилом Хебба;
2. Инициализация состояний  $s_j$  значениями тестового вектора;
3. Выбор начального значения параметра  $T = T_0$ ;
4. Обновление состояний нейронов сети по (3);
5. Возвращение к этапу 4. После  $L$  итераций (количество итераций определяют эмпирически) уменьшение параметра  $T$ ;
6. Возвращение к этапу 4 до тех пор, пока не будет найдено стабильное состояние, соответствующее минимуму энергетической функции.

Уменьшение значения параметра  $T$  предполагается медленным. Если это требование не соблюдать, то сеть игнорирует малые энергетические изменения и может не достичь глобального. Сходимость алгоритма машины Больцмана слишком медленна. На практике используют процедуру с ограничением по времени:

определяют  $T_0$ ;

правило изменения  $T$  определяют рекуррентно как  $T_i = \alpha T_{i-1}$ , где  $i = 1, 2, \dots$ , и  $0.8 < \alpha < 0.99$ ;

определяют конечное значение  $T_k$ .

## 4.4. Сеть с обратным распространением ошибки

### Задача преобразования данных

Сети преобразования данных применяются для большого круга задач: распознавания образов, управления, прогнозирования, идентификации сложных систем, обработки сигналов и т.д.

Все они сводятся к следующей математической постановке. Сеть преобразования данных реализует функцию  $f: A \subset \mathbb{R}^N \rightarrow \mathbb{R}^M$ , устанавливающую соответствие между ограниченным подмножеством  $A$   $N$ -мерного Евклидова пространства и ограниченным подмножеством  $f[A]$   $M$ -мерного Евклидова пространства путем обучения сети соответствию  $Y_l = f(X_l)$  на примерах  $(X_1, Y_1), \dots, (X_l, Y_l), \dots, (X_L, Y_L)$ . Можно определить среднеквадратическую ошибку сети:  $F(W) = \frac{1}{L} \frac{1}{2} \sum_{l=1}^L \sum_{j=1}^M (y_j^{(l)} - y_j'^{(l)})^2$ .

Обычно минимальное значение ошибки отлично от нуля, поскольку не может быть построена ИНС, абсолютно точно аппроксимирующая заданную функцию.

Требуется найти такую весовую матрицу  $W^*$ , чтобы при функционировании сети получалась бы наименьшая среднеквадратическая ошибка  $F(W^*) = F_{\min}$ .

### Закон обучения сети с обратным распространением ошибки

ИНС с обратным распространением ошибки – многослойная двунаправленная сеть, состоящая из  $K$  слоев нейронов. Входной слой состоит из  $N$  нейронов, принимающих значения компонентов  $x_i$  вектора  $X \in A$  и распределяющих их без изменения к следующему слою, причем каждый нейрон на каждом слое принимает выходной сигнал от каждого нейрона предыдущего слоя. Индекс  $(l)$  обозначает номер примера из обучающей выборки.

Выходной  $K$ -й слой сети состоит из  $M$  нейронов и выдает оценку  $Y' = \{y_j'\}$  желаемого выходного вектора  $Y \in f[A]$ ,  $Y = \{y_j\}$ .

Слои сети, которые располагаются между входным и выходным называются скрытыми.

В сети присутствуют два вида связей – прямые и обратные. Прямые связи служат для передачи сигналов от входного слоя вперед по сети, в результате чего сеть вырабатывает выходной сигнал. Обратные связи служат для передачи ошибки от выходного слоя сети ко второму слою через все скрытые слои, в результате чего реализуется обучение сети. Передача данных по прямым связям называется *прямым проходом* алгоритма обучения, а по обратным – *обратным проходом*.

Пусть  $S_{kj}$  – выходной сигнал  $j$ -го нейрона  $k$ -го слоя,  $M_k$  – количество нейронов  $k$ -го слоя. *Солнцем* называется нейрон, выполняющий суммирование данных, приходящих по входным связям и вычисляющий значение функции активации от полученной суммы. Результат передается планетам следующего слоя. *Планетой* назовем элемент сети с обратным распространением ошибки, имеющий единственный вход и единственный выход, и вычисляющий произведение входного данного на значение весового коэффициента, который хранится в его памяти.

Закон обучения для ИНС с обратным распространением ошибки является обобщением закона Уидроу, который может использоваться для многослойных сетей. Его основной недостаток – во многих случаях для сходимости может потребоваться многократное предъявление всей обучающей выборки.

### **Проблемы обучения сетей преобразования данных**

В процессе обучения весовые коэффициенты сети преобразования данных настраиваются так, чтобы она могла воспроизводить выходные данные, содержащиеся в предъявляемых ей обучающих парах. В результате процесса обучения сеть запоминает все обучающие данные, то есть правильно реагирует на тесты, совпадающие с обучающими данными. Однако кроме способности помнить данные обучающей выборки, сеть должна обладать способностью к обобщению.

*Обобщением* называется способность сети корректно осуществлять преобразование вход/выход как для примеров из обучающего множества, так и для данных, не использованных при обучении. Таким образом, обобщение – это способность ИНС к нелинейной интерполяции входных данных.

На рис. 4.8 проиллюстрировано свойство обобщения. Кривая, изображенная на рисунке, отображает нелинейное преобразование, осуществляемое некоторой ИНС. Обучающие и тестовые данные представлены точками на координатной плоскости. Абсциссы точек, представляющих обучающие данные, – это входы сети, а ординаты – желаемые выходы. Для точек, представляющих тестовые данные, абсциссы – это входы, а ординаты – отклики сети. В ситуации, изображенной на рис. 4.8, что сеть обладает *хорошей* способностью к обобщению.

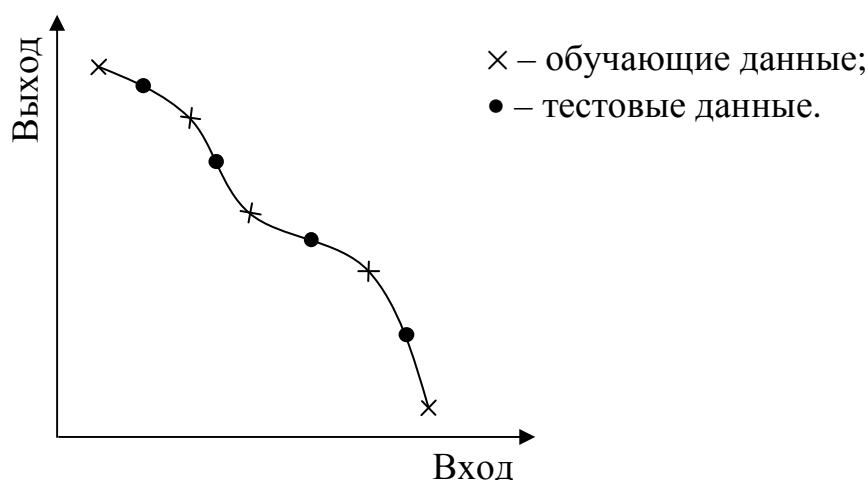


Рис. 4.8. Иллюстрация свойства обобщения

Однако ИНС не всегда в результате обучения приобретает хорошую способность к обобщению. На рис. 4.9 изображена реакция некоторой сети, которую обучали осуществлять то же преобразование. Видно, что в этом случае сеть не может корректно интерполировать входные данные, которые не входят в обучающий набор.

Отсутствие хорошей способности к обобщению может объясняться, например, явлением *переобучения*.

При обучении сети минимизируется ошибка между действительным выходом сети и желаемым выходом из обучающего набора. Но для того, чтобы сеть обладала хорошей способностью к обобщению, важно не только и не столько отсутствие ошибки на этапе обучения, сколько правильное функционирование сети на примерах, не входящих в обучающую выборку.

Может возникнуть и другая ситуация – ошибка верификации начинает расти, а ошибка обучения еще не достигла прием-

лемого уровня, т.е. сеть еще не приобрела способность воспроизводить заданную функцию. В этом случае сеть называют *недообученной*. Обычно эффект недообучения возникает, когда сеть неудачно спроектирована, т.е. неправильно выбраны архитектура и параметры сети.

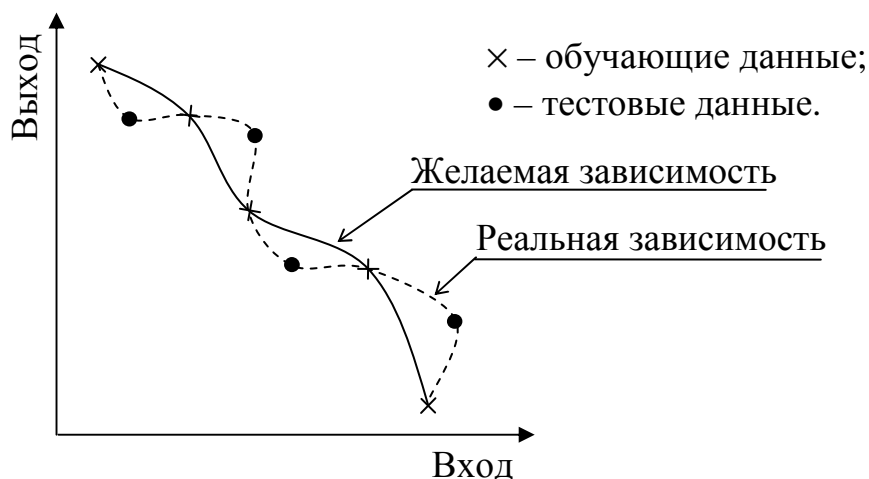


Рис. 4.9. Иллюстрация свойства обобщения. Продолжение

На рис. 4.10 показана ситуация, когда ошибка обучения еще продолжает уменьшаться, а ошибка тестирования уже начинает расти. Для того, чтобы выявить момент, в который надо остановить обучение (на рис. 4.10 он помечен «СТОП!»), применяют так называемую процедуру кросс-верификации. Данные обучающей выборки разбивают на два подмножества. Одно используют непосредственно для обучения, а другое для определения ошибки тестирования, т.е. для проверки или верификации. Когда ошибка верификации начинает расти, обучение прекращают.

Можно выделить два необходимых условия для существования способности сети к обобщению:

1) гладкость нелинейного преобразования, выполняемого сетью, т.е. небольшим приращениям входных данных должны соответствовать небольшие приращения выходных;

2) набор обучающих данных должен быть достаточно представительным, т.е., с одной стороны, время его обработки не должно быть слишком большим, а с другой стороны, он должен полностью учитывать все особенности решаемой задачи преобразования данных.



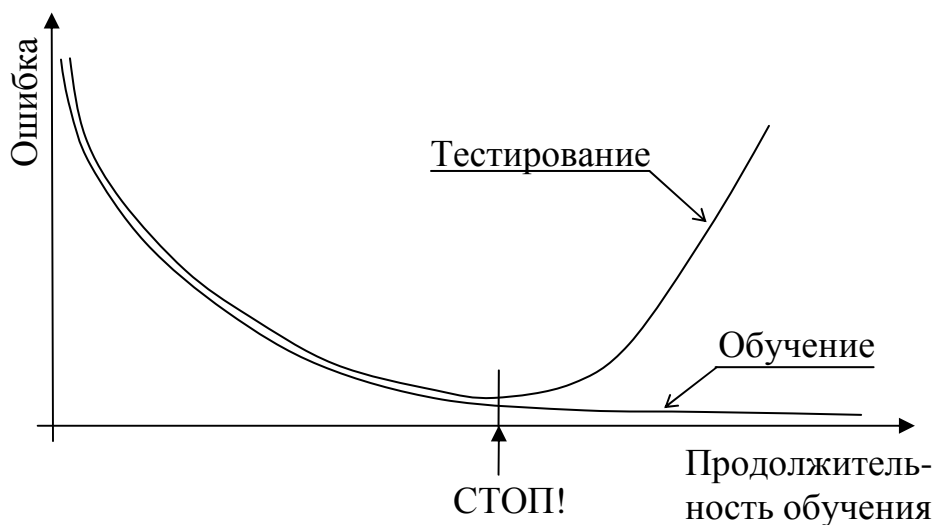


Рис. 4.10. Иллюстрация свойства обобщения. Окончание

### Применение многослойного персептрона для пространственно-временной обработки данных

Многослойная сеть, обучающаяся по правилу обратного распространения ошибки, сегодня является одним из самых популярных методов решения задач с помощью ИНС. Тем не менее, он существенно ограничен в своих возможностях, т.к. сеть обучается и функционирует на статических данных.

Многослойный персептрон может быть использован и для решения другого класса задач – задач нелинейного прогноза на стационарных временных последовательностях. В этом случае данные, содержащиеся в последовательности и подлежащие обработке, должны подаваться на вход сети.

На вход подается временная последовательность  $X = [x(t-1), x(t-2), \dots, x(t-p)]^T$  по одному элементу в каждый момент времени с задержкой  $z^{-1}$ .

Задача состоит в прогнозировании одного значения временной последовательности в некоторый  $t$ -й момент времени с использованием данных о нескольких  $p$  значениях той же временной последовательности, полученных в предыдущие  $t-1, \dots, t-p$  моменты времени.

К сожалению, круг задач, требующих представления исходных данных в виде временной последовательности и позволяю-

щих в этом случае использовать статическую архитектуру многослойного персептрона, ограничен. Основой многослойной сети с динамическими параметрами является нейрон с динамическим характером функционирования, использующий в системе связей задержки во времени. Задержки во времени обычно моделируются с помощью линейных стационарных фильтров.

Фильтры, моделирующие функционирование связей имеют следующие характеристики: формируют отклики только после того, как на вход будет подан сигнал; импульсная характеристика фильтра выполняет функции памяти и поэтому конечна.

Многослойная нейронная сеть, состоящая из нейронов с динамическим функционированием, способна решать задачи пространственно-временной обработки данных.

Существуют два подхода для обучения динамической сети.

Подход 1. Преобразование динамической сети в статическую и обучение статической сети с помощью алгоритма с обратным распространением ошибки. Теоретически, для этой сети возможно применение стандартного алгоритма с обратным распространением ошибки. Однако на практике такой способ оказывается слишком трудоемким.

Подход 2. Для обучения сети используется алгоритм обучения многослойного персептрона – обучение с обратным распространением ошибки. Единственным различием является то, что для динамической сети необходимо переопределить входные данные и параметры алгоритма.

#### **4.5. Сеть Кохонена**

Обучение с учителем используется при решении задач преобразования данных в тех случаях, когда есть возможность составить представительную обучающую выборку, содержащую входные данные и желаемые результаты, которые должны появляться на выходе сети в ответ на предъявление ей этих входных данных. Однако такая возможность есть не всегда. Гораздо чаще бывает, что множество пар не может быть использовано для обучения методом «с учителем», поскольку не является представительным и может возникнуть эффект недообучения сети. Тогда

для решения задачи применяют обучение методом самоорганизации, а для реализации этого метода – сеть Кохонена.

Суть метода самоорганизации в том, что сеть, руководствуясь определенными правилами функционирования, определяет некоторую меру сходства между предъявляемыми ей данными и, на основании этого сходства, относит их к той или иной группе.

Самоорганизующиеся сети могут применяться для широкого круга задач преобразования данных, однако, наиболее эффективным является их использование при решении задач визуализации многомерных данных в пространстве меньшей размерности и кластеризации данных.

Решение этих задач предполагает отображение зависимостей между элементами многомерных данных в простые топологические отношения в пространстве меньшей размерности, обычно двумерном или одномерном. Весовые коэффициенты сети при этом настраиваются так, чтобы близко расположенные друг к другу нейроны выходного слоя активизировались при подаче на вход сходных в некотором смысле входных данных.

Поскольку нейроны выходного слоя сети обычно объединяются в двумерные прямоугольные или гексагональные решетки, сети Кохонена называют также картами Кохонена.

### **Обучение и функционирование сети Кохонена**

Пример структуры сети Кохонена представлен на рис. 4.11.

На вход данной сети поступает последовательность из  $p$  входных образов, представленных векторами  $X = [x_1 \ x_2 \ \dots \ x_N]^T$  в  $N$ -мерном пространстве признаков. Входные образы предъявляются сети по одному в единицу времени.

Выходной слой организован в виде прямоугольной решетки нейронов, состоящей из  $M$  строк и  $K$  столбцов. Предполагается, что нейроны выходного слоя соединены между собой горизонтальными связями, которые на рисунке не представлены. Об этих связях и их роли в обучении сети будет подробно рассказано позже.

Весовые коэффициенты связей между нейронами входного и выходного слоев – это матрица из  $MK$  векторов  $W_j = [w_{j1} \ w_{j2} \ \dots \ w_{jN}]^T$ , где  $j$  – номер нейрона выходного слоя,  $j = 1, 2, \dots, MK$ .

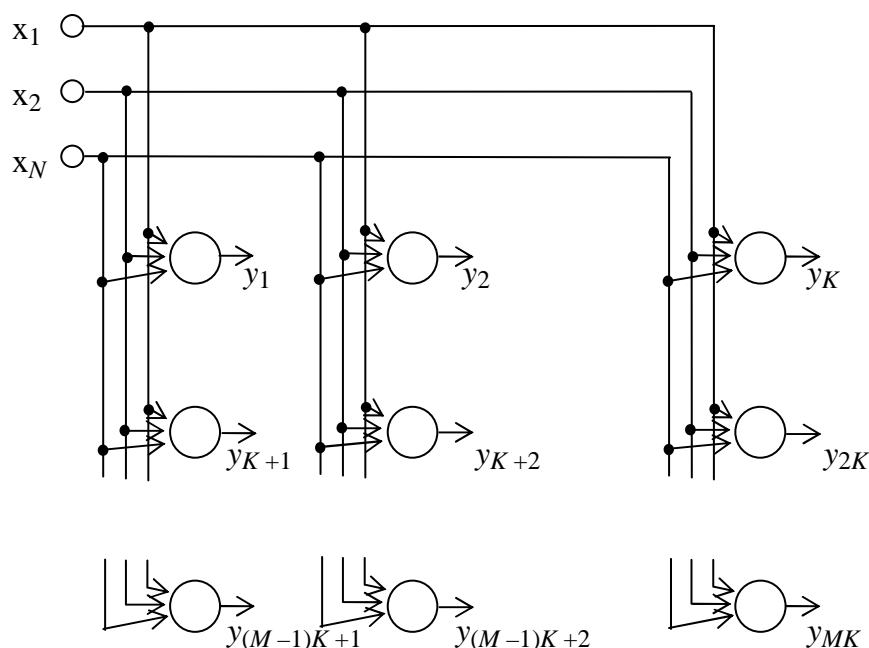


Рис. 4.11. Пример структуры сети Кохонена

Функционирование и обучение сети Кохонена связано с так называемой *процедурой конкуренции*. Основная идея конкуренции состоит в том, что при подаче на вход некоторого образа из входного множества, с помощью специальной процедуры конкуренции определяется единственный нейрон выходного слоя сети, который принимает значение, отличное от нуля, т.е. активизируется. Такой нейрон называют *нейроном-победителем*.

Далее, если сеть еще не обучена, происходит модификация весовых коэффициентов сети, причем изменению подлежат не все весовые векторы сети, а только весовые векторы нейрона-победителя и, возможно, его соседей, которые определяются специальным образом.

Если сеть уже обучена, то положение нейрона-победителя в выходном слое интерпретируется как решение в зависимости от конкретной задачи.

### **Модификация весовых коэффициентов**

Модификация весовых коэффициентов нейрона-победителя осуществляется по закону Хебба. Как известно, при обучении с помощью закона Хебба возможны «взрывы» сети, поэтому предлагается изменить его, добавив так называемый фактор забыва-

ния. Фактор забывания ограничивает рост значений весовых коэффициентов и снижает вероятность «взрыва».

### **Процедура конкуренции**

Сущность процедуры конкуренции – поиск нейрона-победителя – можно определить в терминах расстояния между векторами. Нейроном-победителем является тот нейрон, весовой вектор которого наиболее близок в смысле расстояния по Евклиду к вектору данных, подаваемых на вход сети.

Если  $i(X)$  – номер нейрона, весовой вектор которого наиболее близок к вектору входного сигнала  $X$ , т.е.  $i(X) = \arg \min_j \|X - W_j\|$ , где  $j$  – индекс нейрона выходного слоя, то нейрон  $i(X)$  называется нейроном-победителем для входного вектора  $X$ .

### **Алгоритм обучения сети Кохонена**

1. Инициализация весовых коэффициентов сети различными случайными значениями  $W_j(0)$ , где  $j$  – индексы нейронов выходного слоя сети Кохонена;
2. Предъявление сети очередного входного образа  $X(t)$ ;
3. Поиск нейрона-победителя в процедуре конкуренции нейронов выходного слоя. Нейрон-победитель  $i(X)$  определяется как  $\arg \min_j \|X(t) - W_j\|$ , где  $j$  – индекс нейрона выходного слоя;
4. Определение радиуса соседства  $\Lambda_{i(X)}$  нейрона-победителя;
5. Обновление весов связей нейронов, принадлежащих радиусу соседства нейрона-победителя:
6.  $W_j(t+1) = W_j(t) + \eta(t)\Lambda_{i(X)}(t)(X(t) - W_j(t))$ ;
7. Повторение с п.2, пока изменяются активации выходных нейронов.

### **Контрольные вопросы:**

1. Сформулировать задачи, решаемые ИНС.
2. Дать понятие персептрона.
3. Определить двунаправленную ассоциативную память.
4. Определить область применения сети Кохонена.

## Тема 5. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ

Рассматриваются вопросы интеллектуального анализа данных (ИАД) и их место в экономических ИС. Даются понятия OLTP, хранилища и витрины данных, OLAP и Data Mining.

### **5.1. Аналитические технологии и аналитические информационные системы**

К аналитическим технологиям, или технологиям аналитического анализа данных, относят подкласс информационных технологий, ориентированных на задачи прогнозирования состояния сложных динамических систем в нестационарных и неоднородных средах, разработку сценариев развития ситуаций в условиях комплексной динамической неопределенности, ситуационный анализ текущей обстановки и т.п. Постоянно возрастающий интерес к аналитическим технологиям, рассматриваемым в контексте систем управления, прежде всего, обусловлен тем, что важнейшей составной частью любого процесса управления являются действия по сбору и анализу информации, содержание которых составляют оценка текущей ситуации и прогноз ее развития.

Под ситуацией в данном случае понимается совокупность характеристик объекта управления и среды его функционирования в некоторый момент времени. Оценка ситуации подразумевает вычисление значений этих количественных и качественных характеристик (как правило, с их объединением и обобщением) и их критериальное соотнесение с требованиями (рис. 5.1).

Популярность аналитических технологий обусловлена влиянием трех основных групп факторов.

Во-первых, при реализации современных систем управления и их компонентов необходимо учитывать следующее:

прогресс развития любого объекта управления в абсолютном большинстве случаев связан с его усложнением, а следовательно – с увеличением исследуемых характеристик и в целом начального информационного базиса;

динамичный характер, как объектов управления, так и среды их функционирования, обусловленный множеством явных и

скрытых связей в общем случае материального, информационного и энергетического характера, что находит отражение в необходимости хранения временных рядов, характеризующих наблюдаемые ситуации, и способствует резкому росту информационных потребностей в оценивании и прогнозировании динамики этих ситуаций;

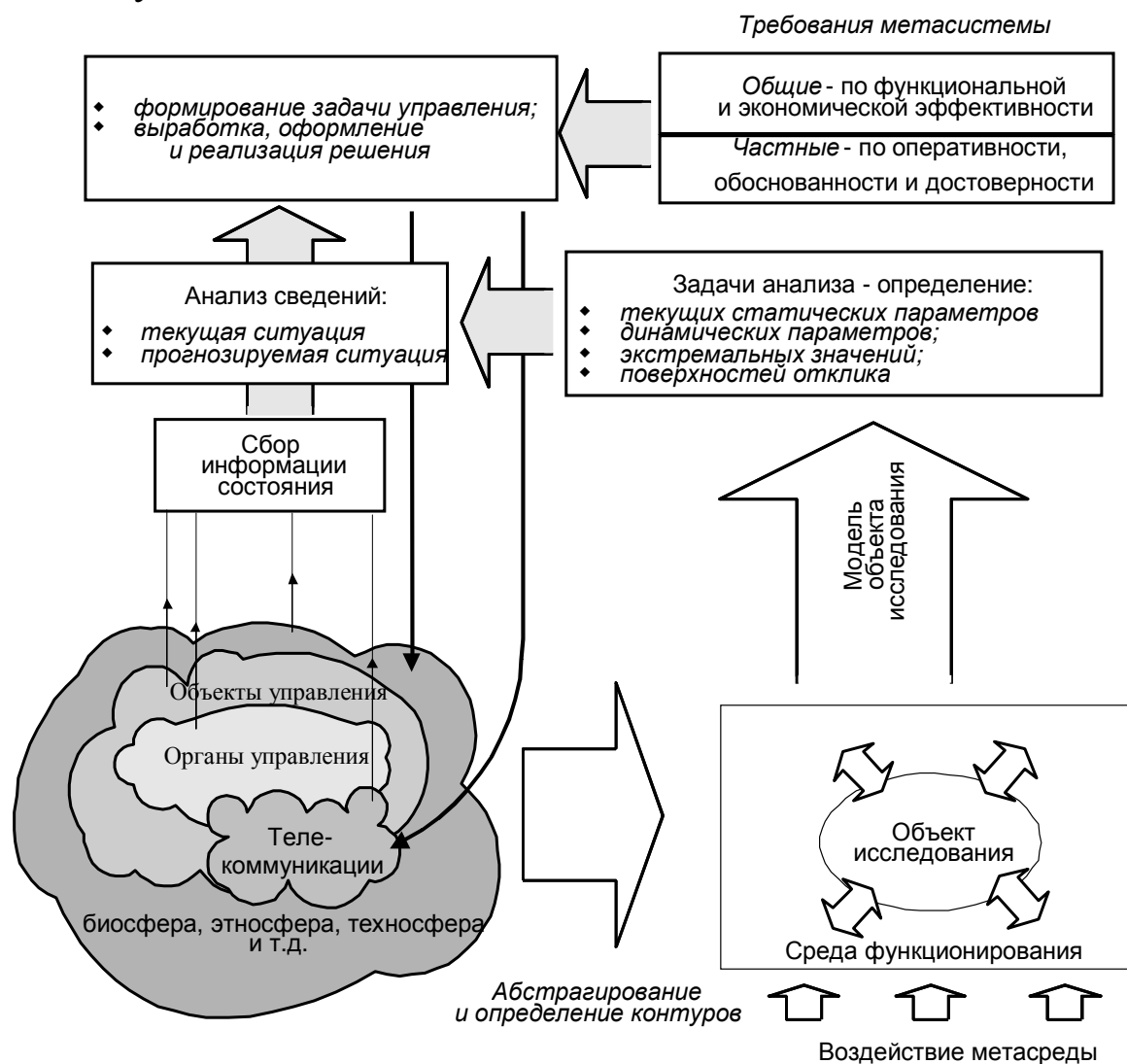


Рис. 5.1. Место задач анализа данных в процессе управления

рост требований к эффективности управления (и в первую очередь – оперативности, обоснованности и достоверности процессов принятия решений) – как следствие возрастающей конкурентной борьбы неантагонистического или антагонистического характера и степени влияния эффективности управления на эффективность и качество функционирования любой системы (со-

вершенствование управления по ряду причин всегда более предпочтительнее совершенствования объекта управления).

Таким образом, наблюдается объективный рост сложности объектов и задач управления, требующий адекватных изменений средств поддержки принятия оперативных и стратегических решений в нестационарных средах.

Во-вторых, в настоящее время создано и широко используется большое множество информационных систем различного уровня, которые относятся к классу информационных систем операционной обработки данных (или так называемым транзакционным системам) – OLTP. Реализуя основные информационные процессы, связанные с функциями учета и контроля (мониторинга), OLTP системы создают возможность накопления первичных данных (и прежде всего - временных рядов наблюдений) для их последующего анализа в интересах подготовки и принятия решений на основе прогноза развития ситуаций.

В третьих, за последнее десятилетие:

существенно изменились возможности средств вычислительной техники, а именно – качественно возросли их вычислительная мощность и возможности реализации эффективного диалога с пользователями, на несколько порядков увеличилось возможности хранения больших объемов информации, резко снизилась удельная стоимость вычислительных ресурсов;

на базе достижений в области информационных технологий и телекоммуникаций стало возможным формирование единых информационных пространств предприятий, организаций, корпораций, сообществ и т. д., резко увеличивающее объемы оперативно доступной информации самого различного характера.

Совокупность указанных факторов привела к осознанию возможности принципиально нового подхода к осуществлению процессов сбора и анализа информации в системах стратегического и оперативного планирования, оперативного и технологического управления, который находит свое выражение в создании и применении аналитических технологий, ориентированных на решение задач поддержки принятия решений. К таким задачам относятся:

оценка текущего и прогнозируемого состояния объекта управления и (или) среды его функционирования;



обнаружение и исследование скрытых закономерностей, факторов, тенденций и взаимосвязей;

обобщение информации как агрегация и интеграция сведений различного характера;

формирование альтернативных решений и выбор "оптимального" в соответствии с заданным критерием, а также результатами анализа сценариев развития ситуаций;

моделирование процесса эволюции состояния объекта в нестационарной неоднородной среде и т.д.

В свою очередь, повышение качества процессов обобщения и анализа информации на основе аналитических технологий и их реализующих современных средств достигается автоматизированной реализацией процессов:

выявления скрытых закономерностей и факторов;

количественной оценки факторов влияния и угроз в сложившейся ситуации;

использования опыта на основе автоматизированного формирования и поиска прецедентов анализируемых ситуаций в массивах ретроспективных данных;

высокодостоверного прогнозирования эволюции состояния объекта исследования, в т.ч. выявления предпосылок к скачкообразному изменению этого состояния.

Перечисленные ключевые возможности аналитических технологий позволяют рассматривать их как аналитическую и информационную основу не только процессов обобщения и анализа информации о текущем и прогнозируемом состояниях объекта управления и среды его функционирования, но и процессов подготовки (выработки и оценки) вариантов решений (рис. 5.2).

Данный аспект применения аналитических технологий осуществляется посредством использования:

результатов прогнозирования развития процессов и явлений для оценки эффективности возможных (в т.ч. альтернативных) вариантов решений, планов и программ действий, а также управляющих воздействий иного рода;

способов ретроспективного анализа содержимого хранилищ данных для поиска типовых ситуаций, требующих управления, и выработки вариантов решений на основе аналогий и накопленного опыта управления;

специально разработанного комплекса анализа эффективности управленческих решений, основанных на формировании виртуальных сценариев развития частично управляемых ситуаций.

Аналитические технологии – обобщающий термин, применяемый для обозначения подходов к комплексному использованию ряда формальных и эвристических методов и информационных технологий. Основными отличительными чертами таких подходов являются:

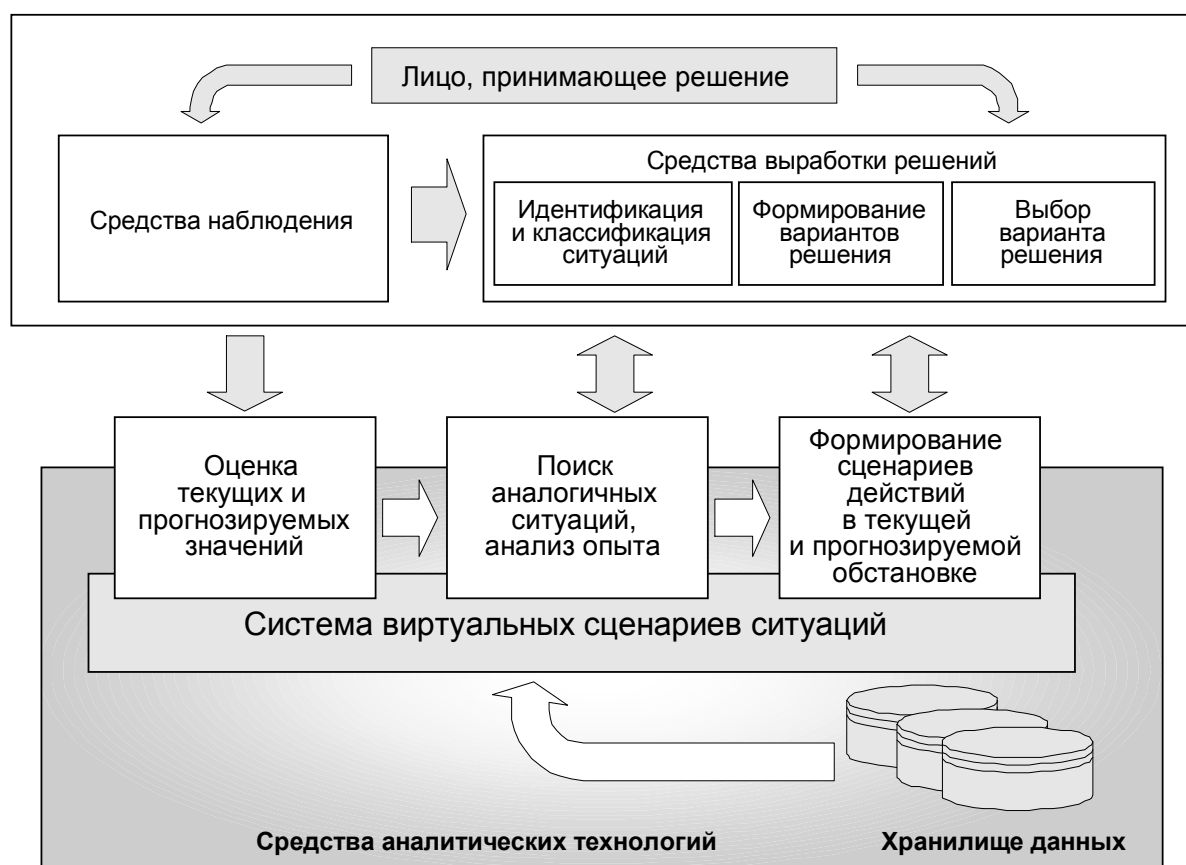


Рис. 5.2. Участие аналитических технологий в подготовке вариантов решений

ориентация на динамические данные, позволяющие строить временные ряды и осуществлять как ретроспективный анализ характеристик объектов, процессов и явлений, так и решение задач оперативного и долгосрочного прогнозирования с выявлением неявных зависимостей и скрытых влияний (воздействий);

объединение элементов технологий хранилищ данных и тематических витрин данных, оперативной аналитической обработки данных, интеллектуального анализа данных и прецедент-

ного прогноза, ставшее возможным благодаря появлению современных высокопроизводительных архитектур и платформ для создания вычислительной среды.

Таким образом, аналитическая технология – это средство решения задач анализа и прогнозирования состояния объектов в нестационарных средах на основе накопления "истории", ретроспективного анализа накапливаемых данных и динамического формирования (улучшения) формальных моделей явлений и процессов, что стало возможным только при определенном уровне эволюции вычислительной техники.

Основным содержанием ИАД является автоматизированный прецедентный анализ сверхбольших объемов ретроспективной информации, сконцентрированных в информационных хранилищах.

Основной задачей, решаемой ИАД, является вскрытие неявных закономерностей, взаимозависимостей и факторов влияния в интересах задач ситуационного анализа и прогноза на основе широкого спектра математических инструментов – от фундаментальных статистических методов регрессионного и многофакторного дисперсионного анализа до новых кибернетических алгоритмов, основанных на нейронных сетях, эволюционном моделировании и т.п. (рис. 5.3).

По существу, методология ИАД сводится к структуризации прогностической информации на основе прецедентного анализа исторических, ретроспективных данных. Важнейшей особенностью ИАД является гармоничное сочетание строгих, формальных методов (алгоритмов) анализа количественной информации с методами неформального, качественного анализа, опирающегося на субъективные знания экспертов и эвристические приемы их работы.

Сочетание прецедентного анализа ретроспективных данных с возможностями прогностических методов позволяет рассматривать средства ИАД как составную (и наиболее важную) часть современных систем поддержки принятия решений (СППР), реализующих контроль текущего и прогнозируемого состояния объекта управления, и способных осуществлять подготовку и количественную оценку возможных вариантов решений (рис. 5.4).

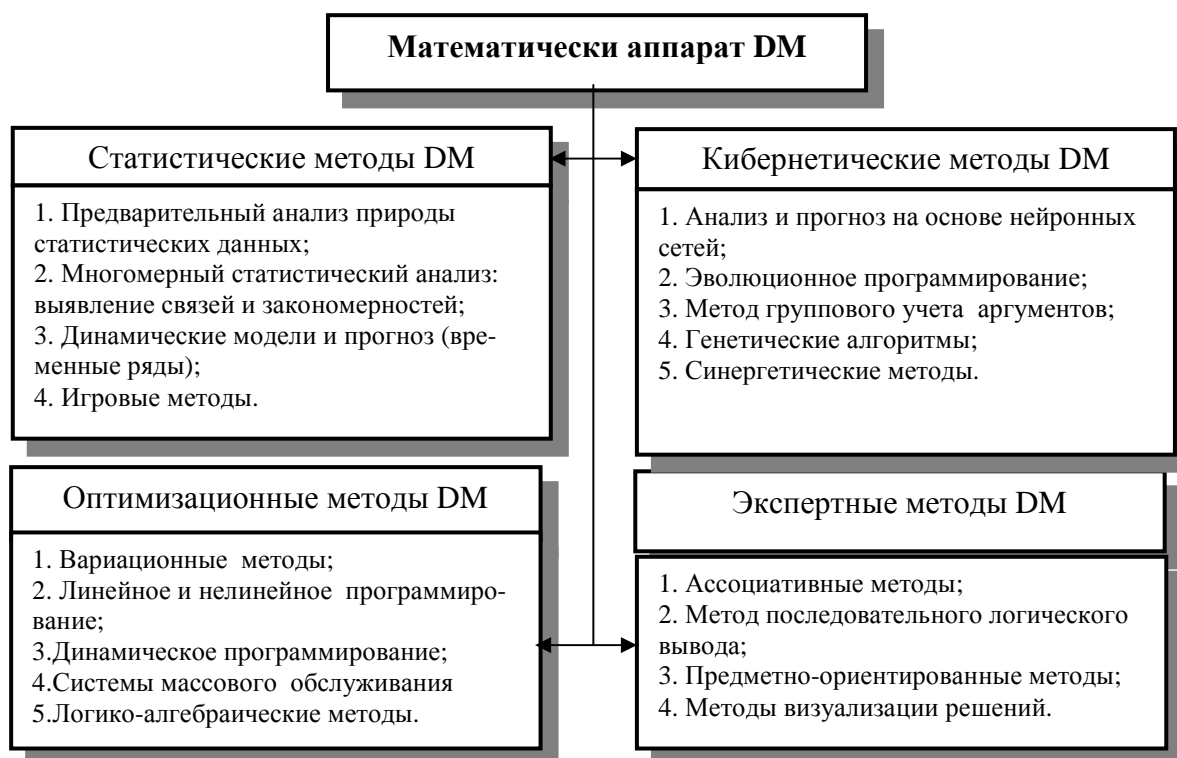


Рис. 5.3. Основные методы ИАД

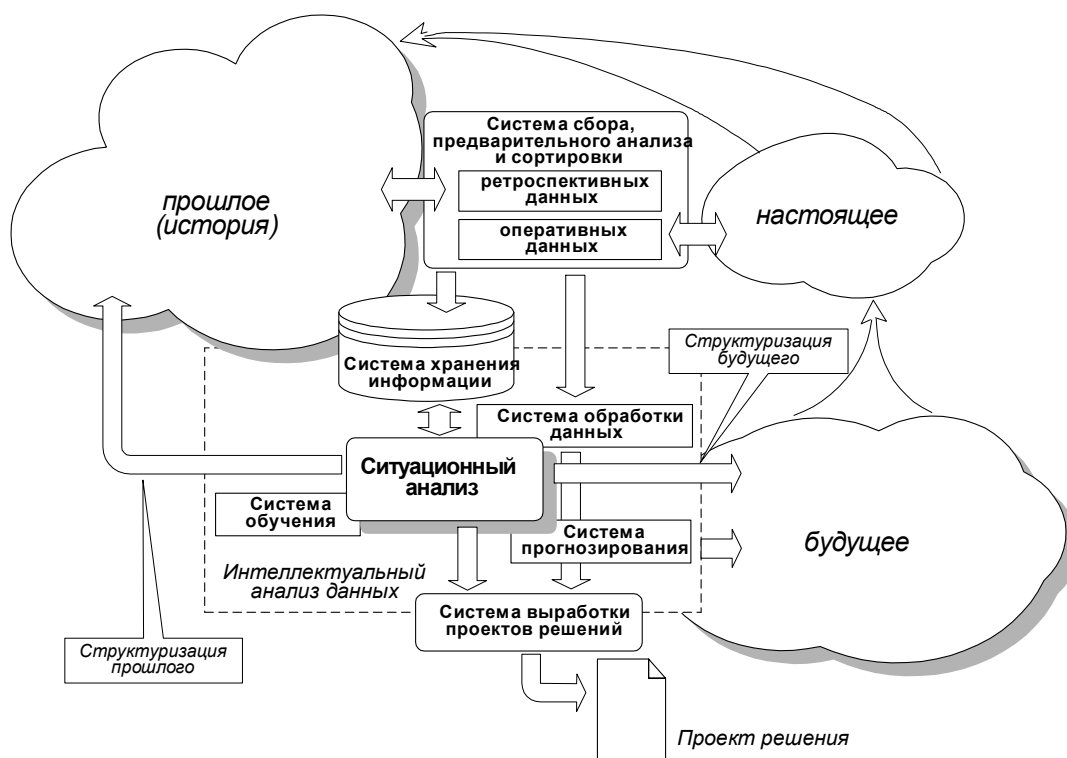


Рис. 5.4. Схема содержания технологии ИАД

Акцент на количественной методологии позволяет перенести центр тяжести процедуры выработки проекта решения с эвристических логико-интуитивных методов, характерных для тра-

диционной экспертной технологии, на мощную глубоко формализованную платформу фундаментальной и прикладной математики. При этом качественный, экспертный анализ также сохраняется, но теперь основным центром его приложения является не всеобъемлющий и трудно охватываемый обзор исторического опыта, а лишь вполне обозримый объем конечного набора уже сформированных вариантов решений на фоне подготовленных прогностических сценариев.

Таким образом, речь, по существу, идет о новой форме гибридного интеллекта, сочетающей получение формально-оптимальных решений с эвристическим принятием терминальных качественных решений.

Методология ИАД существенно шире, чем только прецедентный прогноз и синергетика – обзорные сведения об этих методах приведены для того, чтобы подчеркнуть принципиально новые возможности, которые открывают современные средства хранения и обработки данных.

Следует отметить еще одну весьма существенную проблему, возникающую непосредственно в процессе применения математико-аналитических инструментальных средств (как "классических", так и относящихся к средствам ИАД).

Суть проблемы состоит том, что любой предметный эксперт является глубоким профессионалом в своей области, но он не является экспертом в области математической обработки данных, в существенных тонкостях применения того или иного математического аппарата. В результате некорректного применения математики, заложенной в программно реализованных средствах поддержки работы предметного эксперта, полученные формализованные решения во многих ситуациях оказываются неверными (неадекватными реальным ситуациям); отсюда возникает кризис доверия к методам автоматизированного статистического анализа, машинным методам прогнозирования сценариев и т.п.

Одновременная работа предметного эксперта с математиком также не всегда увенчивается успехом: очень мало математиков способны выйти из своей математической, узко специальной области и понять комплексную проблему предметного эксперта.

Иерархия средств, составляющих основное содержание аналитических технологий и фактически отображающая основную

парадигму построения аналитических информационных систем (АИС), показана на рис. 5.5.

Обязательным условием применения аналитических технологий является наличие неких первичных данных (данных наблюдений, мониторинга), источником которых выступают OLTP-системы. Реализуя основные информационные процессы, связанные с функциями учета и контроля (мониторинга), OLTP системы создают возможность накопления первичных данных (и прежде всего – временных рядов наблюдений) для их последующего анализа в интересах подготовки и принятия решений на основе прогноза развития ситуаций.

Даже если имеющиеся OLTP-системы согласованы по способам классификации и кодирования информации и допускают выполнение распределенных запросов к их БД, их непосредственное использование для аналитической обработки имеющихся в них данных затруднено следующими обстоятельствами:

- распределенное хранение в нескольких разнородных БД существенно затрудняет комплексное использование этих данных из-за сложности их одновременной выборки из нескольких источников, возможных пересечений перечней хранимых данных и (или) нарушений их согласованности (целостности);

- запросы на выборку ретроспективных данных снижают реактивность транзакционных систем;

- глубина ретроспективного анализа определяется принятыми ограничениями для длительности периода архивного хранения данных.

Из-за различий характера хранения и использования данных в транзакционных и аналитических системах данные, получаемые от OLTP-систем (первичные данные) подлежат помещению в хранилище данных (ХД).

ХД – предметно-ориентированный, интегрированный, неизменяемый, поддерживающий хронологию набор данных, организованный для целей поддержки управления. При этом возможность работы пользователей с форматом оперативных данных не предусмотрена, т.е. ХД не обеспечивает доступ к БД транзакционных систем со стороны пользователей.

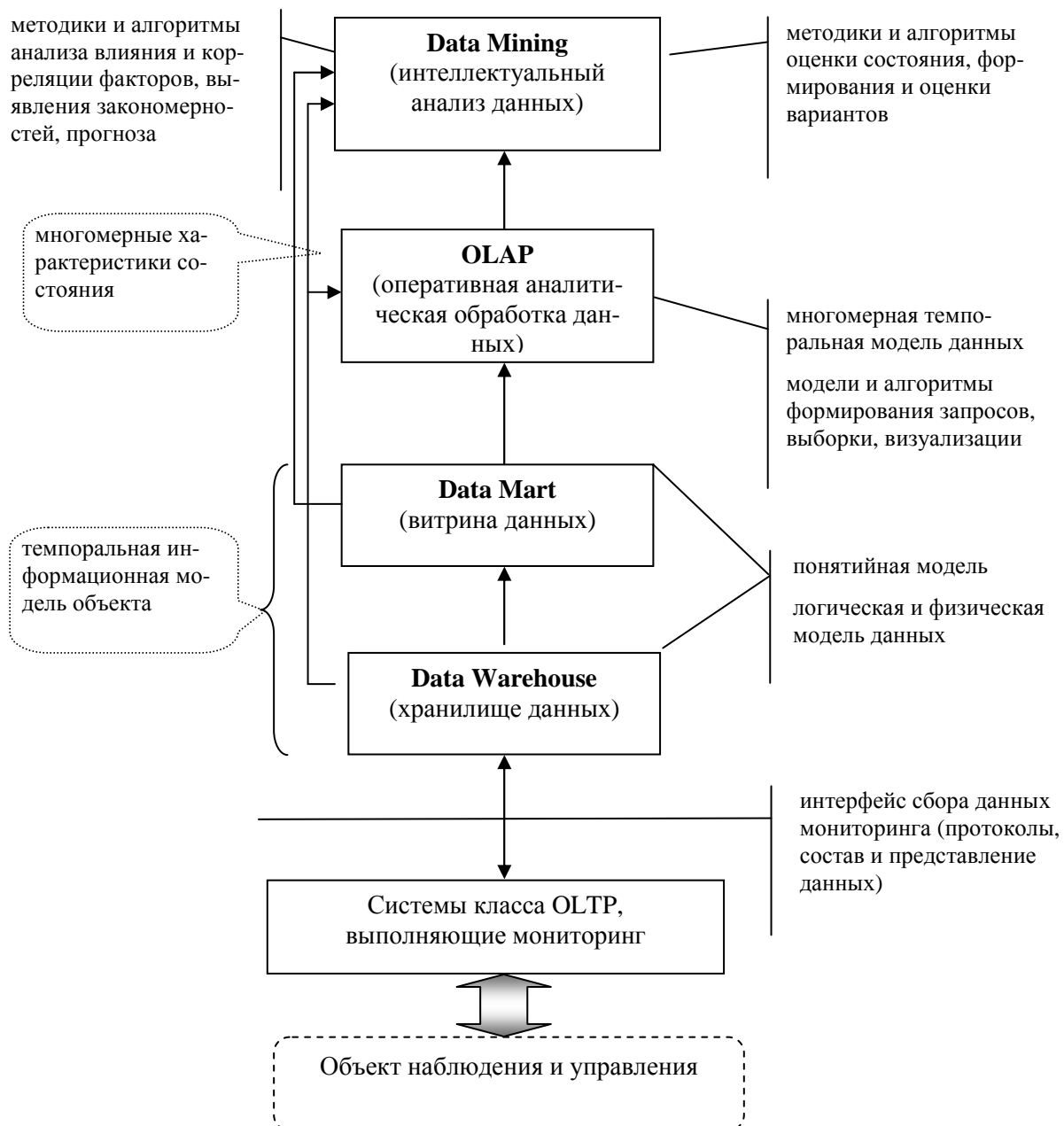


Рис. 5.5. Иерархия средств реализации аналитических технологий

К числу главных преимуществ ХД относят следующее:

1) Единый источник информации: предприятие получает выверенную единую информационную среду, на которой будут строиться все справочно-аналитические приложения в той предметной области, по которой построено хранилище. Эта среда бу-

дет обладать единым интерфейсом, унифицированными структурами хранения, общими справочниками и другими корпоративными стандартами, что облегчает создание и поддержку аналитических систем.

2) Производительность: физические структуры хранилища специальным образом оптимизированы для выполнения абсолютно произвольных выборок, что позволяет строить действительно быстрые системы запросов.

3) Интегрированность: интеграция данных из разных источников уже сделана, поэтому не надо каждый раз производить соединение данных для запросов, требующих информацию из нескольких источников.

4) Историчность и стабильность: информационное хранилище данных нацелено на долговременное хранение информации за период 10-15 лет с адаптацией хранимой информации к изменениям структуры и параметров, происходящих в отображаемом объекте. Благодаря этому появляется возможность осуществлять исторический анализ информации.

5) Независимость: разделение информационного хранилища и OLTP-систем существенно снижает нагрузку на последние со стороны аналитических приложений; тем самым производительность существующих систем не ухудшается, а на практике происходит уменьшение времени отклика и улучшение доступности систем.

Вместе или вместо ХД могут создаваться и использоваться витрины данных (ВД). Есть следующие варианты трактовки данных понятия витрины (киоска) данных:

усеченное ХД – т. е. хранилище, относящееся только к некоторой тематике (например, экономика и финансы предприятия, производство продукции и т. д.);

хранилище, которое не обладает в полной мере всеми свойствами, перечисленными выше;

ХД, создаваемое для решения частной задачи и уничтожаемое после того, как необходимость в нем отпала.

В современных АС ХД и ВД имеют реляционную структуру и являются источниками данных для средств оперативной аналитической обработки и интеллектуального анализа данных.



Основные принципы оперативной аналитической обработки данных сформулировал в 1993 г. Е. Ф. Кодд – автор теории реляционных БД. Позже его определение было переработано в так называемый тест FASMI, требующий, чтобы OLAP-приложение предоставляло возможности быстрого анализа разделяемой многомерной информации:

Fast (быстрый) – анализ должен производиться одинаково быстро по всем аспектам информации;

Analysis (анализ) – должна быть возможность осуществлять основные типы числового и статистического анализа, predetermined разработчиком приложения или произвольно определяемого пользователем;

Shared (разделяемой) – множество пользователей должно иметь доступ к данным, при этом необходимо контролировать доступ к конфиденциальной информации.

Multidimensional (многомерной) – это основная, наиболее существенная характеристика OLAP;

Information (информации) – приложение должно иметь возможность обращаться к любой нужной информации, независимо от ее объема и места хранения.

Моделью представления данных в OLAP служит многомерный куб (или множество связанных кубов, образующее гиперкуб). Многомерность в OLAP-приложениях может быть разделена на три уровня:

многомерное представление данных — средства конечного пользователя, обеспечивающие многомерную визуализацию и манипулирование данными (слой многомерного представления абстрагирован от физической структуры данных и воспринимает данные как многомерные);

многомерная обработка – язык формулирования многомерных запросов и средство, которое может выполнить такой запрос;

многомерное хранение — средства физической организации данных, обеспечивающие эффективное выполнение многомерных запросов.

Таким образом, применение OLAP предполагает построение многомерных кубов данных (МКД) с выполнением агрегации значений, помещаемых в ячейки куба, отображение МКД и манипулирование ими.

Следует отметить, что даже применение только OLAP дает существенный выигрыш в эффективности управления в самых различных областях.

Интеллектуальный анализ данных предполагает применение разного рода математических методов, перечисленных выше, для выполнения содержательной обработки данных с целью получения формальных решений задач анализа и прогнозирования. Источниками исходных данных для различных средств ИАД могут служить как реляционные БД ХД, так и МКД OLAP.

Очевидно, что методология ИАД существенно шире, чем только прецедентный прогноз и синергетика – обзорные сведения об этих методах приведены для того, чтобы подчеркнуть принципиально новые возможности, которые открывают современные средства хранения и обработки данных.

АИС является автоматизированной информационной системой, ориентированной на задачи поддержки принятия решения. В общем случае АИС может рассматриваться как организационно-технический комплекс автоматизированной поддержки принятия решений, создаваемый в интересах решения задач стратегического анализа ситуаций и прогнозирования их развития в различных областях управленческой и исследовательско - аналитической деятельности.

Основными условиями, необходимыми для реализации содержательных возможностей АИС по сбору и аналитическому анализу информации, являются (рис. 5.6):

наличие совокупности математических методов, адекватных задачам анализа информации и специфике обрабатываемых данных (функциональное наполнение АИС, составляющее ее аналитическую оболочку);

организация хранения больших массивов фактографических и документальных сведений и оперативного доступа к ним с использованием гибкой системы запросов (информационная подсистема АИС);

организация сбора и хранения сведений, характеризующих управляемый объект и среду его функционирования, с требуемой достоверностью, полнотой и непротиворечивостью (средства сбора информации);



Рис. 5.6. Основные компоненты АИС

наличие совокупности программных и технических средств, обеспечивающих функционирование АИС (вычислительная среда АИС).

Средства АИС, обеспечивающие выполнение указанных выше условий, можно кратко охарактеризовать следующим образом.

Теоретическую и технологическую базу для создания функционального наполнения (аналитической оболочки) АИС образуют технологии интеллектуального анализа данных и гибридного интеллекта, некоторые аспекты которых описаны выше. Таким образом, в качестве алгоритмического инструментария АИС выступает комплекс методов и средств стратегического анализа и прогнозирования сложных нестационарных процессов (включая процессы со скачкообразным изменением состояния), основанный на сочетании классических методов прогнозирования (экстраполяция, статистические методы предсказания) и новых, но уже известных кибернетических методов прогностики (эволюционное моделирование, нейронные сети и т. д.).

Создание информационной подсистемы базируется на концепции и технологии ХД и OLAP.

Ключевыми особенностями этих технологий являются: ориентация на поддержку хранения больших массивов данных; специальная структура хранения данных (в частности, топологии "звезда" и "снежинка"), что облегчает формирование запросов на выборку данных; разделение информационных массивов, используемых для оперативной обработки и для решения задач анализа; предварительное агрегирование многомерных данных, что обеспечивает быстрый переход от обобщенных к детализированным данным и обратно.

Реализация вычислительной среды и средств сбора информации базируется на известных технологиях и средствах, и осуществляется в рамках той метасистемы, по отношению к которой АИС осуществляет решение задач аналитической службы.

Таким образом, АИС как СППР интегрирует ряд технологий и средств, схематично показанных на рис. 5.7.

Процессы функционирования АИС, являющиеся основой для синтеза ее функциональной структуры, определяются в рамках классического выделения управляющего, основного (содержательного) и обеспечивающего процессов, и формализуются как три взаимосвязанных уровня функционирования АИС:

стратегический (управляющий) – уровень управления охватываемой предметной областью и классом потенциально решаемых задач в рамках динамически формулируемых конкретных проблем анализа и прогностики, представляющий собой процесс управления структурой и содержанием хранимых предметных сведений (включая процессы их сбора), а также множеством программно реализованных математических методов, обеспечивающих решение задач ИАД;

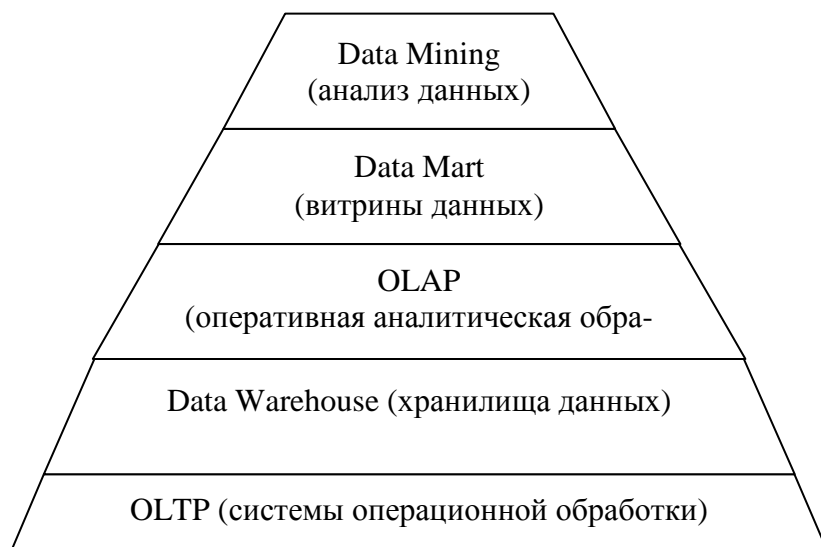


Рис. 5.7. "Пирамида" средств АИС

оперативный (основной) – уровень отдельных задач в рамках общей проблемы, на котором по отношению к каждой задаче выполняются этапы ее постановки, анализа и решения;

технологический (обеспечивающий) – уровень функционирования служб АИС, составляющих ее технологические профили (телекоммуникации, разграничение доступа, защита информации, архивация результатов решения задач и т. д.).

В качестве примера можно привести следующий вариант организации процесса решения задач, т. е. вариант организации функционирования АИС на оперативном уровне (рис. 5.8):

вербальная постановка задачи, подлежащей решению, лицом, принимающим решения (ЛПР), и ее дальнейшая формализация предметным экспертом и экспертом - аналитиком;

выборка данных, необходимых для решения задачи, из хранилища в витрины данных (возможно – многократно);

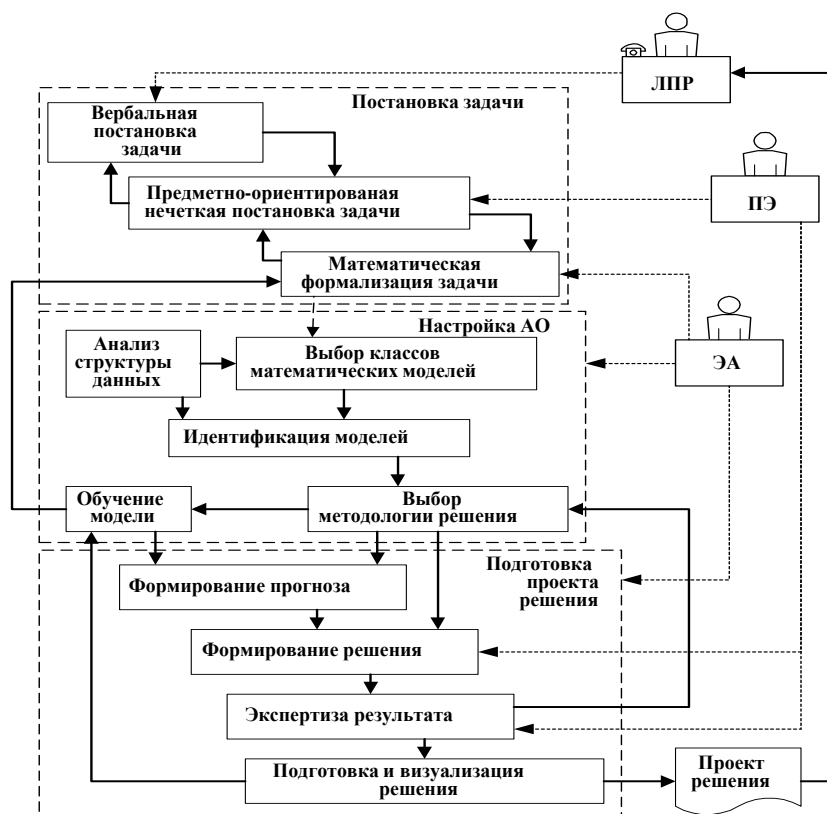


Рис. 5.8. Схема процесса решения задач в АИС

анализ структуры выбранных данных, по результатам которого осуществляется выбор математических моделей и схемы решения задачи;

формирование прогнозов, вариантов решений и их экспертизу – с использованием множества виртуальных сценариев.

При этом предполагается следующее разграничение функций пользователей АИС:

лицо, принимающее решения (ЛПР) – руководитель, осуществляющий постановку проблем и задач и являющийся для АИС конечным потребителем получаемых результатов;

предметный эксперт (ПЭ) – специалист предметной области, несущий основную нагрузку по определению состава и содержательному анализу предметных сведений, а также результатов их обработки;

эксперт-аналитик (ЭА) – специалист, осуществляющий выбор методик решения задач и выполняющий их решение на основе формальных методов обработки;

администратор данных (АД) – лицо, выполняющее действия по сбору необходимых предметных сведений и помещению их в информационное хранилище;

администратор системы – лицо, осуществляющее управление службами АИС, ее конфигурированием и содержательным наполнением (на рис. 5.8 не показан).

Примечательно, что отмеченная выше проблема взаимодействия ПЭ и ЭА решается в АИС путем их совместного участия в решении задач с разграниченными функциями и последовательным уточнением и согласованием содержания каждого этапа.

Детальная архитектура любой АИС и методология ее создания определяются конкретными техническими решениями, которые специфичны для каждого конкретного разработчика.

Таким образом, аналитические технологии и разного рода АИС, построенные как средства реализации этих технологий, на сегодня являются наиболее современным и концентрированным выражением инструментария решения задач анализа, прогнозности и поддержки принятия решений в целом.

Привлекательность применения АИС помимо всего прочего обусловлена тем, что в сложных, нестационарных ситуациях, обусловленных большим числом разнообразных гетерогенных взаимосвязанных факторов, эксперты, как правило, не находят рациональных решений, их мнения оказываются субъективными и противоречивыми. В большинстве случаев мозг эксперта не способен экстраполировать развитие ситуаций, находящихся под воздействием более 3-5 независимых факторов. Для взаимосвязанных воздействий даже опытный эксперт способен корректно учесть не более трех факторов влияния. В то же время большинство реальных стратегических ситуаций требуют учета, как минимум, от 6 до 20 значимых факторов влияния.

Можно ожидать, что внедрение аналитических информационных систем позволит существенно повысить крайне низкий технологический уровень существующих и вновь создаваемых аналитических служб и центров за счет внедрения современных аналитических технологий, поддержанных соответствующими инструментальными средствами анализа данных. Однако следует отметить, что: содержательное наполнение аналитических информационных систем в большой степени зависит от специфики

предметной области и объекта управления, однако формальных методов его определения на сегодняшний день не существует, и выбор конкретных математических методов и обрабатываемых данных осуществляется исключительно эвристически с последующим их уточнением в ходе применения системы; направление исследований, связанное с созданием аналитических информационных систем еще находится в начальной стадии своего развития, оно определилось и бурно развивается с середины 90-х годов, поэтому в настоящее время имеются только лишь отдельные образцы соответствующих инструментальных средств и некоторый опыт их создания и применения.

## 5.2. Методы и стадии ИАД

ИАД (Data Mining) - мультидисциплинарная область, возникающая и развивающаяся на базе таких наук как прикладная статистика, распознавание образов, искусственный интеллект, теория баз данных и т.д. ИАД - это процесс обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.

Суть и цель технологии ИАД можно охарактеризовать так: это технология, которая предназначена для поиска в больших объемах данных неочевидных, объективных и полезных на практике закономерностей.

**Неочевидных** - найденные закономерности не обнаруживаются стандартными методами обработки информации или экспертным путем.

**Объективных** - обнаруженные закономерности будут полностью соответствовать действительности, в отличие от экспертного мнения, которое всегда является субъективным.

**Практически полезных** - выводы имеют конкретное значение, которому можно найти практическое применение.

Различные инструменты ИАД имеют различную степень "дружелюбности" интерфейса и требуют определенной квалификации пользователя. Поэтому программное обеспечение должно соответствовать уровню подготовки пользователя. Использование ИАД должно быть неразрывно связано с повышением квали-



фикации пользователя. Однако специалистов по ИАД, которые бы хорошо разбирались в бизнесе, пока еще мало.

**Извлечение полезных сведений невозможно без хорошего понимания сути данных.** Необходим тщательный выбор модели и интерпретация зависимостей или шаблонов, которые обнаружены. Поэтому работа с такими средствами требует тесного сотрудничества между экспертом в предметной области и специалистом по инструментам ИАД. Построенные модели должны быть грамотно интегрированы в бизнес-процессы для возможности оценки и обновления моделей. В последнее время системы ИАД поставляются как часть технологии хранилищ данных.

**Сложность подготовки данных.** Успешный анализ требует качественной предобработки данных. По утверждению аналитиков и пользователей баз данных, процесс предобработки может занять до 80% процентов всего ИАД-процесса. Таким образом, чтобы технология работала на себя, потребуется много усилий и времени, которые уходят на предварительный анализ данных, выбор модели и ее корректировку.

**Большой процент ложных, недостоверных или бессмысленных результатов.** С помощью ИАД можно отыскивать действительно ценную информацию, которая вскоре даст большие дивиденды в виде финансовой и конкурентной выгоды. Однако ИАД достаточно часто делает множество ложных и не имеющих смысла открытий. Многие специалисты утверждают, что ИАД-средства могут выдавать огромное количество статистически недостоверных результатов. Чтобы этого избежать, необходима проверка адекватности полученных моделей на тестовых данных.

**Высокая стоимость.** Качественная ИАД-программа может стоить достаточно дорого для компании. Вариантом служит приобретение уже готового решения с предварительной проверкой его использования.

**Наличие достаточного количества репрезентативных данных.** Средства ИАД теоретически не требуют наличия строго определенного количества ретроспективных данных. Эта особенность может стать причиной обнаружения недостоверных, ложных моделей и, как результат, принятия на их основе неверных решений. Необходимо осуществлять контроль статистической значимости обнаруженных знаний.

**К методам и алгоритмам ИАД** относятся следующие: искусственные нейронные сети, деревья решений, символьные правила, методы ближайшего соседа и k-ближайшего соседа, метод опорных векторов, байесовские сети, линейная регрессия, корреляционно-регрессионный анализ; иерархические методы кластерного анализа, неиерархические методы кластерного анализа, в том числе алгоритмы k-средних и k-медианы; методы поиска ассоциативных правил, в том числе алгоритм Apriori; метод ограниченного перебора, эволюционное программирование и генетические алгоритмы, разнообразные методы визуализации данных и множество других методов.

Большинство аналитических методов, используемые в технологии ИАД - это известные математические алгоритмы и методы. Новым в их применении является возможность их использования при решении тех или иных конкретных проблем, обусловленная появившимися возможностями технических и программных средств. Следует отметить, что большинство методов ИАД были разработаны в рамках теории искусственного интеллекта.

**ИАД может состоять из двух или трех стадий:**

**Стадия 1.** Выявление закономерностей (свободный поиск).

Осуществляется исследование набора данных с целью поиска скрытых закономерностей. Предварительные гипотезы относительно вида закономерностей здесь не определяются.

Закономерность - существенная и постоянно повторяющаяся взаимосвязь, определяющая этапы и формы процесса становления, развития различных явлений или процессов.

Свободный поиск представлен такими действиями: выявление закономерностей условной логики; выявление закономерностей ассоциативной логики; выявление трендов и колебаний.

**Стадия 2.** Использование выявленных закономерностей для предсказания неизвестных значений (прогностическое моделирование). Здесь обнаруженные закономерности используются непосредственно для прогнозирования.

Прогностическое моделирование включает: предсказание неизвестных значений; прогнозирование развития процессов.

В процессе прогностического моделирования решаются задачи классификации и прогнозирования.

При решении задачи классификации результаты работы первой стадии (индукции правил) используются для отнесения нового объекта, с определенной уверенностью, к одному из известных, predetermined классов на основании известных значений.

При решении задачи прогнозирования результаты первой стадии (определение тренда или колебаний) используются для предсказания неизвестных (пропущенных или же будущих) значений целевой переменной (переменных).

В дополнение к этим стадиям иногда вводят **стадию валидации**, следующую за стадией свободного поиска. Цель валидации - проверка достоверности найденных закономерностей.

**Стадия 3. Анализ исключений** - стадия предназначена для выявления и объяснения аномалий, найденных в закономерностях. Действие, выполняемое на этой стадии, - выявление отклонений. Для выявления отклонений необходимо определить норму, которая рассчитывается на стадии свободного поиска.

### **Классификация технологических методов ИАД:**

непосредственное использование или сохранение данных - исходные данные хранятся в явном детализированном виде и непосредственно используются на стадиях прогностического моделирования;

выявление и использование формализованных закономерностей, или дистилляция шаблонов - один образец (шаблон) информации извлекается из исходных данных и преобразуется в некие формальные конструкции;

логические методы включают: нечеткие запросы и анализы; символьные правила; деревья решений; генетические алгоритмы.

методы кросс-табуляции: агенты, баесовские (доверительные) сети, кросс-табличная визуализация;

методы на основе уравнений - статистические методы и нейронные сети;

кибернетические методы: искусственные нейронные сети, эволюционное программирование; генетические алгоритмы; ассоциативная память (поиск аналогов, прототипов); нечеткая логика; деревья решений; системы обработки экспертных знаний.

### 5.3. Задачи ИАД

**Классификация.** В результате решения задачи классификации обнаруживаются признаки, которые характеризуют группы объектов исследуемого набора данных - классы; по этим признакам новый объект можно отнести к тому или иному классу. Методы решения: ближайшего соседа; k-ближайшего соседа; байесовские сети; индукция деревьев решений; нейронные сети.

**Кластеризация** является логическим продолжением идеи классификации, особенность заключается в том, что классы объектов изначально не predetermined. Результатом кластеризации является разбиение объектов на группы.

В ходе решения задачи поиска **ассоциативных** правил отыскиваются закономерности между связанными событиями в наборе данных. Отличие ассоциации от двух предыдущих задач: поиск закономерностей осуществляется не на основе свойств анализируемого объекта, а между несколькими событиями, которые происходят одновременно.

**Последовательность** позволяет найти временные закономерности между транзакциями. Подобна ассоциации, но ее целью является установление закономерностей не между одновременно наступающими событиями, а между событиями, связанными во времени, т.е. определяется высокой вероятностью цепочки связанных во времени событий. Ассоциация является частным случаем последовательности с временным лагом, равным нулю.

В результате решения задачи **прогнозирования** на основе особенностей исторических данных оцениваются пропущенные или же будущие значения целевых численных показателей.

**Определение отклонений или выбросов**, анализ отклонений или выбросов - обнаружение и анализ данных, наиболее отличающихся от общего множества данных, выявление так называемых нехарактерных шаблонов.

Задача **оценивания** сводится к предсказанию непрерывных значений признака.

**Анализ связей** - задача нахождения зависимостей в наборе данных.

В результате **визуализации** создается графический образ анализируемых данных. Для решения задачи визуализации ис-

пользуются графические методы, показывающие наличие закономерностей в данных.

**Подведение итогов** - задача, цель которой - описание конкретных групп объектов из анализируемого набора данных.

## **5.4. Процесс ИАД**

Процесс ИАД является своего рода исследованием и неразрывно связан с процессом принятия решений. Строится модель и в процессе принятия решений эта модель эксплуатируется. Он включает следующие этапы:

- анализ предметной области;
- постановка задачи;
- подготовка данных;
- построение моделей;
- проверка и оценка моделей;
- выбор модели;
- применение модели;
- коррекция и обновление модели.

### **Этап 1. Анализ предметной области**

В процессе изучения предметной области должна быть создана ее модель. Знания из различных источников должны быть формализованы при помощи каких-либо средств. Это могут быть текстовые описания предметной области или специализированные графические нотации.

### **Этап 2. Постановка задачи**

Включает следующие шаги: формулировка и формализация задачи.

Формулировка задачи включает также описание статического и динамического поведения исследуемых объектов. Описание статики подразумевает описание объектов и их свойств. При описании динамики описывается поведение объектов и те причины, которые влияют на их поведение.

Технология ИАД не может заменить аналитика и ответить на те вопросы, которые не были заданы. Поэтому постановка задачи является необходимым этапом процесса ИАД, поскольку

именно на этом этапе определяется, какую же задачу необходимо решить. Иногда этапы анализа предметной области и постановки задачи объединяют в один этап.

### **Этап 3. Подготовка данных**

Цель этапа: разработка базы данных. Подготовка данных является важнейшим этапом, от качества выполнения которого зависит возможность получения качественных результатов всего процесса. На этап подготовки данных может быть потрачено до 80% всего времени, отведенного на проект. Этап включает в себя:

определение и анализ требований к данным;

сбор данных;

определение необходимого количества данных;

При определении необходимого количества данных следует учитывать, являются ли данные упорядоченными или нет.

Если данные упорядочены и идет работа с временными рядами, желательно знать, включает ли такой набор данных сезонную/циклическую компоненту. В случае присутствия в наборе данных сезонной/циклической компоненты, необходимо иметь данные как минимум за один сезон/цикл.

Если данные не упорядочены, то есть события из набора данных не связаны по времени, в ходе сбора данных следует соблюдать следующие правила.

Недостаточное количество записей в наборе данных может стать причиной построения некорректной модели. С точки зрения статистики, точность модели увеличивается с увеличением количества исследуемых данных. Возможно, некоторые данные являются устаревшими или описывают какую-то нетипичную ситуацию, и их нужно исключить из базы данных. Алгоритмы, используемые для построения моделей на сверхбольших базах данных, должны быть масштабируемыми.

При использовании многих алгоритмов необходимо определенное (желательное) соотношение входных переменных и количества наблюдений. Количество записей в наборе данных должно быть значительно больше количества факторов.

Набор данных должен быть репрезентативным и представлять как можно больше возможных ситуаций. Пропорции пред-

ставления различных примеров в наборе данных должны соответствовать реальной ситуации.

Анализировать можно как качественные, так и некачественные данные. Результат будет достигнут и в том, и в другом случае. Для обеспечения качественного анализа необходимо проведение предварительной обработки данных, которая является необходимым этапом процесса ИАД. Данные, полученные в результате сбора, должны соответствовать определенным критериям качества. Таким образом, можно выделить важный подэтап процесса ИАД - оценивание качества данных. Качество данных - это критерий, определяющий полноту, точность, своевременность и возможность интерпретации данных.

Данные могут быть высокого качества и низкого качества, последние - это так называемые грязные или "плохие" данные.

Данные высокого качества - это полные, точные, своевременные данные, которые поддаются интерпретации.

Такие данные обеспечивают получение качественного результата: знаний, которые смогут поддерживать процесс принятия решений.

Данные низкого качества, или грязные данные - это отсутствующие, неточные или бесполезные данные с точки зрения практического применения (например, представленные в неверном формате, не соответствующем стандарту). Грязные данные появились одновременно с системами ввода данных.

Грязные данные могут появиться по разным причинам, таким как ошибка при вводе данных, использование иных форматов представления или единиц измерения, несоответствие стандартам, отсутствие своевременного обновления, неудачное обновление всех копий данных, неудачное удаление записей-дубликатов и т.д. Необходимо оценить стоимость наличия грязных данных; другими словами, наличие грязных данных может действительно привести к финансовым потерям и юридической ответственности, если их присутствие не предотвращается или они не обнаруживаются и не очищаются. Основные причины появления грязных данных:

данные, которые могут быть автоматически обнаружены и очищены;

данные, появление которых может быть предотвращено;

данные, которые непригодны для автоматического обнаружения и очистки;

данные, появление которых невозможно предотвратить.

Поэтому важно понимать, что специальные средства очистки могут справиться не со всеми видами грязных данных.

Наиболее распространенные виды грязных данных:

пропущенные значения;

дубликаты данных;

шумы и выбросы.

Некоторые значения данных могут быть **пропущены** в связи с тем, что: данные вообще не были собраны; некоторые атрибуты могут быть неприменимы для некоторых объектов.

Методы борьбы с пропущенными значениями:

исключить объекты с пропущенными значениями из обработки;

рассчитать новые значения для пропущенных данных;

игнорировать пропущенные значения в процессе анализа;

заменить пропущенные значения на возможные значения.

**Дубликатами** называются записи с одинаковыми значениями всех атрибутов.

Наличие дубликатов в наборе данных может являться способом повышения значимости некоторых записей. Такая необходимость иногда возникает для особого выделения определенных записей из набора данных. Однако в большинстве случаев, продублированные данные являются результатом ошибок при подготовке данных.

Существует два варианта обработки дубликатов. При первом варианте удаляется вся группа записей, содержащая дубликаты. Этот вариант используется в том случае, если наличие дубликатов вызывает недоверие к информации, полностью ее обесценивает.

Второй вариант состоит в замене группы дубликатов на одну уникальную запись.

**Шумы и выбросы** являются достаточно общей проблемой в анализе данных. Выбросы могут как представлять собой отдельные наблюдения, так и быть объединенными в некие группы. Задача аналитика - не только их обнаружить, но и оценить степень их влияния на результаты дальнейшего анализа. Если выбросы



являются информативной частью анализируемого набора данных, используют робастные методы и процедуры.

Достаточно распространена практика проведения двухэтапного анализа - с выбросами и с их отсутствием - и сравнение полученных результатов.

Различные методы ИАД имеют разную чувствительность к выбросам, этот факт необходимо учитывать при выборе метода анализа данных. Также некоторые инструменты ИАД имеют встроенные процедуры очистки от шумов и выбросов.

Очевидно, что результаты ИАД на основе грязных данных не могут считаться надежными и полезными. Однако наличие таких данных не обязательно означает необходимость их очистки или же предотвращения появления. Всегда должен быть разумный выбор между наличием грязных данных и стоимостью и/или временем, необходимым для их очистки.

**Очистка данных** занимается выявлением и удалением ошибок и несоответствий в данных с целью улучшения качества данных.

Проблемы с качеством встречаются в отдельных наборах данных - таких как файлы и базы данных. Когда интеграции подлежат множество источников данных, необходимость в очистке данных существенно возрастает. Это происходит оттого, что источники часто содержат разрозненные данные в различном представлении. Для обеспечения доступа к точным и согласованным данным необходима консолидация различных представлений данных и исключение дублирующейся информации. Специальные средства очистки обычно имеют дело с конкретными областями - в основном это имена и адреса - или же с исключением дубликатов. Преобразования обеспечиваются либо в форме библиотеки правил, либо пользователем в интерактивном режиме. Преобразования данных могут быть автоматически получены с помощью средств согласования схемы. Метод очистки данных должен удовлетворять ряду критериев.

Он должен выявлять и удалять все основные ошибки и несоответствия, как в отдельных источниках данных, так и при интеграции нескольких источников.

Метод должен поддерживаться определенными инструментами, чтобы сократить объемы ручной проверки и программиро-

вания, и быть гибким в плане работы с дополнительными источниками.

Очистка данных не должна производиться в отрыве от связанных со схемой преобразования данных, выполняемых на основе сложных метаданных.

Функции маппирования для очистки и других преобразований данных должны быть определены декларативным образом и подходить для использования в других источниках данных и в обработке запросов.

Инфраструктура технологического процесса должна особенно интенсивно поддерживаться для ХД, обеспечивая эффективное и надежное выполнение всех этапов преобразования для множества источников и больших наборов данных.

#### **Этапы очистки данных**

В целом, очистка данных включает следующие этапы:

анализ данных;

определение порядка и правил преобразования данных;

подтверждение;

преобразования;

противоток очищенных данных.

#### **Этап 4. Построение модели**

После этого отступления займемся снова этапами процесса ИАД. После окончания этапа подготовки данных можно переходить к построению модели.

С одной стороны, можно говорить, что построенная модель выделила наиболее существенные (или значимые) факторы с точки зрения решаемой задачи. Для решения задачи классификации наиболее значимыми оказались переменные "доход" и "семейное положение", остальные факторы (т.е. остальные показатели исследуемой базы данных), сколько бы их ни было, оказались мало важными и не были включены в модель.

С другой стороны, данная модель, как и любая другая, может обладать свойством неполноты. Примером неучтенного фактора могут быть, скажем, природные катаклизмы, которые повлияли на желание клиента пользоваться услугами туристического агентства.

Для построения моделей используются различные методы и алгоритмы ИАД. Некоторые задачи могут быть решены при помощи моделей, построенных на основе различных методов. Идеальной модели, которая бы позволила решать разнообразные задачи, не существует. Поэтому многие разработчики включают в инструменты ИАД возможность построения различных моделей, многие также обеспечивают возможность расширяемости моделей. Некоторые инструменты ИАД создаются специально для конкретных областей применения.

Среди большого разнообразия методов ИАД должен быть выбран метод или же комбинация методов, при использовании которых построенная модель будет наилучшим образом описывать исследуемый объект.

Иногда для выявления искомым закономерностей требуется использование нескольких методов и алгоритмов. В таком случае одни методы используются в начале моделирования, другие - на дальнейших этапах. Пример: для определения однотипных групп клиентов применялся один из методов кластеризации, в результате клиенты были разбиты на группы, каждой из которых, присвоен код; далее мы пользовались методом деревьев решений. Код группы (результат работы предыдущего метода) использовался для интерпретации полученных закономерностей.

Выбор метода, на основе которого будет построена модель, должен осуществляться с учетом постановки задачи, особенностей набора исходных данных, специфики решаемой задачи, результатов, которые должны быть получены на выходе.

Постановка задачи формализует суть задачи, так, наличие входных и выходных переменных при решении задачи классификации определяет выбор одного из методов "обучение с учителем". Наличие лишь входных переменных определяет выбор другого - метода "обучение без учителя".

### **Этап 5. Проверка и оценка моделей**

Проверка модели подразумевает проверку ее достоверности или адекватности. Эта проверка заключается в определении степени соответствия модели реальности. Адекватность модели проверяется путем тестирования.

Адекватность модели - соответствие модели моделируемому объекту или процессу.

Понятия достоверности и адекватности являются условными, поскольку мы не можем рассчитывать на полное соответствие модели реальному объекту, иначе это был бы сам объект, а не модель. Поэтому в процессе моделирования следует учитывать адекватность не модели вообще, а именно тех ее свойств, которые являются существенными с точки зрения проводимого исследования. В процессе проверки модели необходимо установить включение в модель всех существенных факторов. Сложность решения этой проблемы зависит от сложности решаемой задачи.

Проверка модели также подразумевает определение той степени, в которой она действительно помогает менеджеру при принятии решений.

Оценка модели подразумевает проверку ее правильности. Оценка построенной модели осуществляется путем ее тестирования.

Тестирование модели заключается в "прогонке" построенной модели, заполненной данными, с целью определения ее характеристик, а также в- проверке ее работоспособности. Тестирование модели включает в себя проведение множества экспериментов. На вход модели могут подаваться выборки различного объема. С точки зрения статистики, точность модели увеличивается с увеличением количества исследуемых данных. Алгоритмы, являющиеся основой для построения моделей на сверхбольших базах данных, должны обладать свойством масштабирования.

Если модель достаточно сложна, а значит, требуется много времени на ее обучение и последующую оценку, то иногда бывает можно построить и протестировать модель на небольшой части выборки. Однако этот вариант подходит только для однородных данных, в противном случае необходимо использовать все доступные данные. Построенные модели рекомендуется тестировать на различных выборках для определения их обобщающих способностей. В ходе экспериментов можно варьировать объем выборки (количество записей), набор входных и выходных переменных, использовать выборки различной сложности.

Выявленные соотношения и закономерности должны быть проанализированы экспертом в предметной области - он поможет

определить, как являются выясненные закономерности (возможно, слишком общими или узкими и специфическими).

Для оценки результатов полученных моделей следует использовать знания специалистов предметной области. Если результаты полученной модели эксперт считает неудовлетворительными, следует вернуться на один из предыдущих шагов процесса ИАД, а именно: подготовка данных, построение модели, выбор модели.

Если же результаты моделирования эксперт считает приемлемыми, ее можно применять для решения реальных задач.

### **Этап 6. Выбор модели**

Если в результате моделирования нами было построено несколько различных моделей, то на основании их оценки мы можем осуществить выбор лучшей из них. В ходе проверки и оценки различных моделей на основании их характеристик, а также с учетом мнения экспертов, следует выбор наилучшей. Достаточно часто это оказывается непростой задачей.

Основные характеристики модели, которые определяют ее выбор, - это точность модели и эффективность работы алгоритма.

В некоторых программных продуктах реализован ряд методов, разработанных для выбора модели. Многие из них основаны на так называемой "конкурентной оценке моделей", которая состоит в применении различных моделей к одному и тому же набору данных и последующем сравнении их характеристик.

### **Этап 7. Применение модели**

После тестирования, оценки и выбора модели следует этап применения модели. На этом этапе выбранная модель используется применительно к новым данным с целью решения задач, поставленных в начале процесса ИАД. Для классификационных и прогнозирующих моделей на этом этапе прогнозируется целевой (выходной) атрибут.

### **Этап 8. Коррекция и обновление модели**

По прошествии определенного установленного промежутка времени с момента начала использования модели ИАД следует

проанализировать полученные результаты, определить, действительно ли она "успешна" или же возникли проблемы и сложности в ее использовании.

Однако даже если модель с успехом используется, ее не следует считать абсолютно верной на все времена. Дело в том, что необходимо периодически оценивать адекватность модели набору данных, а также текущей ситуации (следует учитывать возможность изменения внешних факторов). Даже самая точная модель со временем перестает быть таковой. Для того чтобы построенная модель выполняла свою функцию, следует работать над ее коррекцией (улучшением). При появлении новых данных требуется повторное обучение модели. Этот процесс называют обновлением модели. Работы, проводимые с моделью на этом этапе, также называют контролем и сопровождением модели.

Существует много причин, требующих обучить модель заново, т.е. обновить ее, чтобы отразить определенные изменения.

Основными причинами являются следующие:

- изменились входящие данные или их поведение;
- появились дополнительные данные для обучения;
- изменились требования к форме и количеству выходных данных;

- изменились цели бизнеса, которые повлияли на критерии принятия решений;

- изменилось внешнее окружение или среда (макроэкономика, политическая ситуация, научно-технический прогресс, появление новых конкурентов и товаров и т.д.).

Причины, перечисленные выше, могут обесценить допущения и исходную информацию, на которых основывалась модель при построении.

## **5.5. Организация создания и внедрения ИАД**

Организационные факторы это деловая окружающая среда, человеческие факторы.

Знания о предметной области включают факты, которые к данной области относятся, закономерности, характерные для нее, гипотезы о возможных связях между явлениями, процессами и фактами в ней, процедуры для решения типовых задач. Эксперт-

ные знания - это те знания, которыми располагает специалист в некоторой предметной области.

Администратор баз данных - специалист, имеющий знания о том, где и каким образом хранятся данные, как получить к ним доступ и как связать между собой эти данные.

Специалист по добыче данных - специалист по анализу данных, который имеет, как минимум, основы статистических знаний и может осуществлять постановку задач.

Дополнительные специалисты, вовлекаемые в ИАД:

менеджер проектов;

специалист по IT архитектуре;

специалист по архитектуре решений;

специалист по архитектуре данных;

специалист по моделированию данных;

эксперт ИАД;

деловой аналитик.

Инструменты ИАД можно оценивать по различным критериям. Оценка программных средств ИАД с точки зрения конечного пользователя определяется путем оценки набора его характеристик. Их можно поделить на две группы: бизнес-характеристики и технические характеристики. Это деление является достаточно условным, и некоторые характеристики могут попадать одновременно в обе категории.

**Характеристика № 1.** Интуитивный интерфейс.

Интерфейс - среда передачи информации между программной средой и пользователем, диалоговая система, которая позволяет передать человеку все необходимые данные, полученные на этапе формализации и вычисления.

Интерфейс подразумевает расположение различных элементов, в т.ч. блоков меню, информационных полей, графических блоков, блоков форм, на экранных формах.

Для удобства работы пользователя необходимо, чтобы интерфейс был интуитивным.

Интуитивный интерфейс позволяет пользователю легко и быстро воспринимать элементы интерфейса, благодаря чему диалог "программная среда-пользователь" становится проще и доступней.

Понятие интуитивного интерфейса включает также понятие знакомой окружающей среды и наличие внятной нетехнической терминологии (например, для сообщения пользователю о совершенной ошибке).

**Характеристика № 2. Удобство экспорта/импорта данных.**

При работе с инструментом ИАД-пользователь часто применяет разнообразные наборы данных, работает с различными источниками данных. Это могут быть текстовые файлы, файлы электронных таблиц, файлы баз данных. Инструмент ИАД должен иметь удобный способ загрузки (импорта) данных. По окончании работы пользователь также должен иметь удобный способ выгрузки (экспорта) данных в удобную для него среду. Программа должна поддерживать наиболее распространенные форматы данных.

Дополнительное удобство для пользователя создается при возможности загрузки и выгрузки определенной части (по выбору пользователя) импортируемых или экспортируемых полей.

**Характеристика № 3. Наглядность и разнообразие получаемой отчетности**

Эта характеристика подразумевает получение отчетности в терминах предметной области, а также в качественно спроектированных выходных формах в том количестве, которое может предоставить пользователю всю необходимую результативную информацию.

**Характеристика № 4. Легкость обучения работы с инструментарием**

**Характеристика № 5. Прозрачные и понятные шаги ИАД-процесса**

**Характеристика № 6. Руководство пользователя.** Существенно упрощает работу пользователя наличие руководства пользователя, с пошаговым описанием шагов генерации моделей ИАД.

**Характеристика № 7. Удобство и простота использования.** Существенно облегчает работу начинающего пользователя возможность использовать Мастер.

**Характеристика № 8.** Для пользователей, не владеющих английским языком, важной характеристикой является **наличие**



русифицированной версии инструмента, а также документации на русском языке.

**Характеристика № 9. Наличие демонстрационной версии** с решением конкретного примера.

**Характеристика № 10. Возможности визуализации.** Наличие графического представления информации существенно облегчает интерпретируемость полученных результатов.

**Характеристика № 11. Наличие значений параметров, заданных по умолчанию.** Для начинающих пользователей - это достаточно существенная характеристика, так как при выполнении многих алгоритмов от пользователя требуется задание или выбор большого числа параметров. Особенно много их в инструментах, реализующих метод нейронных сетей. В нейросимуляторах чаще всего заранее заданы значения основных параметров, иной раз неопытным пользователям даже не рекомендуется изменять эти значения. Если же такие значения отсутствуют, пользователю приходится перепробовать множество вариантов, прежде чем получить приемлемый результат.

**Характеристика № 12. Количество реализуемых методов и алгоритмов.** Во многих инструментах Data Mining реализовано сразу несколько методов, позволяющих решать одну или несколько задач. Если для решения одной задачи (классификации) предусмотрена возможность использования нескольких методов (деревьев решений и нейронных сетей), пользователь получает возможность сравнивать характеристики моделей, построенных при помощи этих методов.

**Характеристика № 13. Скорость вычислений и скорость представления результатов.**

**Характеристика № 14. Наличие квалифицированного ассистента** (консультации по выбору методов и алгоритмов), консультационная поддержка.

**Характеристика № 15. Возможности поиска, сортировки, фильтрации.**

Такая возможность полезна как для входных данных, так и для выходной информации. Применяется сортировка по различным критериям (полям), с возможностью накладывания условий.

При условии фильтрации входных данных появляется возможность построения модели ИАД на одной из выборок набора

данных. Необходимость и польза от проведения такого анализа была описана в одной из лекций, посвященных процессу Data Mining. Фильтрация выходной информации полезна с точки зрения интерпретации результатов. Так, например, иногда при построении деревьев решений результаты получаются слишком громоздкими, и здесь могут оказаться полезными функция как фильтрации, так и поиска и сортировки. Дополнительное удобство для пользователя - цветовая подсветка некоторых категорий записей.

**Характеристика № 16. Защита, пароль.** Очень часто при помощи Data Mining анализируется конфиденциальная информация, поэтому наличие пароля доступа в систему является желательной характеристикой для инструмента.

**Характеристика № 17. Платформы,** на которых поддерживается работа инструмента.

### **Контрольные вопросы**

1. Дать понятие ИАД.
2. Что такое хранилище данных.
3. Виды структур хранилищ данных.
4. Основные методы ИАД.

## Тестовые задания

1. Охарактеризуйте понятие «Знания»:  
могут быть как активны, так и пассивны;
  - всегда активны;
  - всегда пассивны.
2. Программы искусственного интеллекта отличаются от всех остальных:
  - способностью к самообучению;
  - языком, на котором написаны;
  - быстродействием.
3. Основной теоретической проблемой искусственного интеллекта является:
  - проблема представления знаний;
  - компьютерная логика;
  - разработка компьютерных методов и алгоритмов;
  - распознавание образов.
4. Какую информационную систему можно назвать интеллектуальной:
  - направленную на решение неформализованных (трудно формализуемых) задач;
  - направленную на решение задач, структура решения которых неизвестна;
  - направленную на решение хорошо структурированных задач.
5. Что такое мышление:
  - способность делать выводы;
  - процедуры, позволяющие системе вводить информацию;
  - совокупность процедур, работающих строго в рамках знаний.
6. Наука, изучающая знаковые системы, называется:
  - семантика;
  - семиотика;

## Заключение

### Мультиагентные системы

Поскольку общепринятого определения интеллектуального агента (ИА) до настоящего времени не выработано, то следует говорить о классе агентных объектов (сущностей), включающем множество видов. ИА – это программный или аппаратный объект (сущность), автономно функционирующий для достижения целей, поставленных перед ним владельцем или пользователем, обладающий определенными интеллектуальными способностями (рис. 3.1).

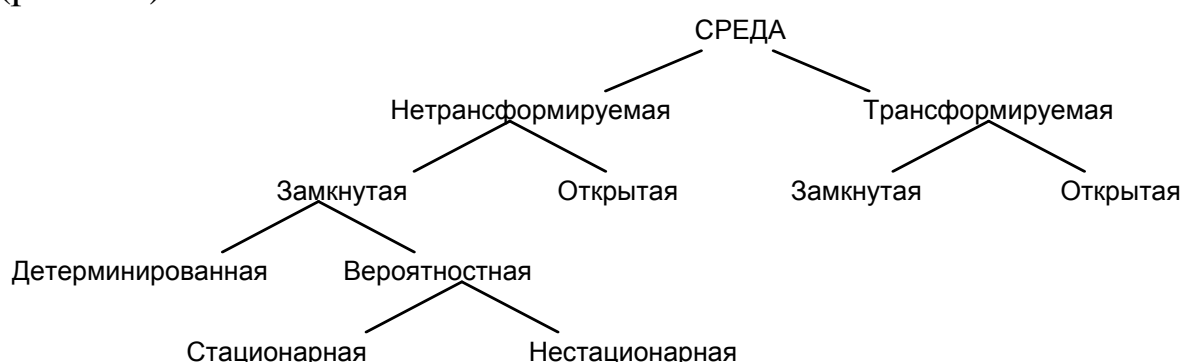


Рис. 3.1. Классификация сред

Для замкнутых сред может быть построено конечное исчерпывающее описание, и функционирующие в таких средах агенты могут обладать полным знанием о среде и ее свойствах или получить эту информацию в процессе своего взаимодействия со средой. Трансформируемые среды могут изменять свои характеристики и реакции на действия агентов в зависимости от тех действий, которые агенты совершают в среде. Вид математического аппарата, позволяющего описать поведение агента в соответствующей среде, и является мерой его интеллектуальной сложности (или разумности).

Две базовые характеристики – автономность и целенаправленность – позволяют отличать ИА от других программных и аппаратных объектов (модулей, подпрограмм, процедур и т. п.). Наличие целесообразности поведения требует, чтобы ИА обладал свойством реактивности. Такой уровень интеллекта соответствует рефлексорному поведению животного. Если же ИА обладает

знаниями о среде, собственных целях и способах их достижения, то такой агент может быть назван разумным (когнитивным). Таким образом, может быть проведена граница между интеллектуальными и неинтеллектуальными агентами.

К настоящему времени сформировался довольно большой список свойств, которыми должны обладать ИА:

автономность – способность к самостоятельному формированию целей и функционированию с самоконтролем своих действий и внутреннего состояния;

общественное поведение – способность согласовать свое поведение с поведением других агентов в условиях определенной среды и правил поведения путем обмена сообщениями на языке коммуникации;

реактивность – способность адаптированно воспринимать состояние внешней среды (среды функционирования и множества других агентов) и своевременно реагировать на происходящие изменения;

активность – способность проявлять инициативу, самостоятельно генерировать цели и действовать рационально для их достижения, а не только пассивно реагировать на внешние события;

базовые знания – постоянная часть знаний агента о себе, о среде, а также постоянные знания о других агентах, которые не изменяются в рамках жизненного цикла агента;

убеждения – переменная часть знаний агента о среде и других агентах, которая может изменяться во времени, но агент может об этом не знать и продолжать использовать их;

желания – состояния и/или ситуации, достижение которых является желательным и важным для агента, однако которые могут быть противоречивыми и не все будут достигнуты;

цели – совокупность состояний, на достижение которых направлено текущее поведение агента;

намерения – это то, что агент обязан сделать в силу своих обязательств по отношению к другим агентам, или то, что вытекает из его желаний (т.е. непротиворечивое подмножество желаний, выбранное по тем или иным причинам и совместимое с принятыми на себя обязательствами);

обязательства – задачи, которые берет на себя агент по просьбе и/или поручению других агентов.

К этому набору свойств могут добавляться следующие:

благжелательность – готовность агентов помогать друг другу и решать именно те задачи, которые им поручат владелец или пользователь;

правдивость – свойство агента не оперировать заведомо ложной информацией;

рациональность – способность агента действовать так, чтобы достигать своих целей, а не избегать их достижения, по крайней мере, в рамках своих знаний и убеждений.

Введенные таким образом необходимые свойства интеллектуальных агентов требуют некоторого обсуждения, так как на искусственный технический объект переносятся социальные свойства, обычно присущие поведению человека.

Общественное поведение агентов может принимать разные формы, которые могут быть классифицированы по уровням взаимодействия:

Уровень 0 – связность, устанавливаемая извне владельцем или пользователем и не воспринимаемая самими агентами;

Уровень 1 – координация. Агенты способны создать ситуацию, позволяющую другим агентам оказаться в нужном месте в нужное время, чтобы в результате их деятельность осуществлялась эффективным образом.

Уровень 2 – кооперация. Агенты допускают, чтобы их поведение частично определялось поведением других агентов, когда они совместно пытаются достичь некоторой общей цели. Такой процесс для своей реализации должен осознаваться всеми участвующими в нем агентами.

Уровень 3 – сотрудничество. Предполагается реальная совместная работа агентов, в процессе выполнения которой может выиграть каждый.

Уровень 4 – образование союза. Предусматривается продолжительная во времени совместная деятельность, в ходе которой агенты создают и поддерживают условия существования союза.

Базовые знания являются необходимым традиционным компонентом для всех интеллектуальных систем, убеждения же должны быть определенным образом интерпретированы в структуре МАС.

Интеллектуальная система (агент) может воспринимать как истинные правила формирования выводов, базовые шкалы и веса критериев, функции или отношения предпочтения и т.д.

Убеждения можно разделить на три класса.

1. Внутренние убеждения агента. Это алгоритмы, сценарии, оценки, заложенные в него при разработке или внесенные в процессе эксплуатации владельцем или пользователем.

2. Индуктивные убеждения, возникающие в результате анализа состояний среды, формируемые производственными правилами.

3. Коммуникативные убеждения, возникающие в результате связи с другими агентами, формируемые производственными правилами.

Приписывание ИА желаний и намерений, по мнению авторов, не является бесспорным, так как в реальных технических системах они могут вступить в противоречие с целями и обязательствами, предусмотренными для них человеком-разработчиком, и привести к потере управляемости и непредсказуемому поведению систем.

В последние годы термин МАС применяют к широкому спектру программных систем, состоящих из различных автономных и полуавтономных компонентов. МАС должны обладать 4 важнейшими свойствами: ситуативности, автономности, гибкости и социальности.

Ситуативность ИА понимается как способность воспринимать окружающую его среду (окружение) и действовать в этой среде, при возможности изменяя ее в своих целях.

Автономность ИА означает его способность взаимодействовать со средой без прямого участия других агентов, для чего он должен уметь контролировать свое внутреннее состояние и выполняемые действия.

Гибкий агент должен демонстрировать качества отзывчивости или предусмотрительности (в зависимости от ситуации). Отзывчивый агент получает стимулы от своего окружения и вовремя отвечает на них соответствующим образом. Предусмотрительный агент не просто реагирует на ситуацию в окружающей среде, но и адаптируется, целенаправленно действует и выбирает альтернативы в различных ситуациях.

Агент обладает свойством социальности, если он может соответствующим образом взаимодействовать с другими программными или человеческими агентами. ИА являются лишь составляющими сложного процесса решения проблем в соответствующей среде. Поведение социального агента и его взаимодействие с другими агентами и средой.

### **Основные направления исследований и архитектуры**

Основными направлениями научного поиска в области МАС являются:

теории агентов, которые рассматривают математические методы и формализмы абстрактного представления структуры и свойств агентов и способы построения рассуждений (логических выводов) в таких формальных системах;

методы коллективного поведения агентов;

архитектуры агентов и МАС;

методы, языки и средства коммуникации агентов;

языки программирования агентов;

методы и средства автоматизированного проектирования;

методы и средства обеспечения мобильности агентов.

В разработке логических формализмов ментальных понятий возникают две основные проблемы: представление синтаксиса языка формализации и создание его семантической модели.

К проблеме описания синтаксиса существует два фундаментальных подхода: применение модальных логик, содержащих не истинностные модальные операторы; применение метаязыков, основанных на многосортной логике первого порядка, содержащей термины, которые определяют формулы некоторого другого объектного языка.

Формализация знаний и убеждений в модальных логических системах приводит к тому, что агенту должны быть известны все общезначимые формулы (пропозициональные тавтологии), а поскольку количество таких формул бесконечно, то такие модели оказываются неприменимыми в реальных технических системах, так как все они имеют ограниченные ресурсы.

Использование метаязыков позволяет представить отношения между терминами метаязыка, определяющего агента и терминами объектного языка, определяющего некоторые формулы.



Недостатком формализма метаязыков является противоречивость некоторых из них, тем не менее, существуют и достаточно успешные метаязыковые конструкции.

К настоящему времени сложилась традиция выделения трех базовых классов архитектур агентных систем и соответствующих им моделей интеллектуальных агентов: делиберативные архитектуры и модели; реактивные архитектуры и модели; гибридные архитектуры и модели.

Делиберативную архитектуру принято определять как архитектуру агентов, содержащих точную символическую модель мира и принимающих решения на основе логического вывода. Делиберативный агент должен обладать следующими свойствами:

содержать эксплицитно представленную базу знаний, заполненную формулами в некотором логическом языке, представляющую его убеждения;

функционировать в следующем цикле: восприятие обстановки (обсервация) – логический вывод – действие ...;

принимать решения о действиях на основе некоторых форм логического вывода.

Поиски путей разрешения проблем, возникающих при использовании в агентных системах классических методов ИИ, привели к появлению нового класса – реактивных архитектур: интеллектуальное поведение может создаваться без явного символического представления знаний; интеллектуальное поведение может создаваться без явного абстрактного логического вывода; интеллект является внезапно возникающим свойством некоторых сложных систем.

В реальном мире интеллект не является экспертной системой или машиной логического вывода, а интеллектуальное поведение возникает как результат взаимодействия агента со средой. Реактивный подход позволяет эффективно использовать множество довольно простых сценариев поведения агентов в рамках установленных реакций на определённые события окружающей среды, но его ограниченность проявляется в практической невозможности полного ситуативного анализа всех возможных активностей агентов. Поэтому в большинстве проектов и действующих систем используются гибридные архитектуры, спектр вариантов построения которых достаточно широк.

## Список литературы

1. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Финансы и статистика – М.: 2004.
2. Барский А. Б. Нейронные сети: распознавание, управление, принятие решений. - М.: Финансы и статистика, 2004. — 176 с.
3. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии: Учеб. пособие. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. - 304 с.
4. Братко И. Программирование на языке Prolog для искусственного интеллекта. – М.: Мир, 1990.
5. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2000. – 384 с.
6. Карминский А.М., Черников Б.В. Информационные системы в экономике: В 2-х ч. Ч. 1. Методология создания: Учеб. пособие. - М.: Финансы и статистика, 2006. - 336 с.
7. Лисс А.А. Искусственные нейронные сети: Учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2003. - 96 с.
8. Мозговой М.В. Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход. - СПб.: Наука и техника, 2006. - 320 с.
9. Рассел Стюарт, Норвиг Питер. Искусственный интеллект: современный подход. - М.: Издательский дом «Вильямс», 2006. - 1408 с.
10. Романов В.П. Интеллектуальные информационные системы в экономике. – М.: Экзамен, 2003. – 496 с.
11. Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф. Проектирование экономических информационных систем. - М.: Финансы и статистика, 2003. - 512 с.
12. Смолин Д.В. Введение в искусственный интеллект. – М.: Физматлит, 2004.
13. Хювнен Э., Сеппянен Й. Мир ЛИСПа. Т. 1. Введение в язык ЛISP и функциональное программирование. — М.: Мир, 1990. – 447 с.
14. Хювнен Э., Сеппянен Й. Мир ЛИСПа. Т. 2. Методы и системы программирования. — М.: Мир, 1990. – 319 с.

## Терминологический словарь

**Аналитические технологии** – обобщающий термин, применяемый для обозначения подходов к комплексному использованию ряда формальных и эвристических методов и информационных технологий.

**Знания** – это совокупность сведений о сущностях (объектах, предметах) реального мира, их свойствах и отношениях между ними в определенной предметной области.

**Интеллектуальные информационные системы** – технические и программные системы, ориентированные на решение большого и очень важного класса задач, называемых неформализованными.

**Модель представления знаний** - это способ и результат формального описания знаний в базе знаний.

**Нейронные вычисления** – это теория разработки и исследования систем обработки информации, использующих механизмы восприятия и переработки информации естественных информационных систем.

**Хранилище данных** – предметно-ориентированный, интегрированный, неизменчивый, поддерживающий хронологию набор данных, организованный для целей поддержки управления.

**Экспертная система** – это программное средство, использующее знания экспертов, для высокоэффективного решения задач в интересующей пользователя проблемной области.

## Извлечение из рабочей программы дисциплины

### 4. Содержание разделов и тем дисциплины

#### Введение

Предмет дисциплины и ее задачи. Основные понятия искусственного интеллекта. Информационные системы, имитирующие творческие процессы.

Понятие интеллектуализации ЭВМ. Искусственный интеллект, технологии искусственного интеллекта. *Основные понятия искусственного интеллекта.* Понятие ИИС, как *информационной системы, имитирующей творческие процессы*, ее основные свойства и место в процессе принятия экономических и управленческих решений. Классификация ИИС. Формализация решения задач.

Структура и содержание дисциплины, ее связь с другими дисциплинами учебного плана и место в подготовке студентов СПбГИЭУ.

#### Тема 1. Модели представления знаний в ИИС

##### 1.1. Понятие «знание» в искусственном интеллекте

*Информация и данные. Информационные модели знаний.*

Понятие «знание», основные свойства знаний и их отличие от данных. Понятие базы знаний и ее отличие от базы данных. Структура и классификация систем, основанных на знаниях. Классификация знаний. *Интеллектуальные информационно-поисковые системы. Основные методы инженерии знаний.*

Организация базы знаний. Декларативная и процедурная формы представления знаний. Методы представления знаний.

##### 1.2. Основные модели представления знаний

Понятие модели представления знаний (МПЗ). Основные МПЗ, их особенности и области применения. Понятие вывода на знаниях.

*Методы представления знаний в базах данных информационных систем.*

##### 1.3. Логическая МПЗ

*Логико-лингвистические модели представления знаний.* Логика первого порядка как формальная основа логической МПЗ.

Метод Эрбрана и метод резолюций. Подстановки и унификации. Поиск ответов на вопросы.

#### 1.4. Продукционная МПЗ

*Системы интеллектуального интерфейса для информационных систем.*

Формальная грамматика как способ представления знаний в продукционной МПЗ. Понятие и форма записи правил продукции. Синтаксические деревья, задачи разбора и вывода. Конечный автомат как вычислительный формализм продукционной МПЗ.

#### 1.5. Фреймовая и сетевая МПЗ

*Фреймовые модели. Модель прикладных процедур, реализующих правила обработки данных.*

Понятие рекурсивной функции. S- и  $\lambda$ - выражения. Язык ЛИСП. Понятие фрейма и его реализация в символике ЛИСПа.

*Функциональная семантическая сеть. Определение и способ задания семантической сети как реализации интегрированного представления данных, категорий типов данных, свойств категорий и операций над данными и категориями. Вывод в семантических сетях.*

### **Тема 2. Экспертные системы**

#### 2.1. Основы построения экспертных систем

Концепция экспертной системы. Назначение и основные свойства. Обобщенная структура экспертной системы. Классификация экспертных систем и инструментальных средств их разработки.

Составные части экспертной системы: база знаний, механизм вывода, механизмы приобретения и объяснения знаний, интеллектуальный интерфейс.

Ограничения, присущие экспертным системам. Особенности экспертных систем экономического анализа.

Статические и динамические экспертные системы.

#### 2.2. Технология разработки экспертных систем

Состав и взаимодействие участников построения и эксплуатации ЭС. Принципы разработки. Этапы проектирования экспертной системы: идентификация, концептуализация, формализация, реализация, тестирование, опытная эксплуатация. Содер-

жание этапов проектирования. Участники процесса проектирования: эксперты, инженеры по знаниям, конечные пользователи.

Организация процесса приобретения и формализации знаний. Эксперт и инженер по знаниям: формы и порядок взаимодействия.

### 2.3. Разработка экспертных систем в среде ИНТЕРЭКСПЕРТ

Основные понятия системы ИНТЕРЭКСПЕРТ. Структура правила экспертной системы. Структура набора правил. Создание набора правил. Консультация с набором правил. Объяснение вывода.

## **Тема 3. Нечеткие знания**

### 3.1. Понятие нечеткости знаний.

Нечеткий вывод знаний. Немонотонность вывода. Обработка нечетких знаний с использованием коэффициентов уверенности. Обработка нечетких множеств.

### 3.2. Неопределенность в экспертных системах

Проблемы неопределенности в экспертных системах. Классификация методов обработки неопределенности знаний. Теория субъективных вероятностей. Байесовское оценивание. Теорема Байеса как основа управления неопределенностью.

### 3.3. Нечеткие множества

Нечеткие множества в системах, основанных на знаниях. Лингвистические шкалы и нечеткие переменные. Функции принадлежности. Арифметические операции над нечеткими переменными. Системы нечеткого вывода. Обработка нечетких знаний в среде ИНТЕРЭКСПЕРТ.

## **Тема 4. Нейронные сети**

### 4.1. Понятие нейронной сети

Биологические нейронные сети. Задачи, решаемые с помощью нейронных вычислений. История теории нейронных вычислений. Способы реализации.

### 4.2. Принципы организации и функционирования искусственных нейронных сетей

Классификация законов и способов обучения. Архитектуры искусственных нейронных сетей. Простой и однослойный персептрон. Классификация линейно разделимых образов. Обучение персептрона.

#### 4.3. Рекуррентные ассоциативные сети

Энергетическая функция рекуррентной сети. Сеть Хопфилда. Двухнаправленная ассоциативная память. Машина Больцмана.

#### 4.4. Сеть с обратным распространением ошибки

Задача преобразования данных. Закон обучения сети с обратным распространением ошибки. Проблемы обучения сетей преобразования данных. Применение многослойного персептрона для пространственно-временной обработки данных.

#### 4.5. Сеть Кохонена

Обучение и функционирование сети Кохонена. Модификация весовых коэффициентов. Процедура конкуренции. Алгоритм обучения сети Кохонена.

### **Тема 5. Интеллектуальный анализ данных**

#### 5.1. Понятие интеллектуального анализа данных

Место интеллектуального анализа данных (ИАД) в ИИС. Описано возникновение, перспективы, проблемы ИАД. Сферы применения ИАД. Технология ИАД как на часть рынка информационных технологий. OLTP, хранилища и витрины данных, OLAP.

#### 5.2. Методы и стадии ИАД

Основные стадии ИАД и действия, выполняемые в рамках этих стадий. Классификации методов ИАД. Сравнительная характеристика некоторых методов, основанная на их свойствах. Основная суть задач ИАД и их классификация. Сопоставление и сравнение понятий «информация» и «знания» в ИАД.

#### 5.3. Задачи ИАД

Классификация и кластеризация: суть, процесс решения, методы решения, применение. Прогнозирование и визуализация: понятие временного ряда, его компоненты, параметры прогнозирования, виды прогнозов, визуализация данных. Основы анализа данных. Деревья решений: элементы дерева решения, процесс его построения. Метод опорных векторов, метод «ближайшего соседа» и байесовской классификации. Их преимущества и недостатки.

#### 5.4. Процесс ИАД

Начальные этапы: процесс подготовки данных, понятия качества данных, грязных данных, этапы очистки данных. Очистка данных: классификации инструментов очистки и редактирования

данных, основные функции инструментов очистки данных, классификация ошибок в данных, возникающие в результате использования средств очистки данных. Построение и использование модели данных: построение, проверка, оценка, выбора и коррекция моделей.

#### 5.5. Организация создания и внедрения ИАД

Организационные факторы. Стандарты ИАД. ИАД-консалтинг, предоставление слуг по эффективному внедрению.

#### **Заключение**

*Тенденции развития теории искусственного интеллекта.*

Понятие интеллектуального агента. Основные направления исследований и архитектуры мультиагентных систем.

Перспективы развития ИИС и их применение в различных областях экономики и менеджмента.