

Система, информация, знания

Понятие информации - одно из основных, ключевых понятий не только в системном анализе, но и в информатике, математике, физике и др. В то же время, это понятие - плохо формализуемое, из-за его всеобщности, объемности, расплывчатости, и трактуется как:

- любая сущность, которая вызывает изменения в некоторой информационно-логической (инфологической - состоящей из сообщений, *данных, знаний*, абстракций, структурных схем и т.д.) модели, представляющей систему (математика, системный анализ);
- сообщения, полученные системой от внешнего мира в процессе адаптивного управления, приспособления (теория управления, кибернетика);
- отрицание энтропии, отражение меры хаоса в системе (термодинамика);
- связи и отношения, устраняющие неопределенность в системе (теория информации);
- вероятность выбора в системе (теория вероятностей);
- отражение и передача разнообразия в системе (физиология, биокибернетика);
- отражение материи, атрибут сознания, "интеллектуальности" системы (философия).

Мы будем рассматривать системное понимание этой категории, ничуть не отрицая приведенные выше понятия и, более того, используя их по мере надобности.

Процесс познания - это иерархическая система актуализации информации, в которой *знания* на каждом следующем уровне иерархии являются интегральным результатом актуализации *знаний* на предыдущем уровне. Это процесс интеграции информационных ресурсов, от получаемых с помощью простого чувственного восприятия и до сложных аксиоматических и абстрактных теорий.

Данные - синтаксические сигналы, образы, актуализируемые с помощью некоторого источника *данных*. Они рассматриваются безотносительно к семантическому их смыслу.

Информация - это некоторая последовательность сведений, *знаний*, которые актуализируемы (получаемы, передаваемы, преобразуемы, сжимаемы, регистрируемы) с помощью некоторых знаков символического, образного, жестового, звукового, сенсомоторного типа.

Информация - это *данные*, рассматриваемые с учетом некоторой их семантической сущности.

Знания - информация, обеспечивающая достижение некоторой цели и структуры.

Информация с мировоззренческой точки зрения - отражение реального мира. Информация - приращение, развитие, актуализация *знаний*, возникающее в процессе целеполагающей интеллектуальной деятельности человека.

Никакая информация, никакое знание не появляется сразу: появлению их предшествует этап накопления, систематизации опытных *данных*, мнений, взглядов, их осмысление и переосмысление. *Знание* - продукт этого этапа и такого системного процесса.

Информация (в системе, о системе) по отношению к окружающей среде (окружению) бывает трех типов: *входная, выходная* и внутренняя.

Входная информация - та, которую система воспринимает от окружающей среды. Такого рода информация называется *входной информацией* (по отношению к системе).

Выходная информация (по отношению к окружающей среде) - та, которую система выдает в окружающую среду.

Внутренняя, **внутрисистемная информация** (по отношению к данной системе) - та, которая хранится, перерабатывается, используется только внутри системы, актуализируется лишь подсистемами системы.

Пример. Человек воспринимает, обрабатывает *входную информацию*, например, *данные* о погоде на улице, формирует выходную реакцию - ту или иную форму одежды. При этом используется внутренняя информация, например, генетически заложенная или приобретенная физиологическая информация о реакции, например, о "морозостойкости" человека.

Внутренние состояния системы и структура системы влияют определяющим образом на взаимоотношения системы с окружающей средой - *внутрисистемная информация* влияет

на *входную* и *выходную информацию*, а также на изменение самой *внутрисистемной информации*.

Пример. Информация о финансовой устойчивости банка может влиять на его деятельность. Накапливаемая (внутрисистемно) социально-экономическая негативная информация (проявляемая, например, социальной активностью в среде) может влиять на развитие системы.

Пример. Генетически заложенная в молекулах ДНК информация и приобретенная информация (храняемая в памяти) влияют на поведение, на адаптацию человека к окружающей среде. В машинах первого поколения внутренняя структура определялась тысячами ламп, причем каждая из них отдельно была невысокой надежности, т.е. подобная система была ненадежной в работе. Это влияло на *входную информацию*, на функционирование системы, например, такие ЭВМ не были способны на работу в многозадачном режиме, в режиме реального времени (обработки сообщений по мере получения входных *данных*).

В живой и неживой природе информация может также передаваться некоторой структурой. Такую информацию называют (часто это условно можно отнести к информации) **структурной информацией**.

Пример. Структурные кольца среза дерева несут информацию о возрасте дерева. Структура питания хищника (или трофическая структура) несет информацию о хищнике, о среде его обитания. Структура плавников рыбы часто несет информацию о глубине среды ее обитания. Структура фирмы может объяснять многие явления и поведение фирмы.

Информация по отношению к конечному результату проблемы бывает:

- исходная (на стадии начала использования актуализации этой информации);
- промежуточная (на стадии от начала до завершения актуализации информации);
- результирующая (после использования этой информации, завершения ее актуализации).

Пример. При решении системы линейных алгебраических уравнений информация о методах решения, среде реализации, входных *данных* (источники, точность и т.д.), размерности системы и т.д. является исходной информацией; информация о совместности системы уравнений, численных значениях корня и т.д. - результирующая; информация о текущих состояниях коэффициентов уравнений, например, при реализации схемы Гаусса - промежуточная.

Информация (по ее изменчивости при актуализации) бывает:

- постоянная (не изменяемая никогда при ее актуализации);
- переменная (изменяемая при актуализации);
- смешанная - условно-постоянная (или условно-переменная).

Возможна также классификация информации и по другим признакам:

- по стадии использования (первичная, вторичная);
- по полноте (избыточная, достаточная, недостаточная);
- по отношению к цели системы (синтаксическая, семантическая, прагматическая);
- по отношению к элементам системы (статическая, динамическая);
- по отношению к структуре системы (структурная, относительная);
- по отношению к управлению системой (управляющая, советующая, преобразующая);
- по отношению к территории (федеральная, региональная, местная, относящаяся к юридическому лицу, относящаяся к физическому лицу);
- по доступности (открытая или общедоступная, закрытая или конфиденциальная);
- по предметной области, по характеру использования (статистическая, коммерческая, нормативная, справочная, научная, учебная, методическая и т.д., смешанная) и другие.

Информация в философском аспекте бывает:

- мировоззренческая;
- эстетическая;
- религиозная;
- научная;
- бытовая;

- техническая;
- экономическая;
- технологическая.

Все это (вместе с личностным аспектом человека как вида) составляет **ноосферу** общества - более высокое состояние биосферы, возникшее в результате эволюции, структурирования, упорядочивания (как статического, так и динамического) и гармонизации связей в природе и обществе под воздействием целеполагающей деятельности человечества. Это понятие было введено В.И. Вернадским в качестве отображения концепции этапа эволюции общества и природы, т.е. системы, в рамках которой потенциально может быть реализовано гармоническое, устойчивое развитие (эволюция) систем "Общество" и "Природа", а также постепенное слияние, гармонизация наук о природе и об обществе.

Основные **свойства информации** (и сообщений):

- полнота (содержит все необходимое для понимания информации);
- актуальность (необходимость) и значимость (сведений);
- ясность (выразительность сообщений на языке интерпретатора);
- адекватность, точность, корректность интерпретации, приема и передачи;
- интерпретируемость и понятность интерпретатору информации;
- достоверность (отображаемого сообщениям);
- избирательность;
- адресность;
- конфиденциальность;
- информативность и значимость (отображаемых сообщений);
- массовость (применимость ко всем проявлениям);
- кодируемость и экономичность (кодирования, актуализации сообщений);
- сжимаемость и компактность;
- защищенность и помехоустойчивость;
- доступность (интерпретатору, приемнику);
- ценность (предполагает достаточный уровень потребителя).

Информация может оказаться и вредной, влияющей негативно на сознание, например, воспитывающей восприятие мира от безразличного или же не критического - до негативного, "обозленного", неадекватного. Информационный поток - достаточно сильный раздражитель.

Пример. Негативной информацией могут быть сведения о крахе коммерческого банка, о резком росте (спаде) валютного курса, об изменении налоговой политики и др.

Информация не существует без других типов ресурсов: энергии, вещества, организации, как и они не могут существовать без информации. Любые взаимодействия систем (подсистем) - взаимодействия всегда материо-энерго-информационные. Выявление (систематизация, структурирование), описание (формализация), изучение, применение инвариантов этих взаимодействий и составляет основную задачу науки как человеческой деятельности.

Методы получения и использования информации можно разделить на три группы, иногда разграничиваемые лишь условно:

1. **эмпирические методы** или методы получения эмпирической информации (эмпирических данных);
2. **теоретические методы** или методы получения теоретической информации (построения теорий);
3. **эмпирико-теоретические методы** (смешанные, полуэмпирические) или методы получения эмпирико-теоретической информации.

Охарактеризуем кратко **эмпирические методы**:

1. Наблюдение - сбор первичной информации или эмпирических утверждений о системе (в системе).
2. Сравнение - установление общего и различного в исследуемой системе или системах.
3. Измерение - поиск, формулирование эмпирических фактов.

4. Эксперимент - целенаправленное преобразование исследуемой системы (систем) для выявления ее (их) свойств.

Кроме классических форм их реализации, в последнее время используются и такие формы как опрос, интервью, тестирование и другие.

Охарактеризуем кратко *эмпирико-теоретические методы*.

1. Абстрагирование - установление общих свойств и сторон объекта (или объектов), замещение объекта или системы ее моделью. Абстракция в математике понимается в двух смыслах: а) абстракция, абстрагирование - метод исследования некоторых явлений, объектов, позволяющий как выделить основные, наиболее важные для исследования свойства, стороны исследуемого объекта или явления, так и игнорировать несущественные и второстепенные; б) абстракция - описание, представление объекта (явления), получаемое с помощью метода абстрагирования; особо важно в информатике такое понятие как абстракция потенциальной осуществимости, которое позволяет нам исследовать конструктивно объекты, системы с потенциальной осуществимостью (т.е. они могли бы быть осуществимы, если бы не было ограничений по ресурсам); используются и абстракция актуальной бесконечности (существования бесконечных, неконструктивных множеств, систем и процессов), а также абстракция отождествления (возможности отождествления любых двух одинаковых букв, символов любого алфавита, объектов, независимо от места их появления в словах, конструкциях, хотя их информационная ценность при этом может быть различна).
2. Анализ - разъединение системы на подсистемы с целью выявления их взаимосвязей.
3. Декомпозиция - разъединение системы на подсистемы с сохранением их взаимосвязей с окружением.
4. Синтез - соединение подсистем в систему с целью выявления их взаимосвязей.
5. Композиция - соединение подсистем в систему с сохранением их взаимосвязей с окружением.
6. Индукция - получение *знания* о системе по *знаниям* о подсистемах; индуктивное мышление: распознавание эффективных решений, ситуаций и затем проблем, которые оно может разрешать.
7. Дедукция - получение *знания* о подсистемах по *знаниям* о системе; дедуктивное мышление: определение проблемы и затем поиск ситуации, его разрешающей.
8. Эвристики, использование эвристических процедур - получение *знания* о системе по *знаниям* о подсистемах системы и наблюдениям, опыту.
9. Моделирование (простое моделирование) и/или использование приборов - получение *знания* об объекте с помощью модели и/или приборов; моделирование основывается на возможности выделять, описывать и изучать наиболее важные факторы и игнорировать при формальном рассмотрении второстепенные.
10. Исторический метод - поиск *знаний* о системе путем использования ее предыстории, реально существовавшей или же мыслимой, возможной (виртуальной).
11. Логический метод - метод поиска *знаний* о системе путем воспроизведения ее некоторых подсистем, связей или элементов в мышлении, в сознании.
12. Макетирование - получение информации по макету объекта или системы, т.е. с помощью представления структурных, функциональных, организационных и технологических подсистем в упрощенном виде, сохраняющем информацию, которая необходима для понимания взаимодействий и связей этих подсистем.
13. Актуализация - получение информации с помощью активизации, инициализации смысла, т.е. переводом из статического (неактуального) состояния в динамическое (актуальное) состояние; при этом все необходимые связи и отношения (открытой) системы с внешней средой должны быть учтены (именно они актуализируют систему).
14. Визуализация - получение информации с помощью наглядного или визуального представления состояний актуализированной системы; визуализация предполагает возможность выполнения в системе операции типа "передвинуть", "вернуть",

"укрупнить", "уменьшить", "удалить", "добавить" и т.д. (как по отношению к отдельным элементам, так и к подсистемам системы). Это метод визуального восприятия информации.

Кроме указанных классических форм реализации теоретико-эмпирических методов, в последнее время часто используются и такие формы как мониторинг (система наблюдений и анализа состояний системы), деловые игры и ситуации, экспертные оценки (экспертное оценивание), имитация (подражание), верификация (сопоставление с опытом и заключение об обучении) и другие формы.

Охарактеризуем кратко *теоретические методы*.

1. Восхождение от абстрактного к конкретному - получение *знаний* о системе на основе *знаний* о ее абстрактных проявлениях в сознании, в мышлении.
2. Идеализация - получение *знаний* о системе или о ее подсистемах путем мысленного конструирования, представления в мышлении систем и/или подсистем, не существующих в действительности.
3. Формализация - получение *знаний* о системе с помощью знаков или же формул, т.е. языков искусственного происхождения, например, языка математики (или математическое, формальное описание, представление).
4. Аксиоматизация - получение *знаний* о системе или процессе с помощью некоторых, специально для этого сформулированных аксиом и правил вывода из этой системы аксиом.
5. Виртуализация - получение *знаний* о системе созданием особой среды, обстановки, ситуации (в которую помещается исследуемая система и/или ее исследующий субъект), которую реально, без этой среды, невозможно реализовать и получить соответствующие *знания*.

Эти методы получения информации системно применяются в любой сфере деятельности ([рис. 5.1](#)).

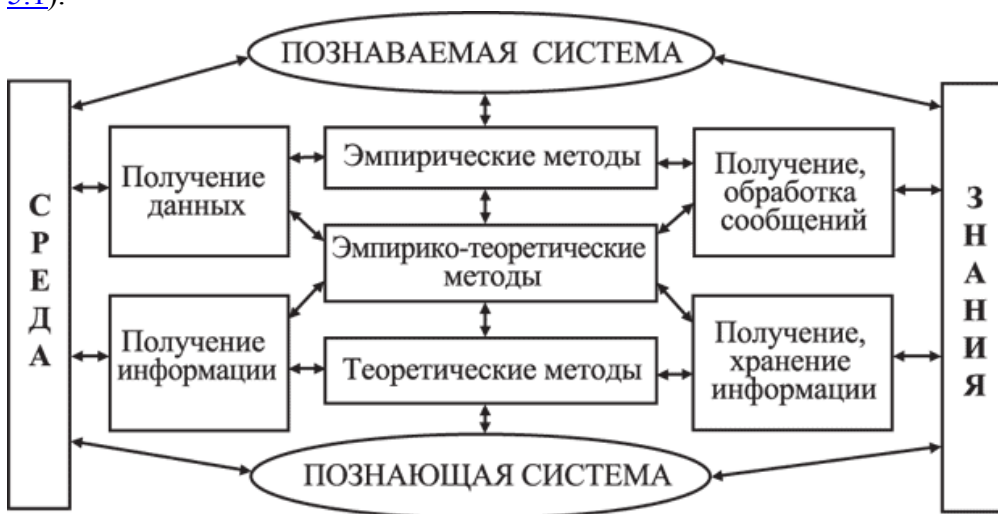


Рис. 5.1. Структура познания системы

Пример. Для построения модели планирования и управления производством в рамках страны, региона или крупной отрасли, нужно решить проблемы:

- определить структурные связи системы (как вертикальные, так и горизонтальные), уровни управления и принятия решений, ресурсы; при этом чаще используются методы наблюдения, сравнения, измерения, эксперимента, анализа и синтеза, дедукции и индукции, эвристический, исторический и логический, макетирование и др.;
- определить гипотезы, цели, возможные проблемы планирования; наиболее используемые методы: наблюдение, сравнение, эксперимент, абстрагирование, анализ, синтез, дедукция, индукция, эвристический, исторический, логический и др.;

- конструирование эмпирических моделей системы; наиболее используемые методы: абстрагирование, анализ, синтез, индукция, дедукция, формализация, идеализация и др.;
- поиск решения проблемы планирования и просчет различных вариантов, директив планирования, поиск оптимального решения; используемые чаще методы: измерение, сравнение, эксперимент, анализ, синтез, индукция, дедукция, актуализация, макетирование, визуализация, виртуализация и др.

Информация, таким образом, может быть рассмотрена как кортеж $A = \langle X, Y, f \rangle$, где носитель X - сведения, *знания* о предметной области, множество Y - сообщения, отражающие эти сведения, отношение f - отношение кодирования между элементами X, Y , т.е. их актуализации.

Пример. Пусть $X = \{\text{супруги, дети супругов}\}$, $Y = \{\text{"Иванов Петр Сидорович", "Иванова Ольга Николаевна", "Иванов Олег Петрович", "Иванова Наталья Петровна", "мать", "отец", "сын", "дочь", "родители", "дети"}\}$, отношение f может быть задано (словесно) перечислением связей вида: "Иванов Олег Петрович - супруг Ивановой Ольги Николаевны", "Иванова Наталья Петровна - дочь Ивановой Ольги Николаевны" и т.д.

Пример. Пусть $X = \{\text{арифметические операции}\}$, $Y = \{\text{"-(взятие противоположного числа)", "+(сложение)", "-(вычитание)", "\times(\text{умножение})", "/(\text{деление})", "\sqrt{\text{(извлечение квадратного корня)}}$, f определим как установление соответствия "унарная операция".

Таким образом, основная задача науки состоит в построении, исследовании, актуализации или хранении множеств с заданным классом X однотипных задач, Y - классом структур и ресурсов, связываемых с этими задачами, и f - процессами их сопоставления и актуализации с помощью некоторых ресурсов.

Такие задачи мы решаем в ежедневной жизни, но в то же время часто правило f нельзя отыскать или построить явно или конструктивно. В этом случае приходится заменять искомым закон f с помощью подходящих явных или конструктивных представлений f, X, Y и/или Z (см. [рис. 5.2](#)) и применять эти представления всякий раз.

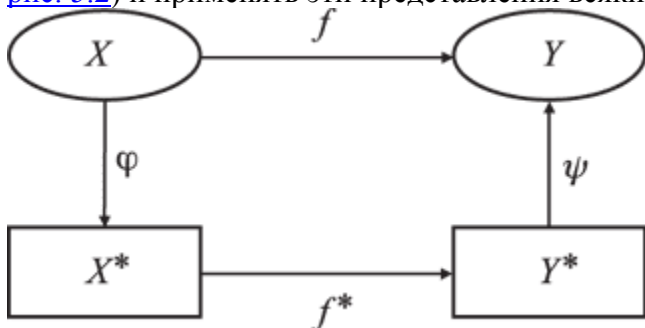


Рис. 5.2. Инвариант всех решаемых проблем науки

Правило Φ задает правило кодирования или интерпретации входного алфавита, правило Ψ - правило декодирования или интерпретации выходного алфавита, т.е. входной и выходной коды (правила, функции). При этом справедливы законы:

$$y = f(x) = \Psi(f^*(\Phi(x))).$$

Правило f^* подбирают так, чтобы, в отличие от f , его можно было бы найти и/или исследовать, применить. Для каждого сообщения x из X определена триада:

$$(x, y^*, y): x^* = \Phi(x), y^* = f^*(x^*), y = \Psi(y^*).$$

Информация - содержание сообщения, сообщение - форма проявления или актуализации информации. Информация всегда имеет носитель, актуализация информации связана с изменением носителя, ресурсов.

Пример. Сведения о сути товара могут быть изложены в рекламе, передаваемой различными сообщениями (по телевидению, по радио, в газете и т.д.). При этом соответствие этой рекламы действительности может быть независимо от типа сообщений, т.е. имеется

третья сторона информации (кроме ее абстрактной сущности, ее представления сообщениями) - соответствие сведений, заложенных в информации, с проявлениями реальной системы.

Задачи и упражнения

1. Для задачи решения квадратного уравнения указать *входную, выходную, внутрисистемную информацию*, их взаимосвязи.
2. Построить тактику изучения (исследования) эпидемии гриппа в городе только *эмпирическими (теоретическими, смешанными) методами?*
3. *Эмпирическими (теоретическими, эмпирико-теоретическими) методами* получить информацию о погоде (опишите в общих чертах подходы).

Информационные системы

Информация используется для управления, но и сама она подвержена управляющим воздействиям. Основная цель этих воздействий - поддержка информационных потоков и магистралей, способствующих достижению поставленных целей при ограниченных материально-энергетических, информационно-организационных, пространственно-временных ресурсах.

Рассмотрим сказанное на примере маркетингового информационного управления.

Пример. Маркетинговое информационное управление - планирование и прогнозирование поведения системы, корпорации, на основе рыночной информации, информационных процессов и информационных технологий на рынке, в сфере бизнеса с учетом поведения и привычек покупателя и продавца, их интерактивного контакта, оперативной реакции. Чтобы понять клиентов фирмы, ее конкурентов, дилеров и т.п., не обойтись без маркетинговых исследований. Заметим, что такие исследования нужны не только в коммерческих, но и в некоммерческих организациях. Вуз интересуется, какую репутацию он имеет в глазах абитуриентов, студентов. Политическая партия, организация интересуется своим рейтингом, мнением потенциальных избирателей о ее кандидатах. Управляющие могут привлекать высококвалифицированных исследователей-маркетологов, ибо в своих собственных интересах получать информацию, которая позволяет принимать правильные решения. Они должны хорошо знать технологию проведения маркетингового исследования и *интерпретации* полученной информации, которая состоит из следующих процедур: определение проблемы, целей и расстановки их приоритетов; сбор и анализ первичной информации; анализ вторичной информации; рекомендации и использование результатов. Определение проблемы - формулирование предмета маркетингового исследования: провести информационный анализ и выяснить, какую информацию и для прояснения чего собирать? Анализ вторичной информации - анализ не устаревших данных, которые были собраны ранее для целей, не связанных с решением исследуемой проблемы, особенно данных, собранных из независимых источников, которая, как правило, весьма достоверна. Возможные источники вторичной информации: планы и финансовые отчеты; данные о сбыте; данные о прибылях и убытках; счета клиентов; данные о запасах; результаты предыдущих исследований; письменные сообщения (текущая информация); жалобы потребителей, стандарты для определения результативности; периодические издания; книги, монографии и другие неперiodические публикации; коммерческие исследовательские организации и др. Сбор и анализ первичных данных - сбор и анализ актуальной, "свежей" информации для решения конкретной исследуемой проблемы. Здесь важно отвечать на следующие вопросы: кого или что следует исследовать? какая информация должна собираться? кому поручить сбор данных? какие методы сбора данных использовать? сколько будет стоить исследование? какова методика сбора данных? как долго собирать данные? когда и где собирать информацию? в какой форме собирать, как и где хранить информацию? Рекомендации и использование результатов позволяют вырабатывать и принимать решения. Правильно, полно и корректно собранная информация позволяет маркетологам: получать преимущества; снижать финансовый риск; определять отношения потребителей; следить за внешней средой; координировать стратегию и тактику поведения; оценивать деятельность других и

собственную; повысить доверие к рекламе; получать поддержку в решениях; подкреплять интуицию; улучшать эффективность и др. Маркетинговые исследования - часть интегрированного информационного процесса, и на них воздействуют факторы окружающей среды (конкуренция, правительство, экономика и др.).

Во многих областях и в системном анализе важное значение имеет понятие "*информационная система*". Такая система отождествляется часто с некоторой системой поддержки (автоматизации) интеллектуальных работ, в частности, поиска информации, администрирования, экспертизы, принятия решений, управления, распознавания, накопления знаний, обучения и др.

Информационная система - система, в которой ее элементы, цель, ресурсы, структура (организация) рассматриваются, в основном, на информационном уровне (хотя, естественно, имеются и другие уровни рассмотрения, например, энергетический уровень).

Любая *информационная система* имеет следующие типы основных подсистем:

1. подсистема информационного обеспечения (данных);
2. подсистема интеллектуального обеспечения (информации, знаний);
3. подсистема технического обеспечения (аппаратуры);
4. подсистема технологического обеспечения (технологии);
5. подсистема коммуникативного обеспечения (интерфейса);
6. подсистема анализа и проектирования;
7. подсистема оценки адекватности и качества, верификации;
8. подсистема организационного взаимодействия и управления персоналом;
9. подсистема логистики (планирования и движения товаров и услуг).

Информационная среда - это среда (т.е. система и ее окружение) из взаимодействующих *информационных систем*, включая и информацию, актуализируемую в этих системах.

Пример. Можно выделить три основных подхода к использованию информационного менеджмента в социально-экономических системах.

1. "Отношений с общественностью" (PR - Public Relations, Пиар), при котором разрабатываются и используются системы управления социально-экономической информацией с целью создания более адекватной и благоприятной среды (включая и все виды ресурсов), общественного сознания для реализации интересов государства, монополии и человека, согласования их интересов, подчас противоречивых. Широко используются при этом методы опроса населения, изучения общественного мнения, рекламирования, прогнозирования и моделирования (особенно для повышения устойчивости и регуляции систем).

2. "Объединения достижений НТР и человека", при котором разрабатываются и реализуются системы массового обучения достижениям НТР, новым информационным технологиям, делопроизводству и т.д. с целью адаптации человека к системам с возросшими техническими и технологическими возможностями, требованиями к качеству продукции и соотношению "качество - цена".

3. "Организационного гуманизма", при котором разрабатываются и реализуются системы помещения трудящихся в стимулирующие их работу культурно-образовательные, социально-психологические, гуманистические и материальные среды с целью раскрытия их потенциальных возможностей и способностей.

Пример. Технологический и социально-экономический отрыв стран Запада и СНГ определяется, в первую очередь, не отсутствием у нас квалифицированных рабочих, хорошей образовательной системы, ресурсов, а отсутствием адекватной им системы информационного и экономического управления, его регулирования, а не саморегулирования (самоорганизации), недостаточным вниманием к третьему подходу к информационному менеджменту. Реформы должны происходить не для обогащения небольшой группы населения, а для повышения благосостояния трудящейся массы (особенно среднего класса - базы любого государства),

обеспечения его творческого и плодотворного труда. Это наиболее важно и в то же время трудно обеспечиваемо в условиях экономических реформ.

Информационная система управления - система, предназначенная для управления, - как другой системой, так и внутри системы (т.е. в качестве управляющей подсистемы).

Различают также основные 6 типов *информационных систем управления* (тип системы определяется целью, ресурсами, характером использования и предметной областью):

1. Диалоговая система обработки запросов (Transaction Processing System) - для реализации текущих, краткосрочных, тактического характера, часто рутинных и жестко структурируемых и формализуемых процедур, например, обработки накладных, ведомостей, бухгалтерских счетов, складских документов и т.д.

2. Система информационного обеспечения (Information Provision System) - для подготовки информационных сообщений краткосрочного (обычно) использования тактического или стратегического характера, например, с использованием данных из базы данных и структурированных, формализованных процедур.

3. Система поддержки принятия решений (Decision Support System) - для анализа (моделирования) реальной формализуемой ситуации, в которой менеджер должен принять некоторое решение, возможно, просчитав различные варианты потенциального поведения системы (варьируя ее параметры); такие системы используются как в краткосрочном, так и в долгосрочном управлении тактического или стратегического характера в автоматизированном режиме.

4. Интегрированная, программируемая система принятия решения (Programmed Decision System) предназначена для автоматического, в соответствии с программно реализованными в системе, структурированными и формализованными критериями оценки, отбора (выбора) решений; используются как в краткосрочном, так и в долгосрочном управлении тактического (стратегического) характера.

5. Экспертные системы (Expert System) - информационные консультирующие и (или) принимающие решения системы, которые основаны на структурированных, часто плохо формализуемых процедурах, использующих опыт и интуицию, т.е. поддерживающие и моделирующие работу экспертов, интеллектуальные особенности; системы используются как в долгосрочном, так и в краткосрочном оперативном прогнозировании, управлении.

6. Интеллектуальные системы, или системы, основанные на знаниях (Knowledge Based System) - поддерживают задачи принятия решения в сложных системах, где необходимо использование знаний в достаточно широком диапазоне, особенно в плохо формализуемых и плохо структурируемых системах, нечетких системах и при нечетких критериях принятия решения; эти системы наиболее эффективны и применимы для сведения проблем долгосрочного, стратегического управления к проблемам тактического и краткосрочного характера, повышения управляемости, особенно в условиях многокритериальности. В отличие от экспертных систем, в системах, основанных на знаниях, следует по возможности избегать экспертных и эвристических процедур и прибегать к процедурам минимизации риска. Здесь более существенно влияние профессионализма персонала, ибо при разработке таких систем необходимо сотрудничество и взаимопонимание не только разработчиков, но и пользователей, менеджеров, а сам процесс разработки, как правило, происходит итерационно, итерационными улучшениями, постепенным переходом от процедурных знаний (как делать) - к непроцедурным (что делать).

Фундаментальная ошибка с неустраняемыми последствиями в *информационных системах* - принятие неправильных стратегических решений и критериев оценки решений.

Пример. На заре компьютеризации школ многие регионы приняли решение: как можно быстрее оснастить школы компьютерами; в результате многие из них затем не могли освободиться от несовершенной, неприспособленной для задач обучения техники (КУВТ, УКНЦ, ДВК и т.п.) и получить современную технику, так как фактически техника у них была. Стратегически правильным подходом в этой ситуации был бы подход по принципу: "подождём, пока схлынет мутная вода, затем пойдет чистая".

При построении (выборе, адаптации) *информационной системы* можно использовать две основные концепции, два основных подхода (третья концепция - их комбинация):

- ориентация на проблемы, которые необходимо решать с помощью этой *информационной системы*, т.е. проблемно-ориентированный подход (или индуктивный подход);
- ориентация на технологию, которая доступна (актуализируема) в данной системе, среде, т.е. технологически-ориентированный подход (или дедуктивный подход).

Выбор концепции зависит от стратегических (тактических) и(или) долгосрочных (краткосрочных) критериев, проблем, ресурсов.

Если вначале изучаются возможности имеющейся технологии, а после определяются актуальные проблемы, которые можно решить с их помощью, то необходимо опираться на технологически-ориентированный подход.

Если же вначале определяются актуальные проблемы, а затем внедряется технология, достаточная для решения этих проблем, то необходимо опираться на проблемно-ориентированный подход.

Ошибки в выборе подхода (проблем, технологии) могут привести не только к ошибочным стратегиям и (или) тактике, но и к полному краху системы.

При этом обе концепции построения *информационной системы* зависят друг от друга: внедрение новых технологий изменяет решаемые проблемы, а изменение решаемых проблем - приводит к необходимости внедрения новых технологий; и то, и другое влияет на принимаемые решения.

Дороговизна, важность, актуальность информации определяют цели и важность (приоритеты) в управлении *информационными системами* (в *информационных системах*).

Системное проектирование (разработка) и использование *информационной системы* должно пройти следующий **жизненный цикл информационной системы**:

1. предпроектный анализ (опыт создания других аналогичных систем, прототипов, отличия и особенности разрабатываемой системы и др.), анализ внешних проявлений системы;
2. внутрисистемный анализ, внутренний анализ (анализ подсистем системы);
3. системное (морфологическое) описание (представление) системы (описание системной цели, системных отношений и связей с окружающей средой, другими системами и системных ресурсов - материальных, энергетических, информационных, организационных, людских, пространственных и временных);
4. определение критериев адекватности, эффективности и устойчивости (надежности);
5. функциональное описание подсистем системы (описание моделей, алгоритмов функционирования подсистем);
6. макетирование (макетное описание) системы, оценка взаимодействия подсистем системы (разработка макета - реализации подсистем с упрощенными функциональными описаниями, процедурами, и апробация взаимодействия этих макетов с целью удовлетворения системной цели), при этом возможно использование "макетов" критериев адекватности, устойчивости, эффективности;
7. "сборка" и тестирование системы - реализация полноценных функциональных подсистем и критериев, оценка модели по сформулированным критериям;
8. функционирование системы;
9. определение целей дальнейшего развития системы и ее приложений;
10. сопровождение системы - уточнение, модификация, расширение возможностей системы в режиме ее функционирования (с целью ее эволюционирования).

Эти этапы - основные для информационного реинжиниринга систем.

Пример. Для решения текущих задач ведения бизнеса с учетом разнородных корпоративных интересов используют специальные системы планирования ресурсов предприятия (ERP), как правило, на основе удаленного ("клиент-сервер") доступа к единой базе данных и единых, предопределенных бизнес-процессах, транзакциях, с открытой

многоплатформенной архитектурой. Системы ERP позволяют различным организациям (география не имеет значения), с различными *информационными системами* создавать общую интегрированную *информационную систему*, используя для этого опыт и решения, разработанные фирмой-поставщиком ERP.

Пример. Особенно актуален реинжиниринг бизнес-процессов (BRP), т.е. процесс реинтеграции и рационализации производственных процессов. Реинжиниринг - это не только уменьшение размерности и сложности системы, не только реструктуризация, не только сокращение числа уровней управления, не только стремление к автоматизации, а фундаментальный, системный пересмотр всей системы (т.е. ее проблем и ресурсов), с целью существенного (например, порядкового) улучшения показателей ее эффективности, устойчивости, рентабельности, жизнеспособности (не только стратегического, но и тактического характера). Это веление времени, современного производства (товаров, услуг, знаний), так как проблемы часто (до реинжиниринга) необходимо разделить на сотни более мелких, коллективы - на сотни более мелких рабочих групп, знания необходимо привлекать как специализированные, так и более широкого диапазона, распределять ресурсы - например, географически. Все это нужно согласовывать по времени, по пространству, по структуре (организации), и число менеджеров при этом резко возрастает.

Пример. Глобализация (рынков, производства, конкуренции и др.) предъявляет свои требования по сближению национальных систем менеджмента. Появилась новая парадигма (т.е. система определяющих концепций, взглядов) глобального менеджмента (GMP) или глобального телекоммуникационного менеджмента. Ее можно назвать системой стратегического глобального реинжиниринга, основанной на знаниях, горизонтальных связях, корпоративной динамике (способность к гибкой онлайн-реакции компании) ориентированной на клиента (продукт, услуги).

Индустрия *информационных систем* опирается на следующие процессы:

1. повышение мультимедийности, гипермедийности;
2. повышение дружелюбности к пользователю;
3. интеграция;
4. повышение открытости;
5. распределенность;
6. объектно-ориентированный подход;
7. метабазирование данных и *информационных систем*;
8. мультиагентное рассмотрение и др.

В последнее время рассматриваются (проектируются, разрабатываются и используются) так называемые корпоративные *информационные системы*, т.е. *информационные системы* в масштабе корпорации, организации.

Примеры. *Информационная система* пенсионных выплат населению, *информационная система* здравоохранения региона, *информационная система* биржевой деятельности.

Для разработки корпоративной *информационной системы* необходимо осуществить следующие мероприятия:

1. информационное обследование корпорации с целью выяснения ее основных целей функционирования, элементов, структуры, направлений, приоритетов и задач деятельности, информационных потоков и технологий, эволюционных возможностей корпорации, критериев оценки эффективности системы (результат этого этапа - проект *информационной системы*);

2. выбрать одну из двух основных концепций разработки *информационной системы* - проблемно-ориентированную или технологически-ориентированную (результат этого этапа - архитектура *информационной системы*, например, архитектура "клиент-сервер" с удаленным сервером баз данных, а также инструментальное и другое обеспечение системы, например, ОС UNIX, ORACLE);

3. определить ключевые элементы, подсистемы, в частности, подсистему управления корпоративной базой данных, подсистему автоматизации делопроизводства,

подсистему согласования, принятия и контроля решений, подсистему тренинга (результат этапа - структура системы и подсистем, например, с использованием Lotus Notes, Action Workflow, EDMS - Electronic Document Management Systems, CBR Express).

При разработке целей, определении ресурсов необходимо тесное взаимодействие управляющего, проектирующего, разрабатывающего и пользовательского звена системы. Здесь недопустимы ложные критерии конфиденциальности и защиты информации, всегда влияющие негативно на стратегическое и долгосрочное планирование и прогнозирование, а также непрофессионализм принятия решений в каждом звене.

Главный лозунг разработки *информационных систем*: "Разработка *информационной системы* осуществляется не для внедрения (использования) *информационной системы*, а для обеспечения эффективного управления, функционирования, планирования и прогнозирования, эволюции системы, которую она информационно поддерживает".

Дадим ряд утверждений, формулируемых нами содержательно в виде **аксиом управления информационными системами**.

Аксиома 1. Количество информации в любой подсистеме иерархической системы определяется (как правило, мультипликативно) количеством сигналов, исходящих от подсистемы нулевого уровня (исходной вершины) и достигающих данной подсистемы (или входящих в данную подсистему), и энтропией этих сигналов.

Аксиома 2. Энтропия любого элемента управляющей подсистемы при переходе в новое целевое состояние (при смене цели) определяется исходным (от нулевого уровня) информационным потоком и энтропией этого элемента.

Аксиома 3. Энтропия всей управляющей подсистемы при переходе в новое целевое состояние определяется (как правило, аддитивно, интегрально) энтропией всех ее элементов.

Аксиома 4. Полный информационный поток, направленный на объект управления за период его перехода в новое целевое состояние, равен разности энтропии всей управляющей подсистемы при переходе в новое целевое состояние и энергии объекта управления, затрачиваемой объектом управления на переход в новое состояние.

Аксиома 5. Информационная работа управляющей подсистемы по преобразованию ресурсов состоит из двух частей - работы управляющей подсистемы, затраченной на компенсацию исходной энтропии, и работы, направленной на управляемый объект, т.е. на удерживание системы в устойчивом состоянии.

Аксиома 6. Полезная работа управляющей подсистемы в течение некоторого промежутка времени должна соответствовать полному информационному потоку, воздействующему на управляемую систему (в соответствии с аксиомой 4) за рассматриваемый период времени.

Соответствие, позволяющее переходить от абстракции "Информация" к ее конкретизации "Сообщение", называется *интерпретацией* информации с помощью определенной знаковой системы, некоторого алфавита, т.е. системы, с помощью которой представляется сообщение. *Интерпретация* информации всегда связана со смыслом (с семантикой) и с *пониманием* (с прагматикой). Соответствие такого типа всегда устанавливается при отождествлении данного сообщения с информацией, при актуализации информации.

Информация, которая может быть актуализируема в некоторой *информационной системе*, отражается некоторой математической (алгебраической) структурой (см. [лекцию 3](#)).

Пример. Часто эта алгебраическая структура - полугруппа, а проблема преобразования информации сводится к известной проблеме теории полугрупп, а именно - к проблеме тождества слов в свободных полугруппах.

Интерпретация информации - переход от представления элементов этой математической структуры к его семантическому смыслу.

Понимание - это соотнесение данной математической структуры с некоторыми элементами или системами реального мира (выяснение прагматического, например, экономического смысла).

Любая *интерпретация* I , которая соответствует некоторому сообщению S , может быть оценена своим информационным содержанием $I(S)$, и, таким образом, *интерпретация* есть отображение $I: R \rightarrow A$, где R - заданные сообщения, A - заданная информация.

Совокупность, кортеж вида $V = \langle A, R, I \rangle$ и есть формально *информационная система*.

Задачи и упражнения

1. Написать эссе на тему "Инжиниринг и реинжиниринг *информационных систем*".
2. Привести пример системы, указать ее управляющую (информационную) подсистему, определить тип *информационной системы управления*.
3. Построить (спроектировать) один несложный проект *информационной системы* (проходя весь жизненный цикл проектирования).

Интеллектуальные системы

Предметом информатики является обработка информации по известным законам. Предметом *ИИ* является изучение интеллектуальной деятельности человека, подчиняющейся заранее неизвестным законам. *ИИ* это все то, что не может быть обработано с помощью алгоритмических методов.

Системой будем называть множество элементов, находящихся в отношениях друг с другом и образующих причинно-следственную связь.

Адаптивная система - это система, которая сохраняет работоспособность при непредвиденных изменениях свойств управляемого объекта, целей управления или окружающей среды путем смены алгоритма функционирования, программы поведения или поиска оптимальных, в некоторых случаях просто эффективных, решений и состояний. Традиционно, по способу адаптации различают самонастраивающиеся, самообучающиеся и самоорганизующиеся системы.

Под алгоритмом будем понимать последовательность заданных действий, которые однозначно определены и выполнимы на современных ЭВМ за приемлемое время для решаемой задачи.

Под *ИС* будем понимать *адаптивную систему*, позволяющую строить программы целесообразной деятельности по решению поставленных перед ними задач на основании конкретной ситуации, складывающейся на данный момент в окружающей их среде.

Сделаем два важных дополнения к данному *определению*.

1. К сфере решаемых *ИС* задач относятся задачи, обладающие, как правило, следующими особенностями:
 - в них неизвестен алгоритм решения задач (такие задачи будем называть интеллектуальными задачами);
 - в них используется помимо традиционных данных в числовом формате информация в виде изображений, рисунков, знаков, букв, слов, звуков;
 - в них предполагается наличие выбора (не существует алгоритма - это значит, что нужно сделать выбор между многими вариантами в условиях неопределенности). Свобода действий является существенной составляющей интеллектуальных задач.
2. *Интеллектуальные робототехнические системы (ИРС)* содержат переменную, настраиваемую модель внешнего мира и реальной исполнительной системы с объектом управления. Цель и управляющие воздействия формируются в *ИРС* на основе *знаний* о внешней среде, объекте управления и на основе моделирования ситуаций в реальной системе.

О каких признаках интеллекта уместно говорить применительно к *интеллектуальным системам*? *ИС* должна уметь в наборе фактов распознать существенные, *ИС* способны из имеющихся фактов и *знаний* сделать выводы не только с использованием дедукции, но и с помощью аналогии, индукции и т. д. Кроме того, *ИС* должны быть способны к самооценке - обладать рефлексией, то есть средствами для оценки результатов собственной работы. С

помощью подсистем объяснения *ИС* может ответить на вопрос, почему получен тот или иной результат. Наконец, *ИС* должна уметь обобщать, улавливая сходство между имеющимися фактами.

Можно ли считать шахматную программу *интеллектуальной системой*? Если шахматная программа осуществляет выбор и принятие решений в условиях неопределенности на основе эффективных методов принятия решений и эвристик, корректируя свою игру от партии к партии в лучшую сторону, то такую программу можно считать достаточно интеллектуальной.

Следует определить также понятие *знания* - центрального понятия в *ИС*. Рассмотрим несколько определений.

1. **Знания** есть результат, полученный познанием окружающего мира и его объектов.
2. **Знания** - система суждений с принципиальной и единой организацией, основанная на объективной закономерности.
3. **Знания** - это формализованная информация, на которую ссылаются или которую используют в процессе логического вывода (рис. 1.1).
4. Под **знаниями** будем понимать совокупность фактов и правил. Понятие правила, представляющего фрагмент *знаний*, имеет вид:
если <условие> то <действие>

Например, если X истинно и Y истинно, то Z истинно с достоверностью P.

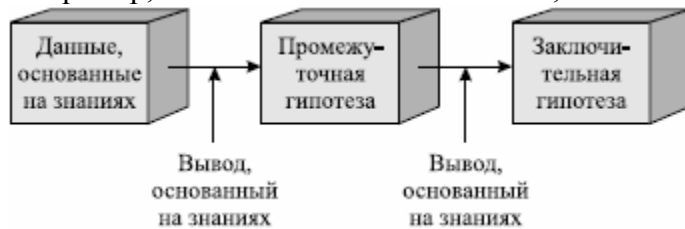


Рис. 1.1. Процесс логического вывода в *ИС*

Определения 1 и 2 являются достаточно общими философскими определениями. В *ИС* принято использовать определение 3 для определения *знаний*. Определение 4 есть частный случай определения 3.

Под статическими *знаниями* будем понимать *знания*, введенные в *ИС* на этапе проектирования. Под динамическими *знаниями* (опытом) будем понимать *знания*, полученные *ИС* в процессе функционирования или эксплуатации в реальном масштабе времени.

Знания можно разделить на факты и правила. Под фактами подразумеваются *знания* типа «А это А», они характерны для баз данных. Под правилами (продукциями) понимаются *знания* вида «ЕСЛИ-ТО». Кроме этих *знаний* существуют так называемые метазнания (*знания о знаниях*). Создание продукционных систем для представления *знаний* позволило разделить *знания* и управление в компьютерной программе, обеспечить модульность продукционных правил, т. е. отсутствие синтаксического взаимодействия между правилами. При создании моделей представления *знаний* следует учитывать такие факторы, как однородность представления и простота понимания. Выполнить это требование в равной степени для простых и сложных задач довольно сложно.

Среди важнейших классов задач, которые ставились перед *ИИ* с момента его зарождения как научного направления (с середины 50-х годов XX века), следует выделить следующие трудно формализуемые задачи, важные для задач *робототехники*: доказательство теорем, управление роботами, распознавание изображений, машинный перевод и понимание текстов на естественном языке, игровые программы, машинное творчество (синтез музыки, стихотворений, текстов).

Изучение приемов *доказательства теорем* сыграло важную роль в развитии *ИИ*. Формализация дедуктивного процесса с использованием логики предикатов помогает глубже понять некоторые компоненты рассуждений. Многие неформальные задачи, например, медицинская диагностика, допускают формализацию как задачу на *доказательство теорем*. Поиск *доказательства математической теоремы* требует не только произвести дедукцию,

исходя из гипотез, но также создать интуитивные догадки и гипотезы о том, какие промежуточные утверждения следует доказать для вывода *доказательства* основной теоремы.

Для решения трудно формализуемых задач и, в частности, для работы со *знаниями* были созданы языки программирования для задач *ИИ*: LISP (1960 год, J. MacCarthy), Пролог (1975-79 годы, D. Warren, F. Pereira), ИнтерLISP, FRL, KRL, SMALLTALK, OPS5, PLANNER, QA4, MACSYMA, REDUCE, РЕФАЛ, CLIPS. К числу наиболее популярных традиционных языков программирования для создания *ИС* следует также отнести С++ и Паскаль.

Рождение *робототехники* выдвинуло задачи машинного зрения и *распознавания изображений* в число первоочередных.

В традиционном *распознавании образов* появился хорошо разработанный математический аппарат, и для не очень сложных объектов оказалось возможным строить практически работающие системы классификации по признакам, по аналогии и т. д. В качестве признаков могут рассматриваться любые характеристики распознаваемых объектов. Признаки должны быть инвариантны к ориентации, размеру и вариациям формы объектов. Алфавит признаков придумывается разработчиком системы. Качество *распознавания* во многом зависит от того, насколько удачно придуман алфавит признаков. *Распознавание* состоит в априорном получении вектора признаков для выделенного на изображении отдельного распознаваемого объекта, и лишь затем в определении того, какому из эталонов этот вектор соответствует.

Методы *ИИ* нашли применение при создании автоматических консультирующих систем. Появляются экспертные системы (ЭС), или системы, основанные на *знаниях*.

ЭС DENDRAL (середина 60-х годов, Стэнфордский университет) расшифровывала данные масс-спектрографического анализа.

ЭС MYCIN (середина 70-х годов, Стэнфордский университет) ставила диагноз при инфекционных заболеваниях крови.

ЭС PROSPECTOR (1974-1983 годы, Стэнфордский университет) обнаруживала полезные ископаемые.

ЭС SOPHIE обучала диагностированию неисправностей в электрических цепях. ЭС XCON помогала конфигурировать оборудование для систем VAX фирмы DEC, ЭС PALLADIO помогала проектировать и тестировать СБИС-схемы.

ЭС JUDITH помогает специалистам по гражданским делам и вместе с юристом и с его слов усваивает фактические и юридические предпосылки дела, а затем предлагает рассмотреть различные варианты подходов к разрешению дела.

ЭС LRS оказывает помощь в подборе и анализе информации о судебных решениях и правовых актах в области кредитно-денежного законодательства, связанного с использованием векселей и чеков.

ЭС «Ущерб» на основе российского трудового законодательства обеспечивает юридический анализ ситуации привлечения рабочих и служащих к материальной ответственности при нанесении предприятию материального ущерба действием или бездействием.

Язык Пролог (1975-79 годы) становится одним из основных инструментов создания ЭС. Язык CLIPS (C Language Integrated Production System) начал разрабатываться в космическом центре Джонсона NASA в 1984 году. Язык CLIPS свободен от недостатков предыдущих инструментальных средств для создания ЭС, основанных на языке LISP. Появляется инструментарий EXSYS, ставший в начале 90-х годов одним из лидеров по созданию ЭС. В начале XXI века появляется теория интеллектуальных агентов и экспертных систем на их основе. Web-ориентированный инструментарий JESS (Java Expert System Shell), использующий язык представления *знаний* CLIPS, приобрел достаточную известность в настоящее время. Среди отечественных инструментальных средств следует отметить веб-ориентированную версию комплекса АТ-ТЕХНОЛОГИЯ, разработанного на кафедре Кибернетики МИФИ. В этом комплексе вся прикладная логика как комплекса в целом, так и разработанных в нем веб-интегрированных ЭС, сосредоточена на стороне сервера.

Достижения в области *машинного перевода* оказались очень полезными для *ИИ*, так как они показали, что анализ естественно-языковых конструкций невозможен без создания семантической модели, в которой интерпретируется смысл слов. Создание семантической модели привело к пониманию необходимости создания языка для внутреннего представления *знаний*. И, наконец, важным достижением явилось понимание того, что анализ текстов на естественном языке (ЕЯ) состоит из четырех основных этапов: морфологический анализ, синтаксический анализ, семантический анализ, прагматический анализ.

Задача разбора предложения на естественном языке даже с ограниченным словарем в сотню слов является типичной задачей *ИИ*. Для достаточно формальных предметных областей и конкретных наборов предметов, обладающих однозначными, недвусмысленными характеристиками и названиями, такая система была создана Т. Виноградом в 1971 году. Его знаменитая программа SHRDLU работает в абстрактном мире, состоящем из разноцветных кубиков и пирамид. В этом мире функционирует «однорукий» робот Робби, способный общаться с человеком на естественном (английском) языке и совершать манипуляции с объектами этого мира. Этот робот также способен понимать достаточно сложные выражения и самостоятельно планировать свою деятельность и объяснять мотивы тех или иных действий.

Надо отметить, что даже для английского языка, который служит основой для всех современных языков программирования в силу своей лаконичности и достаточно формальной семантики, до сего дня не удалось создать более-менее эффективную программную систему, способную адекватно понимать СМЫСЛ фраз из достаточно больших областей *знаний*, например, нашего обыденного мира.

В разборе и *понимании естественного русского языка* массу проблем создает сложная падежная система, склонения, времена, отсутствие формального порядка следования членов предложения.

Научить компьютер играть в шахматы - одна из интереснейших задач в сфере *игровых программ*, использующих методы *ИИ*. Она была поставлена уже на заре вычислительной техники, в конце 50-х годов. В шахматах существуют определенные уровни мастерства, степени качества игры, которые могут дать четкие критерии интеллектуального роста машины. В 1974 году впервые прошел чемпионат мира среди шахматных программ в рамках очередного конгресса IFIP (International Federation of Information Processing) в Стокгольме. Победителем этого состязания стала советская шахматная программа «Каисса» (Каисса - богиня, покровительница шахмат). Эта программа была создана в Москве, в Институте проблем управления Академии наук в команде разработчиков программы-чемпиона, лидерами которой были Владимир Арлазаров, Михаил Донской и Георгий Адельсон-Вельский. «Каисса» показала всему миру способности русских специалистов в области эвристического программирования.

Для различных музыкальных задач было создано специальное программное обеспечение: системы многоканального сведения; системы обработки звука; системы синтеза звука; системы интерактивной композиции; программы алгоритмической композиции и др.

Среди направлений работ в области *ИИ* следует также выделить **НЕЙРОКИБЕРНЕТИКУ**, или иначе говоря, подход к разработке машин, демонстрирующих «разумное» поведение, на основе архитектур, напоминающих устройство мозга и называемых **нейронными сетями** (НС). В 1942 году, когда Н. Винер определил концепции кибернетики, В. Мак-Каллок и В. Питс опубликовали первый фундаментальный труд по НС, где говорилось о том, что любое хорошо заданное отношение вход-выход может быть представлено в виде формальной НС. Одна из ключевых особенностей нейронных сетей состоит в том, что они способны обучаться на основе опыта, полученного в обучающей среде. В 1957 году Ф. Розенблат изобрел устройство для распознавания на основе НС - перцептрон, который успешно различал буквы алфавита, хотя и отличался высокой чувствительностью к их написанию.

НС хорошо подходят для *распознавания образов* и решения задач классификации, оптимизации и прогнозирования. Поэтому основными областями применения НС являются:

1. промышленное производство и *робототехника*;
2. военная промышленность и авиация;
3. банки и страховые компании;
4. службы безопасности;
5. биомедицинская промышленность;
6. телевидение и связь; и другие области.

Переход к ЭВМ пятого поколения означал резкий рост «интеллектуальных» способностей компьютера и возможность диалога между компьютером и непрофессиональным пользователем на естественном языке, в том числе в речевой форме или путем обмена графической информацией - с помощью чертежей, схем, графиков, рисунков. В состав ЭВМ пятого поколения также должна войти система решения задач и логического мышления, обеспечивающая способность машины к самообучению, ассоциативной обработке информации и получению логических выводов. Уровень «дружелюбия» ЭВМ по отношению к пользователю повысится настолько, что специалист из любой предметной области, не имеющий навыков работы с компьютером, сможет пользоваться ЭВМ при помощи естественных для человека средств общения - речи, рукописного текста, изображений и образов.

Здесь отметим только основные компоненты программного обеспечения (ПО), планируемые для систем пятого поколения:

- базовая программная система, включающая систему управления базой *знаний* (СУБЗ), систему приобретения и представления *знаний*, систему решения задач и получения выводов, систему обучения и объяснения решений;
- базовая прикладная система, включающая *интеллектуальную систему* автоматизированного проектирования (САПР) сверхбольших интегральных схем (СБИС) и архитектур ЭВМ, *интеллектуальную систему* программирования, систему *машинного перевода* и понимания ЕЯ, систему *распознавания образов* и обработки изображений (не менее 100 000 единиц информации в виде изображений), систему распознавания речи (не менее 10 000 слов), базы *знаний* (БЗ) о предметных областях, а также утилитные системы для ввода программ и данных, обеспечивающие диагностику и обслуживание.

Система и управление

Благодаря постоянным потокам информации (от системы к окружающей среде и наоборот) система осуществляет целесообразное взаимодействие с окружающей средой, т.е. управляет или бывает управляема. Информация стала средством не только производства, но и управления.

Своевременная и оперативная информация может позволить стабилизировать систему, приспособившись и(или) адаптироваться, восстанавливаться при нарушениях структуры и(или) подсистем. От степени информированности системы, от богатства опыта взаимодействия системы и окружающей среды зависит развитие и *устойчивость* системы.

Информация обладает также определенной избыточностью: чем больше сообщений о системе, тем полнее и точнее она управляется.

Пример. При передаче сообщений часто применяют способ двукратной (избыточной) последовательной передачи каждого символа (что позволяет избавляться от помех, "шумов" при передаче и осуществлять, например, контроль четности сигналов, по результатам которого выявляется количество сбоев). Пусть в результате сбоя при передаче приемником принято было слово вида "пррасооснтоо". Определим, какое осмысленное (имеющее семантический смысл) слово русского языка передавалось передатчиком. Легко заметить, что "претендентами на слово" являются слова "праспо", "проспо", "рроспо", "рраспо", "праспо", "рроспо", "просто" и "рраспо". Из всех этих слов осмысленным является только слово "просто".

Суть задачи *управления системой* - отделение ценной информации от "шумов" (бесполезного, иногда даже вредного для системы возмущения информации) и выделение

информации, которая позволяет этой системе существовать и развиваться. Управление - это целенаправленная актуализация знаний. Управление и особая форма - самоуправление, - высшая форма актуализации знаний.

Управление в системе - внутренняя функция системы, осуществляемая независимо от того, каким образом, какими элементами системы она должна выполняться.

Управление системой - выполнение внешних функций управления, обеспечивающих необходимые условия функционирования системы (см. [рис. 7.1](#)).



Рис. 7.1. Общая схема управления системой

Управление системой (в системе) используется для различных целей:

1. увеличения скорости передачи сообщений;
2. увеличения объема передаваемых сообщений;
3. уменьшения времени обработки сообщений;
4. увеличения степени сжатия сообщений;
5. увеличения (модификации) связей системы;
6. увеличения информации (информированности).

Как правило, эти цели интегрируются.

В целом информация используется для двух основных глобальных целей: сохранения стабильного функционирования системы и перевода системы в заданное целевое состояние.

Пример. Появление возможности управлять электрическими и магнитными колебаниями сделало массово доступным радио, телевидение, при этом скорость передачи информации достигла скорости света; пропускная способность телеканала по сравнению с пропускной способностью телефонного канала выросла примерно в 2000 раз, ускорение обработки - в миллионы раз. Возросла и сжатость информации, и информативность сообщений.

Управление любой системой (в любой системе) должно подкрепляться необходимыми ресурсами - материальными, энергетическими, информационными, людскими и организационными (административного, экономического, правового, гуманитарного, социально-психологического типа). При этом характер и степень активизации этих ресурсов может повлиять (иногда лишь косвенно) и на систему, в которой информация используется. Более того, сама информация может быть зависима от системы.

Пример. В средствах массовой информации правительство чаще ругают, актеров чаще хвалят, спортсменов упоминают обычно в связи со спортивными результатами, прогноз погоды бывает, как правило, кратким, новости политики - официальными.

Управление - непрерывный процесс, который не может быть прекращен, ибо движение, поток информации в системе не прекращается.

Цикл управления *любой системой* (в любой системе) таков:

- { сбор информации о системе →
- обработка и анализ информации →
- получение информации о траектории →
- выявление управляющих параметров →

определение ресурсов для управления →
управление траекторией системы }

Основные правила организации информации для управления системой:

1. выяснение формы и структуры исходной (входной) информации;
2. выяснение средств, форм передачи и источников информации;
3. выяснение формы и структуры выходной информации;
4. выяснение надежности информации и контроль достоверности;
5. выяснение форм использования информации для принятия решений.

Пример. При управлении полетом ракеты, наземная станция управления генерирует и в определенной форме, определенными структурами посылает входную информацию в бортовую ЭВМ ракеты; при этом сигналы отсеиваются от возможных "шумов", осуществляется контроль входной информации на достоверность и только затем бортовая ЭВМ принимает решение об уточнении траектории, ее корректировке.

Если число возможных состояний системы S равно N , то общее количество разнообразия системы равно

$$V(N) = \log_2 N.$$

Пусть управляемая система обладает разнообразием $V(N_1)$, а управляющая - $V(N_2)$. Цель управляющей системы - уменьшить значение $V(N_1)$ за счет изменения $V(N_2)$. В свою очередь, изменение $V(N_1)$, как правило, влечет изменение и $V(N_2)$, а именно, управляющая система может эффективно выполнять присущие ей функции управления лишь при условии, если верно неравенство

$$V(N_2) \geq V(N_1).$$

Это неравенство выражает **принцип Эшби** (необходимого разнообразия управляемой системы): управляющая подсистема системы должна иметь более высокий уровень организации (или большее разнообразие, больший выбор), чем управляемая подсистема, т.е. многообразие может быть управляемо (разрушено) лишь многообразием.

Пример. Менеджер фирмы должен быть более подготовлен, более грамотен, организован, свободен в своих решениях, чем, например, продавец фирмы. Малые, средние фирмы, ООО, АО - необходимый фактор разнообразия, успешного развития бизнеса, так как они более динамичны, гибки, адаптируемы к рынку. В развитых рыночных системах они имеют больший вес, например, в США доля крупных корпораций не более 10%.

Функции и задачи управления системой:

1. Организация системы - полное, качественное выделение подсистем, описание их взаимодействий и структуры системы (как линейной, так и иерархической, сетевой или матричной).
2. Прогнозирование поведения системы, т.е. исследование будущего системы.
3. Планирование (координация во времени, в пространстве, по информации) ресурсов и элементов, подсистем и структуры системы, необходимых (достаточных - в случае оптимального планирования) для достижения цели системы.
4. Учет и контроль ресурсов, приводящих к тем или иным желаемым состояниям системы.
5. Регулирование - адаптация и приспособление системы к изменениям внешней среды.
6. Реализация тех или иных спланированных состояний, решений.

Функции и задачи управления системой взаимосвязаны, а также взаимозависимы.

Пример. Нельзя, например, осуществлять полное планирование в экономической системе без прогнозирования, учета и контроля ресурсов, без анализа спроса и предложения - основных регуляторов рынка. Экономика любого государства - всегда управляемая система, хотя подсистемы управления могут быть организованы по-разному, иметь различные элементы, цели, структуру, отношения.

По характеру управления, охвата подсистем и подцелей (цели системы) управление может быть:

1. стратегическое, направленное на разработку, корректировку стратегии поведения системы;
2. тактическое, направленное на разработку, корректировку тактики поведения системы.

По времени управляющего воздействия системы могут быть: долгосрочно и краткосрочно управляемые.

Иногда отождествляют стратегическое и долгосрочное, тактическое и краткосрочное управление, но это не всегда верно.

Пример. Любая серьезная экономическая система стратегического управления должна включать в себя управляющую (информационную) подсистему, обрабатывающую, актуализирующую стратегическую информацию об инновационных мероприятиях, инвестиционных условиях, о возможностях и состояниях рынков товаров, услуг, ценных бумаг, доступных ресурсах, финансовых условиях и критериях, принципах и методах управления и др. Такие системы обычно имеют следующие цели и, часто, соответствующие им структуры:

1. управление координацией (Project Integration Management);
2. управление целями (Project Scope Management);
3. управление временем (Project Time Management);
4. управление стоимостью (Project Cost Management);
5. управление качеством (Project Quality Management);
6. управление людскими ресурсами (Project Human Resource Management);
7. управление коммуникациями (Project Communication Management);
8. управление рисками (Project Risk Management);
9. управление поставками (Project Procurement Management).

Все эти функции тесно переплетены между собой.

Выявление управляющих параметров и их использование для *управления системой* может также способствовать уменьшению сложности системы. В свою очередь, уменьшение сложности системы может сделать систему управляемой.

Система называется устойчивой структурно (динамически; вычислительно; алгоритмически; информационно; эволюционно или самоорганизационно), если она сохраняет тенденцию стремления к тому состоянию, которое наиболее соответствует целям системы, целям сохранения качества без изменения структуры или не приводящим к сильным изменениям структуры (динамики поведения; вычислительных средств; алгоритмов функционирования системы; информационных потоков; эволюции или самоорганизации - см. ниже) системы на некотором заданном множестве ресурсов (например, на временном интервале). Расплывчатое понятие "сильное изменение" каждый раз должно быть конкретизировано, детерминировано.

Пример. Рассмотрим маятник, подвешенный в некоторой точке и отклоняемый от положения равновесия на угол $0 \leq \varphi \leq \pi$. Маятник будет структурно, вычислительно, алгоритмически и информационно *устойчив* в любой точке, а при $\varphi=0$ (состояние покоя маятника) - *устойчив* и динамически, и эволюционно (самоорганизационные процессы в маятнике на микроуровне мы не учитываем). При отклонении от устойчивого состояния равновесия маятник, самоорганизуясь, стремится к равновесию. При $\varphi=\pi$ маятник переходит в динамически неустойчивое состояние. Если же рассматривать лед (как систему), то при температуре таяния эта система структурно неустойчива. Рынок при неустойчивом спросе-предложении неустойчив структурно.

Чем многообразнее входные сигналы (параметры) системы, число различных состояний системы, тем многообразнее обычно выходные сигналы, тем сложнее система, тем актуальнее проблема поиска инвариантов управления.

Понятие сложности детализируется в различных предметных областях по-разному. Для конкретизации этого понятия необходимо учитывать предысторию, внутреннюю структуру (сложность) системы и управления, приводящие систему к *устойчивому* состоянию. Впрочем, все внутренние связи на практике достаточно трудно не только описать, но и обнаружить. В

этих случаях помогает выяснение и описание связности системы, связной и *асимптотической устойчивости* ее.

Асимптотическая устойчивость системы состоит в возврате системы к равновесному состоянию при $t \rightarrow \infty$ из любого неравновесного состояния.

Пример. Известная игрушка "Ванька-встанька" - пример такой системы.

Пусть система S зависит от вектора факторов, переменных $x=(x_1, x_2, \dots, x_n)$.

Матрицей системы назовем матрицу $E=||e_{ij}||$ из 1 и 0: $e_{ij}=1$ лишь тогда, когда переменная x_i оказывает влияние на x_j .

Связная устойчивость состоит в *асимптотической устойчивости* системы при любых матрицах E .

Пример. Рассмотрим множество друзей $X=\{\text{Иванов, Петров, Сидоров}\}$ и городов $Y=\{\text{Москва, Париж, Нальчик}\}$. Тогда можно построить 3D-структуру в R_3 (в пространстве трех измерений - высота, ширина, длина), образуемую связыванием элементов X и Y , например, по принципу "кто где был" (рис. 7.2). В этой структуре были использованы сетевые 2D-структуры X, Y (которые, в свою очередь, использовали 1D-структуры). При этом элементы X и Y можно брать как точки, элементы пространства нулевого измерения R_0 .

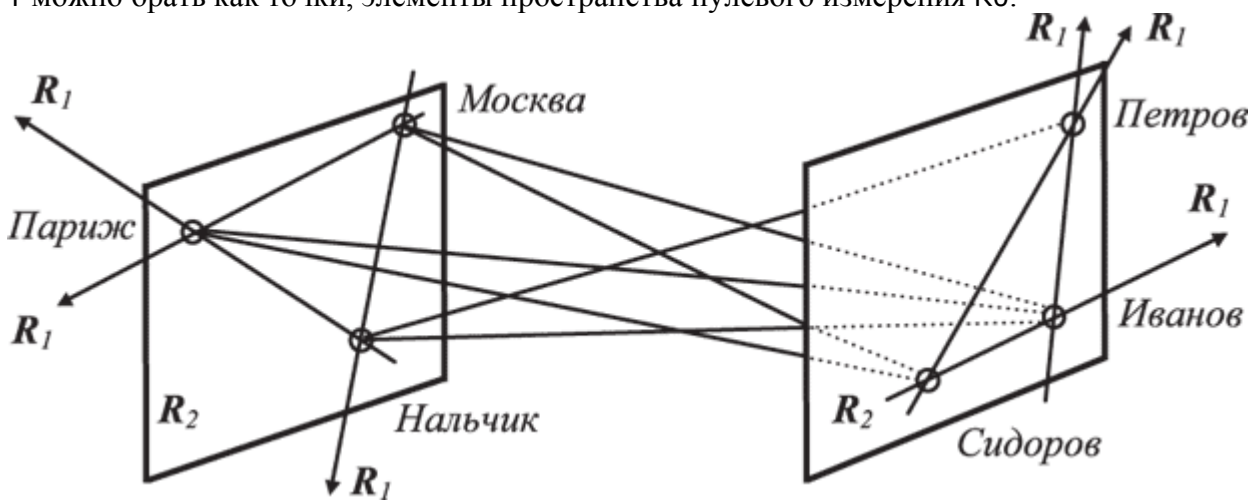


Рис. 7.2. Геометрическая иллюстрация сложных связных структур

При системном анализе различных систем, особенно социально-экономических, удобным инструментом их изображения и изучения является инструментарий когнитивной структуризации и системно-когнитивная концепция.

Когнитология - **междисциплинарное** (философия, нейропсихология, психология, лингвистика, информатика, математика, физика и др.) научное направление, изучающее методы и модели формирования знания, познания, универсальных структурных схем мышления.

Цель **когнитивной** структуризации - формирование и уточнение гипотезы о функционировании исследуемой системы, т.е. структурных схем причинно-следственных связей, их качественной и(или) количественной оценки.

Причинно-следственная связь между системами (подсистемами) A и B положительна (отрицательна), если увеличение или усиление A ведет к увеличению или усилению (уменьшению или ослаблению) B .

Когнитивная схема (карта) ситуации представляет собой ориентированный взвешенный граф, который строится по правилам:

1. вершины взаимнооднозначно соответствуют выделенным факторам ситуации, в терминах которых описываются процессы в ситуации;
2. выявляются и оцениваются (положительное влияние, отрицательное влияние) причинно-следственные связи выделенных факторов друг на друга.

Пример. *Когнитивная структурная схема* для анализа проблемы энергопотребления может иметь следующий вид ([рис. 7.3](#)):

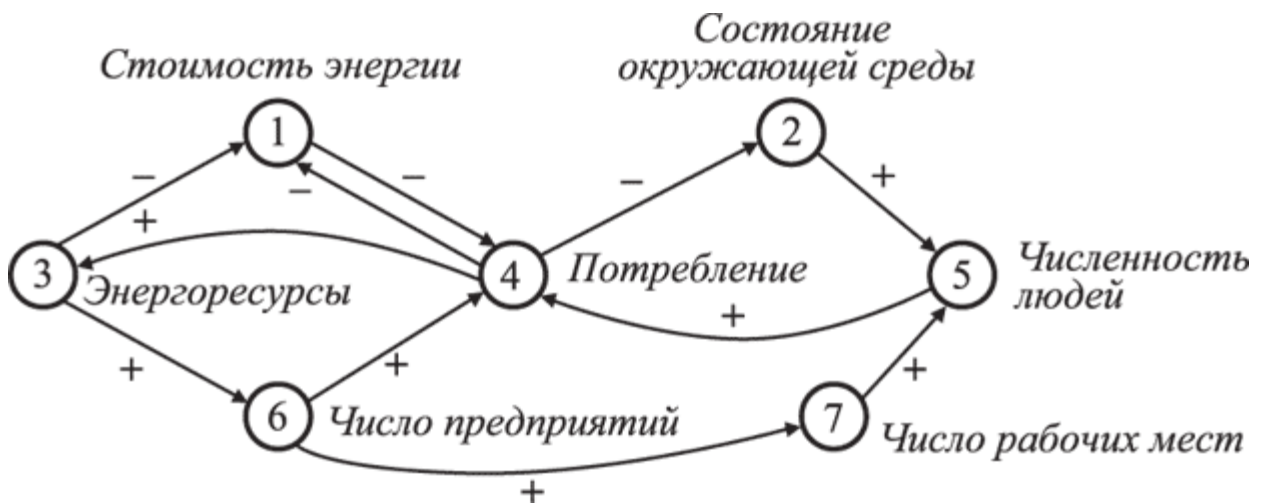


Рис. 7.3. Пример когнитивной карты

Кроме когнитивных схем (схем ситуаций) могут использоваться когнитивные решетки (шкалы, матрицы), которые позволяют определять стратегии поведения (например, производителя на рынке). Решетка образуется с помощью системы факторных координат, где каждая координата соответствует одному фактору, показателю (например, финансовому) или некоторому интервалу изменения этого фактора. Каждая область решетки соответствует тому или иному поведению. Показатели могут быть относительными (например, от 0 до 1), абсолютными (например от минимального до максимального), биполярными ("высокий или большой" - "низкий или маленький").

Пример. Такие решетки могут быть полезны, в частности, при оптимизации долевого распределения основной группы налогов между федеральным и региональным бюджетами, при выработке стратегии повышения бюджетного самообеспечения и др. На [рис. 7.4](#) показана решетка в биполярной системе показателей; зона D - наиболее, зона А - наименее благоприятная.



Рис. 7.4. Когнитивная решетка финансовой устойчивости фирмы

Когнитивный инструментарий позволяет снижать сложность исследования, формализации, структурирования, моделирования системы.

Когнитивная карта не отражает детальный характер или динамику изменения влияний в зависимости от изменения ситуации. Для этого необходимо построить соответствующую процедуру когнитивного системного анализа, по схеме, приводимой ниже.

Процедура когнитивного анализа системы, ситуации.

1. Выделение основных факторов системы.
2. Определение в выделенных факторах целевых факторов.
3. Определение факторов, которые могут влиять на целевые факторы.

4. Определение факторов, которые могут объяснять развитие системы, и их группировка в кластеры факторов (как правило, это иерархическая система, на нижнем уровне которой находятся наиболее элементарные, на следующем, - интегральные от них и т.д.).
5. Выделение в кластере группы интегральных факторов и характеризующих их показателей, которые могут быть информативными (поясняющими тенденции развития системы), и их детализация, формализация, математизация.
6. Определение связей между кластерами.
7. Определение связей и характера (например, положительный, отрицательный) и силы взаимовлияний внутри кластеров.
8. Проверка адекватности когнитивной схемы, т.е. сопоставление полученных результатов с логико-историческими проявлениями системы.
9. Корректировка, уточнение схемы.

Эта процедура лежит в основе системно-когнитивной концепции, на основе которой пытаются адекватно, структурированно, с помощью простых базовых когнитивных операций и, по возможности, формально, математически отразить и автоматизировать сущность процесса познания человеком, например, процессов вербализации, синтаксического синтеза, семантического анализа, макетирования, виртуализации и др.

1. восприятие, регистрация свойства, отношения, объекта, процесса, системы;
2. присвоение уникального имени свойству, отношению, объекту, процессу, системе;
3. шкалирование и кластеризация, классификация;
4. обобщение;
5. сравнение;
6. идентификация, узнавание объекта по его проявлениям;
7. морфологический анализ (например, связей элементов);
8. синтаксический анализ (например, атрибутов элементов и классов);
9. семантический анализ (например, связей классов);
10. верификация, сопоставление с опытом и заключение об обучении;
11. планирование эксперимента;
12. принятие решения.

Задачи и упражнения

1. Привести примеры использования (актуализации) принципа необходимого разнообразия управляемой системы и объяснить, что он регулирует.
2. Привести конкретную цель *управления системой* и управления для некоторой социально-экономической системы. Привести пример взаимосвязи *функций и задач управления системой*. Выделить параметры, с помощью которых можно *управлять системой*, изменять цели управления.
3. Построить когнитивную схему (решетку) одной проблемы на выбор.

Информация и самоорганизация систем

Рассматриваются основные понятия информационной синергетики - самоорганизация, самоорганизующаяся система, аксиомы самоорганизации информационных систем, примеры.

Цель лекции: введение в информационную синергетику и ознакомление с самоорганизующимися системами.

Любая открытая система эволюционирует, начиная с состояния наибольшей энтропии (неопределенности), спиралеобразно, актуализируя все новые связи и отношения, стремясь к организованности и порядку в системе в процессе взаимоотношений со средой, перестраивая свою структуру с целью уменьшения энтропии.

Пример. На телевизионной игре "Что? Где? Когда?" обсуждение вопроса часто начинается хаотично, спонтанно, независимо и в конце обсуждения может organizоваться в единодушное принятие правильного решения.

Самоорганизация - это образование пространственной, временной, информационной или функциональной организации, структуры (точнее, стремление к организованности, к образованию новой структуры) за счет внутренних ресурсов системы в результате целенаправленных взаимодействий с окружением системы.

Система является **самоорганизующейся**, если она без целенаправленного воздействия извне (с целью создания или изменения структуры системы) обретает пространственную, временную, информационную или функциональную структуру.

Пример. Одна макроструктура (лед) при нагревании переходит в другую макроструктуру (жидкость) с совершенно другими свойствами (например, механическими), а при дальнейшем нагревании переходит в другую макроструктуру (пар), снова с иными микроскопическими свойствами.

Самоорганизация (явная или неявная) наблюдается в сложных открытых системах. **Самоорганизации** присущ атрибут - управление. Автоматической **самоорганизации** системы не происходит, для этого необходимо управляющее воздействие. **Самоорганизация** - лишь возможный путь развития, эволюции системы. Это путь движения системы к упорядоченности, пусть и относительной. Четких мер, критериев упорядоченности нет, даже для физических, химических, биологических систем, где проблема порядка, равновесия давно изучаются.

Самоорганизация может наблюдаться как в живых, так и в неживых системах.

Пример. История развития ЭВМ - пример **самоорганизации**: от 1-го поколения ЭВМ (40-50-е годы XX века) с электронными лампами и быстродействием порядка 10⁴ операций в секунду, до 1-го поколения оптических ВМ (конец 90-х годов) с голографической памятью, с логикой на потоках фотонов, нейроподобных архитектурах и быстродействием порядка 10¹² операций в секунду.

Пример. Человеческое общество развивается спиралевидно, циклически: циклически повторяются **катастрофы**, засухи, неурожаи, эпидемии и т.п. Например, происходит переход от малого ледникового периода к общему постепенному потеплению, а число экстремальных природных явлений не только не уменьшается, но и увеличивается, в частности, в XVIII м веке было лишь 66 лет, в которых отмечались землетрясения в России.

Пример. Известный ученый А.Л. Чижевский, наблюдая в 20-х годах XX века за пятнами на Солнце и изучая их образование, обнаружил, что некоторые периоды усиления солнечной активности и периоды усиления военных действий Первой мировой войны совпадают. Интересную закономерность открыл Р. Вульф во второй половине 19-го века: $w = k(n + 10m)$, где k - коэффициент пропорциональности, определяемый разрешающей способностью телескопа, n - общее число замеченных солнечных пятен, m - число групп пятен, w - число Вульфа, по которым можно определять солнечную активность. Существует, таким образом, 11-летний цикл солнечной активности.

Любая деятельность вопреки эволюционным процессам в системе, вопреки принципам **самоорганизации**, - противосистемна.

Пример. Любые экономические решения, противоречащие основному регулятору рынка, основному механизму ее организации - соотношению "спрос-предложение" приводят к вредным последствиям для системы и для ее **самоорганизации**. Например, выпуск товаров в объеме, превышающем спрос на рынке, может привести к снижению спроса.

В рамках идеи ноосферы, гармоничных взаимоотношений человека и природы, человек выступает как органическая часть природы.

Окружение человека (включая природу и общество) - нестабильное, неустойчивое, неравновесное, развивающееся. При рассмотрении проблем такого мира надо учитывать два его противоположных и взаимосвязанных, взаимно обуславливающих друг друга качества, - стабильность и нестабильность, порядок и хаос, определенность и неопределенность.

Нестабильность и неустойчивость не всегда есть зло, отрицательное качество, подлежащее устранению. Неустойчивость может, в соответствии с законами синергетики, выступать условием стабильного и динамического саморазвития, которое происходит за счет

уничтожения, изъятия нежизнеспособных форм. Устойчивость и неустойчивость в системе, образование новых структур и разрушение старых, сменяя друг друга, развивают, эволюционируют систему. Порядок и беспорядок возникают и существуют одновременно: один включает в себя другой - это два аспекта одного целого, они дают нам различное видение мира. Из-за этого мы не можем полностью держать под контролем окружающий мир нестабильных процессов, например, полностью контролировать социально-экономические процессы.

Современная наука и технология имеют дело со сложными системами, связь между которыми осуществляется не только через порядок, через структуры порядка, но и через хаос. Только в единстве порядка и хаоса может быть исследована эволюция сложной системы. Сложная система - целое, состоящее из устойчивых и неустойчивых частей. Здесь целое - уже простая сумма частей. Эволюция такой системы ведет к новому качеству, включая и отношения с человеком. Человек находится не вне изучаемого объекта, а внутри его, познавая это целое по его составным частям, объединяя естественные науки, усиливая междисциплинарные связи, сближая естественные и гуманитарные проблемы наук, науку и искусство. Идеи, принципы, методы и технологии современного естествознания (синергетики, информатики, системного анализа, физики открытых систем и др.), все шире внедряются в гуманитарную и социально-экономическую сферы. Есть и обратные процессы.

Пример. Естествознание и гуманитарные науки все чаще исследуют процессы и системы в совокупности с человеком: медико-биологические проблемы, экологические, включая биосферу в целом (глобальная экология), биотехнологии (генная инженерия), системы "человек-машина" и т.д. Специфику современной науки все более определяют комплексные исследовательские программы (в которых принимают участие специалисты различных областей знания), междисциплинарные исследования. Например, красота - это не только категория гуманитарная, отражение гармонии материального мира, но и категория научная, красота теоретических построений. Поиски красоты, т.е. единства и симметрии законов природы, - примечательная черта современной физики, математики, биологии, синергетики и других естественных наук. Исторический метод прилагается к большему кругу систем, например, вводится даже в квантово-механическую интерпретацию, где он ранее не применялся.

Компьютеризация и информационные технологии позволяют передавать машине все более усложняющиеся логические операции. Человеческий мозг освобождается от формализуемой, стандартизированной, рутинной логической деятельности.

Пример. Эвристические процедуры, интуиция, опыт человека, эксперта находят применение в программировании, например, при разработке антивирусных программ.

Эволюцию системы можно, как сказано выше, понимать как целенаправленное (на основе выбора) движение, изменение этой системы (как неравновесной) по некоторой траектории развития, состоящей из точек состояний.

Пример. Чтобы жить достойно, люди и правительство должны ориентироваться на социально-экономическую перспективу, т.е. траектория развития любой страны должна быть эволюционной.

Устойчивость системы - ее способность сохранять свое движение по траектории на таком уровне потребления ресурсов, который может самоподдерживаться, саморегулироваться достаточно долго.

Традиционная макроэкономика ориентируется на непрерывный и, чаще всего, количественный рост, а не на устойчивость. Для развития, эволюции требуется все больше материальных, энергетических, информационных ресурсов, а их рост сужает пространство устойчивого развития общества, снижает жизнеспособность.

Пример. При достаточно высоком уровне образования и развитой системе образования, научно-техническая и технологическая области последние два десятилетия слабо развивались в России. Например, в США в 1996 г. на науку расходы государства составляют 2,8-2,9% ВВП страны, в Японии - 3,3%, в России - 0,59%. По показателю достаточности и уровню

квалификации трудовых ресурсов Россия занимает 46-е место. По оценкам специалистов, если Россия в ближайшие годы не поднимется хотя бы на 20-е, то ее экономический крах гарантирован.

Эффективность системы - способность системы оптимизировать (глобально-потенциально или локально-реально) некоторый критерий *эффективности* типа соотношений "затраты на обеспечение ресурсом - объем поступлений новых ресурсов".

Пример. Для социально-экономической системы - это способность производить социально-экономический эффект и не ухудшать движение по пути к достижению поставленной цели. Например, критерием *эффективности* банковской системы может быть не только прибыль, но и кредитование, возврат кредитов.

Эффективными можно считать действия в системе, которые поддерживают *самоорганизацию* системы при низком уровне энтропии за счет неравновесных процессов взаимного обмена энергией, веществом и информацией с окружающей средой.

Актуальна разработка механизмов, которые обеспечивали бы устойчивое развитие общества (в частности, социально-экономических систем) и каждого его члена в отдельности без количественного увеличения ресурсов, с помощью произведенного труда, стоимости и капитала.

Пример. Показателями развития общества могут служить ВНД - валовой национальный доход и ВВП - валовой национальный продукт, но и они не позволяют полно оценивать устойчивость развития общества, его систем, не позволяют оценивать, живет ли общество по средствам, заботясь о будущих поколениях, т.е. адекватны ли "кредитные социо-экономико-экологические отношения природы и общества", развитие культуры, науки и др.

Эволюция системы определяется борьбой организации и дезорганизации в системе, накоплением и усложнением информации, ее организацией и самоорганизацией, сложностью и разнообразием внутрисистемных процессов. Важным критерием *эффективности* системы (политики) является ее динамическая, структурная и организационная предсказуемость, отсутствие аномалий и обеспечение динамического роста, наличие и динамическая актуализация критериев оценки принимаемых решений.

Современному обществу и природе, с их множеством возможных путей развития, нельзя навязывать эти пути, они избираются на принципах самоуправления и саморегулирования, а именно, за счет целенаправленных воздействий на процессы с целью возврата траектории эволюции на желаемую траекторию (если в результате, например, стохастических воздействий система отклонилась от траектории).

При этом, в соответствии с *принципами синергетики*, необходимо учесть, что в неустойчивой социально-экономической среде действия каждого отдельного человека (микропроцессы) могут повлиять на всю систему в целом (макропроцессы).

Пример. В условиях неустойчивой экономической политики действия отдельных структур могут повлиять на социально-экономические процессы общества, что наблюдалось, например, в Ираке и других странах.

Стратегическое планирование в социально-экономических системах - ресурсообеспеченные и целенаправленные действия руководства, ведущие к разработке наилучших в каком-то смысле (например, локально-оптимальных) стратегий динамического поведения всей системы, которые приводят в окрестность поставленных целей.

Стратегическое планирование - инструмент, помогающий принимать управленческие решения по осуществлению основных задач:

1. распределения ресурсов;
2. адаптации к изменениям внешних факторов;
3. внутренней координации и мобилизации;
4. осознания организационных стратегий и целей (краткосрочных, среднесрочных, долгосрочных), динамической переоценки достижимости целей.

Пример. Планирование в социально-гуманитарной системе необходимо для достижения следующих целей:

1. повышение контрольных функций;
2. предвидение требований социальной и гуманитарной политики;
3. обеспечение своевременной реакции на изменения в системе;
4. улучшение социально-гуманитарного и экономического состояния;
5. уменьшение неопределенности, риска, увеличение *эффективности* и др.

Козэволюция - сопряженное, взаимообусловленное изменение систем или частей внутри целого. Это принцип глобальной эволюции. Само это понятие пришло из эволюционной популяционной теории.

Понятие *коэволюции* тесно связано с понятием "*самоорганизации*". *Самоорганизация* имеет дело со структурами, состояниями развивающихся систем, а *коэволюция* - с отношениями между такими системами, с взаимосвязями эволюционных изменений.

Пример. В последние годы активно формируется новое направление исследований - эволюционная экономика. Волнообразный, циклический характер действия основных законов наблюдается в устойчивой системе. Особенно заметен волнообразный характер социально-экономических процессов переходного периода. Государственное регулирование, его масштабы и значение претерпевают существенные изменения по мере эволюции экономики. Уменьшение роли государства будет чередоваться с периодами ее волнообразного возрастания. Снижение роли и значения системы государственного регулирования при эволюции социально-экономической системы будет чередоваться с ее усилением на определенных этапах переходного периода, будут наблюдаться периоды либерализации и контроля над ценами и заработной платой, волнообразный характер процесса приватизации и т.п. Волнообразный характер социально-экономических процессов можно объяснить следующим образом. Несмотря на различие рыночных и не рыночных государственных методов хозяйствования, их действие во многом взаимодополняющее. Более того, в развитой экономике рыночные (часто - стихийные) и государственные (часто - плановые) методы сочетаются и диффундируют, обеспечивая возврат на устойчивую траекторию развития при отклонениях от нее. В этом и заключается первопричина волнообразности.

Волнообразный и циклический характер имеют многие процессы социально-экономической и гуманитарной сфер, например, в сфере политики, права, информации и печати, религии, национальных отношений, миграционных процессов, распространения технологий, активности военных действий и др. Многие из этих циклических процессов связаны с циклами солнечной активности.

Катастрофами называются скачкообразные дестабилизирующие изменения, возникающие в виде отклика системы на плавное изменение условий окружающей среды. Эти изменения - внезапны, непрогнозируемы с уверенной точностью, резки по отношению к темпу изменения условий среды. Если представить себе траекторию эволюции системы как множество точек, каждая из которых есть точка в пространстве факторов окружающей среды, то у траектории системы могут быть точки **бифуркации** - раздвоения, качественного изменения траектории.

Пример. Так называемый "черный вторник" на валютном рынке возник на фоне плавных, ничего катастрофического не предвещающих, условий среды (внешне эти условия плавно менялись накануне).

Управляемая социально-экономическая система при определённой цели, определенных начальных данных и определённых ресурсах имеет определенную область достижимости, в которой она может достичь цели при этих ресурсах за любое время.

Сформулируем основные **аксиомы** теории информационных динамических процессов (**информационной синергетики**).

Аксиома 1. Развитие (эволюция) системы определяется некоторой целью и информационными ресурсами системы, ее информационной открытостью.

Аксиома 2. При стремлении к цели система воспринимает входную информацию, которая используется и для изменения внутренней структуры самой системы, внутрисистемной информации.

Аксиома 3. Изменение внутрисистемной информации происходит таким образом, чтобы увеличивалась негэнтропия (мера порядка) системы, уменьшалась энтропия (мера беспорядка) в системе.

Аксиома 4. Любое изменение внутренней структуры системы или внутрисистемной информации оказывает воздействие на выходную информацию системы (т.е. на окружающую среду системы); внутренняя энтропия изменяет внешнюю энтропию системы.

Большое значение при исследовании управляемости системы, ее управляющих параметров, развития системы во времени, в пространстве, по структуре имеют **синергетические принципы**, сформулированные И.Пригожиным и его последователями, в частности, следующие:

1. принцип эволюции системы, необратимости процессов ее развития;
2. принцип возможного решающего воздействия (при определенном стечении обстоятельств) малых изменений поведения системы на ее эволюцию;
3. принцип множественности (или многовариантности) путей развития системы и возможности выбора оптимальных из них;
4. принцип невмешательства в процессы самоуправяемого развития и непредсказуемости эволюционного поведения системы и, в то же время, - учёт возможности организовать управляющие воздействия на ресурсы и процессы в системе;
5. принцип учета стохастичности и неопределенности процессов (поведения систем);
6. принцип взаимодействия усложнения организации, устойчивости и темпов развития систем;
7. принцип учета факторов стабильности и нестабильности системы (возникновения устойчивости из неустойчивого поведения), порядка и хаоса в системе (возникновения порядка из хаоса), определенности и неопределенности;
8. принцип взаимовлияния устойчивости среды отдельной подсистемы или элемента (микросреды) и процессов во всей системе (макросреды).

Так как синергетика - теория возникновения новых качественных свойств и структур, а возникновение смысла (интерпретация и понимание сообщений) всегда связано с качественными изменениями в системе, то можно говорить об информационной *самоорганизации*. Информация - синергетическая среда, с помощью которой поддерживается вся система, ее отдельные подсистемы и которая генерирует информацию о том, как должна развиваться (саморазвиваться) система.

Важным условием рождения информации в системах является их открытость. В замкнутых системах, согласно второму началу термодинамики (энтропия замкнутой системы не может убывать и растет до тех пор, пока не достигнет максимума, а, следовательно, информация становится минимальной), структуры распадаются (на макроскопическом уровне). Поэтому информация не может рождаться и храниться в системах в состоянии теплового равновесия, так как в замкнутых системах всегда устанавливается тепловое равновесие.

Пример. Магнитный диск в состоянии теплового равновесия размагничивается и не может хранить информацию. Со временем дискеты приходят в состояние теплового равновесия, и магнитное покрытие разрушается, информация теряется.

Открытые системы поддерживают "дистанцию" от состояния теплового равновесия - за счет потоков ресурсов (вещества, энергии, информации) и за счет *самоорганизации*, вследствие которой эти потоки существуют и направляются в соответствии с подчиненностью постоянно (от элементов - к подсистемам, от них - к системе).

Пример. Структурной единицей нервной системы является нейрон - нервная клетка. Кора переднего отдела мозга содержит несколько десятков миллиардов нейронов. Нейроны бывают различного типа: сенсорные (от рецептора кожи к спинному мозгу); сетчатки (от рецепторов сетчатки глаза - к зрительному нерву); двигательные (от рецепторов мышц - к двигательной коре). Они образуют своеобразные регистры (зрительные, слуховые, тактильные

и др.). Нейрон служит для передачи информации за счет нервных импульсов. Расшифровка нервных импульсов (информации) происходит в соответствующих областях коры головного мозга. Нейроны коры мозга функционируют параллельно. Это - их замечательное преимущество (по сравнению с памятью другого вида). В коре переднего мозга около 50 млрд. нейронов. Они организованы, примерно, в 600 млн. функционирующих параллельно систем. Производительность такого типа "процессора" (распределенного матричного или нейросистемы) очень впечатляет (оцените примерно ее!). Особенностью мозга является высокое качество, скорость обработки информации. Нейроны выполняют обработку со скоростью всего около 100 инструкций в секунду (сравнить с ЭВМ, выполняющей миллионы инструкций в секунду), но они быстрее и эффективнее решают наиболее сложные (для ЭВМ, в частности) задачи распознавания и классификации, принятия решений и другие плохо формализуемые и структурируемые проблемы. Человеческий мозг - это система параллельно работающих подсистем, структур, самоорганизующихся с помощью ассоциативных связей для выработки, принятия логических (алгоритмических, рациональных) решений. Там, где невозможно принять такое решение (т.е. не удастся ассоциировать такие связи), принимается эвристическое решение. На каждом нейроне коры головного мозга одновременно (параллельно) обрабатываются возбуждения разного типа: мотивации, целеполагания, внешние возбуждения - отражения текущего состояния управляемого объекта, возбуждения памяти (опыта). Их согласованная обработка дает картину объекта и позволяет принимать решения. Так, мозг, непрерывно перебирая результаты всех прошлых действий в аналогичных ситуациях и сравнивая их с текущей ситуацией, выбирает вариант, наиболее подходящий, целесообразный и эффективный в данной конкретной ситуации. Если при этом не найдется такая ситуация, то выбирается (прогнозируется многокритериально) такое состояние, результат которого будет наиболее адекватен; этот результат и запоминается далее. У человека существует самостоятельная потребность в информации. Нормальная жизнедеятельность возможна лишь тогда, когда из внешней среды имеется приток не только вещества, энергии, но и информации, когда нет явлений "информационного голода". Получение новой информации связано со сжатием информации, например, с пересылкой в долговременную память (подсознание) образов, смыслов и т.д.

На этапе *самоорганизации* вырабатывается коллективное, корпоративное поведение (т.е. новый уровень иерархии образования смысла, семантики). В живых системах при этом используется не только связь со средой, но и генетически заложенная информация или информация *самоорганизации*.

Пример. Стадо буйволов (каждый из которых в отдельности достаточно беззащитен перед стаей хищников) во время нападения самоорганизуется: молодежь - в центре, самцы - по окружности ("рогами наружу"). Это важно для выживания всего стада.

Информация может быть неполной, образной, например, в виде фрагментов, по которым быстро восстанавливается (самоорганизуется) более полная информация. Особенно важно быстро и полно восстанавливать эту информацию. Поэтому необходим процесс обучения, сжатия и передачи информации, знаний от поколения к поколению. Так как области знаний расширяются и углубляются, а информация лавинообразно растет, то важно находить синергетические инварианты, принципы, технологии ее передачи.

Наблюдаемая математизация и информатизация современной науки убедительно показывает, что их *эффективность* зависит как от данной науки, сложности и возможности адекватного описания ее законов и принципов математическими и информационными моделями, так и от используемого математического аппарата.

Задачи и упражнения

1. Написать эссе на тему "*Самоорганизация в живой природе*".
2. Написать эссе на тему "*Самоорганизация в неживой природе*".
3. Привести пример *самоорганизующейся системы* и на её основе пояснить *синергетические принципы* И. Пригожина (предварительно ознакомившись с ними).

Распознавание изображений

Современные роботы, снабженные телевизионными камерами, способны достаточно хорошо видеть, чтобы работать с реальным миром. Они могут делать заключения о том, какого типа объекты присутствуют, в каких они находятся отношениях между собой, какие группы образуют, какой текст содержат и т. д. Однако сложные задачи распознавания, например, распознавание похожих трехмерных быстро движущихся объектов или неразборчивого рукописного текста требуют совершенствования методов и средств для своего решения.

Общая характеристика задач распознавания образов и их типы.

Под образом понимается структурированное описание изучаемого объекта или явления, представленное вектором *признаков*, каждый элемент которого представляет числовое значение одного из *признаков*, характеризующих соответствующий объект. Общая структура системы распознавания и этапы в процессе ее разработки показаны на [рис. 4.1](#).

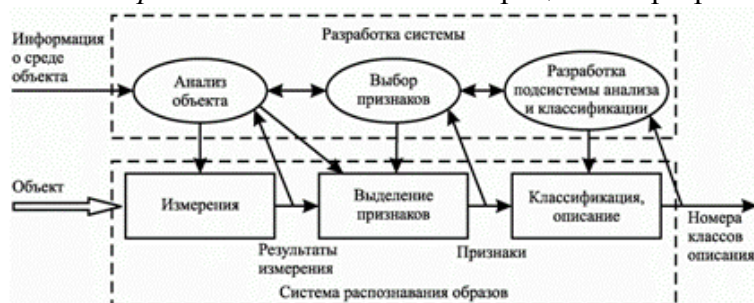


Рис. 4.1. Структура системы распознавания

Суть задачи распознавания - установить, обладают ли изучаемые объекты фиксированным конечным набором *признаков*, позволяющим отнести их к определенному классу.

Задачи распознавания имеют следующие *характерные черты*.

1. Это *информационные задачи*, состоящие из двух этапов: а) приведение исходных данных к виду, удобному для *распознавания*; б) собственно *распознавание* (указание принадлежности объекта определенному классу).
2. В этих задачах можно *вводить понятие аналогии или подобия объектов и формулировать понятие близости объектов* в качестве основания для зачисления объектов в один и тот же класс или разные классы.
3. В этих задачах можно *оперировать набором прецедентов-примеров*, классификация которых известна и которые в виде формализованных описаний могут быть предъявлены алгоритму *распознавания* для настройки на задачу в процессе обучения.
4. Для этих задач *трудно строить формальные теории и применять классические математические методы* (часто недоступна информация для точной математической модели или выигрыш от использования модели и математических методов не соизмерим с затратами).
5. В этих задачах *возможна "плохая" информация* (информация с пропусками, разнородная, косвенная, нечеткая, неоднозначная, вероятностная).

Целесообразно выделить следующие типы задач *распознавания*.

1. Задача *распознавания* - отнесение предъявленного объекта по его описанию к одному из заданных классов (обучение с учителем).
2. Задача автоматической классификации - разбиение множества объектов (ситуаций) по их описаниям на систему непересекающихся классов (таксономия, кластерный анализ, обучение без учителя).
3. Задача выбора информативного набора *признаков* при *распознавании*.
4. Задача приведения исходных данных к виду, удобному для *распознавания*.
5. *Динамическое распознавание* и динамическая классификация - задачи 1 и 2 для динамических объектов.
6. Задача прогнозирования - это задачи 5, в которых решение должно относиться к некоторому моменту в будущем.

Пример. Задано 10 классов объектов (рис. 4.2а). Требуется определить *признаки таблицы обучения, пороги* и построить оценки близости для классов объектов, показанных на рис. 4.2б. Предлагаются следующие *признаки таблицы обучения*:

- x₁- количество вертикальных линий минимального размера;
- x₂- количество горизонтальных линий;
- x₃- количество наклонных линий;
- x₄- количество горизонтальных линий снизу объекта.

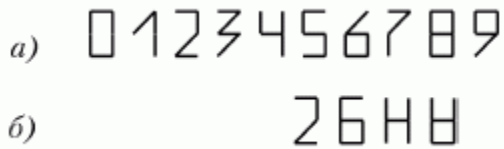


Рис. 4.2. Пример задачи по распознаванию
 На рис. 4.3 приведена *таблица обучения* и пороги $\epsilon_1=1, \epsilon_2=1, \epsilon_3=1, \epsilon_4=1, \delta=1$.

Из этой таблицы видно, что неразличимость символов 6 и 9 привела к необходимости ввода еще одного признака x₄.

	X ₁	X ₂	X ₃	X ₄	
	4	2	0		0
	2	0	1		1
	1	2	1		2
	0	2	2		3
	3	1	0		4
	2	3	0		5
✓	2	2	1	1	6
	1	1	1		7
	4	3	0		8
✓	2	2	1	0	9
	$\epsilon_1=1$	$\epsilon_2=1$	$\epsilon_3=1$	$\epsilon_4=1$	$\delta=1$

Рис. 4.3. Таблица обучения для задачи по распознаванию
 Теперь может быть построена *таблица распознавания* для объектов на рис. 4.2б.

	Объект	Результат распознавания
	x ₁	
	x ₂	
	x ₃	
	x ₄	
Объект 1		
1		
2		
1		
Цифра 2		
Объект 2		
3		
3		
0		
1		
Цифра 8 или 5		
Объект 3		
4		
1		
0		

Объект 4

4
2
0
1

Читателю предлагается самостоятельно ответить на вопрос: что будет, если увеличить *пороги* $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \delta$? Как изменится качество *распознавания* в данной задаче?

Актуальные задачи распознавания.

Среди множества интересных задач по распознаванию (распознавание отпечатков пальцев, распознавание по радужной оболочке глаза, распознавание машиностроительных чертежей и т. д.) следует выделить **задачу определения реальных координат заготовки и определения шероховатости обрабатываемой поверхности**, рассмотренную в лекции 10. Другой актуальной задачей является *распознавание машинописных и рукописных текстов* в силу ее повседневной необходимости. Практическое значение задачи *машинного чтения печатных и рукописных текстов* определяется необходимостью представления, хранения и использования в электронном виде огромного количества накопленной и вновь создающейся текстовой информации. Кроме того, большое значение имеет оперативный ввод в информационные и управляющие системы информации с машиночитаемых бланков, содержащих как напечатанные, так и рукописные тексты. В связи с этим рассмотрим принципы и подход к *распознаванию в задаче машинного чтения печатных и рукописных текстов*, описанные в работе. Для решения данной задачи используются следующие основные принципы.

1. Принцип целостности - распознаваемый объект рассматривается как единое целое, состоящее из структурных частей, связанных между собой пространственными отношениями.
2. Принцип двунаправленности - создание модели ведется от изображения к модели и от модели к изображению.
3. Принцип предвидения заключается в формировании гипотезы о содержании изображения. Гипотеза возникает при взаимодействии процесса "сверху-вниз", разворачивающегося на основе модели среды, модели текущей ситуации и текущего результата восприятия, и процесса "снизу-вверх", основанного на непосредственном грубом *признаковом* восприятии.
4. Принцип целенаправленности, включающий сегментацию изображения и совместную интерпретацию его частей.
5. Принцип "не навреди" - ничего не делать до распознавания и вне распознавания, то есть без "понимания".
6. Принцип максимального использования модели проблемной среды.

Указанные принципы реализованы в пакете программ "Графит", в программах FineReader-рукопись и FormReader - для распознавания рукописных символов и, частично, в программе FineReader для *распознавания печатных текстов*. Входящая в FormReader программа *чтения рукописных текстов* была выпущена в 1998 году одновременно с системой АВВУ FineReader 4.0. Эта программа может читать все рукописные строчные и заглавные символы, допускает ограниченные соприкосновения символов между собой и с графическими линиями и обеспечивает поддержку 10 языков. Основное применение программы - распознавание и ввод информации с машиночитаемых бланков. В системе АВВУ FormReader при *распознавании рукописных текстов* используются структурный, растровый, *признаковый*, дифференциальный и лингвистический уровни распознавания. Для более подробного освоения подходов к *распознаванию машинописных и рукописных текстов* в системе АВВУ FormReader читателю рекомендуется непосредственно ознакомиться с работой А. Шамиса, при этом знание основ машинной графики на уровне подразумевается. С другими работами по распознаванию читатель может познакомиться в литературе. Завершая этот раздел лекции,

отметим особенности задачи зрительного восприятия роботов по сравнению с традиционными задачами *распознавания образов* и машинной обработки изображений:

- необходимость построения комплексного описания среды на основе учета значительной априорной информации (модели проблемной среды) в отличие от традиционной задачи выделения фиксированных *признаков* или измерения отдельных параметров;
- необходимость анализа трехмерных сцен не только в плане анализа трехмерных объектов по их плоским проекциям, но и в плане определения объемных пространственных отношений;
- необходимость анализа изображений, включающих одновременно несколько произвольно расположенных объектов (в общем случае произвольной формы) в отличие от традиционной задачи, когда для распознавания предъявляется, как правило, один объект;
- необходимость анализировать реальную динамическую среду, а не статические изображения;
- отсутствие постоянной фиксированной задачи и необходимость оперативно решать возникающие по ходу дела задачи;
- необходимость следить за изменениями в среде, которые могут порождать новые оперативные задачи;
- необходимость организации системного процесса взаимодействия в реальном времени нескольких подсистем робота ("глаз-мозг", "глаз-мозг-рука").

Системы представления знаний

Традиционно, *системы представления знаний (СПЗ)* для ИС используют следующие основные виды *моделей*: *фреймы*, *исчисления предикатов*, *системы продукций*, *семантические сети*, нечеткие множества. Рассмотрим эти *модели* подробно.

Фрейм (рамка в переводе с англ.) - это единица представления знаний, запомненная в прошлом, детали которой могут быть изменены согласно текущей ситуации. *Фрейм* представляет собой структуру данных, с помощью которых можно, например, описать обстановку в комнате или место встречи для проведения совещания. С помощью *фреймов* можно описать ситуацию, сценарий, роль, структуру и т.д.

Фрейм отражает основные свойства объекта или явления. Структура *фрейма* записывается в виде списка свойств, называемых во *фрейме* слотами. Рассмотрим запись *фрейма* на языке FRL (Frame Representation Language) [32] - языке, похожем на LISP, но только внешне из-за наличия скобок.

Например, *фрейм* СТОЛ может быть записан в виде 3 слотов: слот НАЗНАЧЕНИЕ (purpose), слот ТИП (type) и слот ЦВЕТ (colour) следующим образом:

(frame СТОЛ

(purpose (value(размещение предметов для
деятельности рук)))

(type (value(письменный)))

(colour (value (коричневый))))

Во *фрейме* СТОЛ представлены только ДЕКЛАРАТИВНЫЕ средства для описания объекта, и такой *фрейм* носит название *фрейм-образец*. Однако существуют также *фреймы-экземпляры*, которые создаются для отображения фактических ситуаций на основе поступающих данных и ПРОЦЕДУРАЛЬНЫХ средств (демонов), например, следующих:

IF-DEFAULT - по умолчанию

IF-NEEDED - если необходимо

IF-ADDED - если добавлено

IF-REMOVED - если удалено

Слот IS-A или АКО (A Kind Of) определяет иерархию *фреймов* в сети *фреймов*. Такая связь обеспечивает наследование свойств. Слот *isa* указывает на *фрейм* более высокого уровня, откуда неявно наследуются свойства аналогичных слотов.

Рассмотрим фрагмент описания из "мира блоков" (рис. 2.1) в виде *фреймов*.

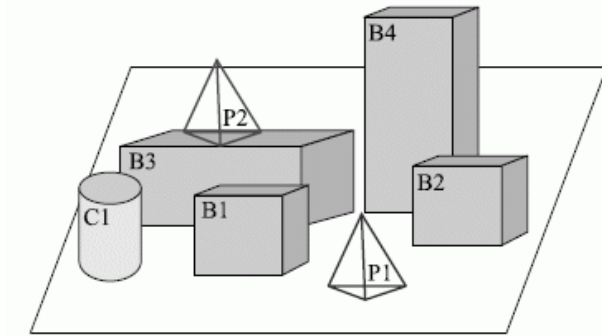


Рис. 2.1. "Мир блоков"

```
(frame (name (Cube))
  (isa (Block World))
  (length (NULL))
  (width (IF-DEFAULT (use length)))
  (height (IF-DEFAULT (use length))))
(frame (name (B1))
  (isa (Cube))
  (color (red))
  (length (80)))
(frame (name (B2))
  (isa (Cube))
  (color (green))
  (length (65))
  (who_put (value (NULL))
    (IF_NEEDED (askuser))))
```

Слот *isa* указывает на то, что объекты B1 и B2 являются подтипом объекта *Cube* и наследуют его свойства, а именно, $length = width = height$. Демон *IF_NEEDED* запускается автоматически, если понадобится узнать, кто поставил B2 на стол. Полученный ответ (Робби) будет подставлен в значение слота *who_put*. Аналогично работают демоны *IF-ADDED* и *IF-REMOVED*.

Допустим, однорукому роботу Робби дается приказ "Возьми желтый предмет, который поддерживает пирамиду". На языке представления знаний (ЯПЗ) вопрос записывается так:

```
(object ? X
  (color (yellow))
  (hold ? Y
    (type (pyramid))))
```

Программа сопоставления с образцом находит в базе знаний описание объектов:

```
(frame (name (B3))
  (type (block))
  (color (yellow))
  (size (20 20 20))
  (coordinate (20 50 0))
  (hold (P2)))
```

и

```
(frame (name (P2))
  (type (pyramid))
  ...)
```

Ответ получен $X = B3$, $Y = P2$ и Робби выдается команда `take(object=B3)`.

Исчисления предикатов

Традиционная булева алгебра и исчисление высказываний [33] не всегда подходят для выражения логических рассуждений, проводимых людьми, более удобен для этого язык логики предикатов. Под *исчислением предикатов* понимается формальный язык для представления отношений в некоторой предметной области. Основное преимущество *исчисления предикатов* - хорошо понятный мощный механизм математического вывода, который может быть непосредственно запрограммирован.

Предикатом называют предложение, принимающее только два значения: "истина" или "ложь". Для обозначения предикатов применяются логические связки между высказываниями:

\neg - не, \vee - или, \wedge - и, \supset - если, а также квантор \exists существования и квантор всеобщности \forall

$\exists x(\dots)$ - существует такой x , что ...

$\forall x(\dots)$ - для любого x

Таким образом, логика предикатов оперирует логическими связками между высказываниями, например, она решает вопросы: можно ли на основе высказывания A получить высказывание B и т.д.

Рассмотрим некоторые примеры. Высказывание "у каждого человека есть отец" можно записать:

$\forall x \exists y (\text{человек}(x) \supset \text{отец}(y,x))$

Выражение "Джон владеет красной машиной" записывается, например, так:

$\exists x (\text{владеет}(\text{Джон}, x) \supset \text{машина}(x) \wedge \text{красный}(x))$

Рассмотрим вывод, дающий заключение на основе двух предпосылок:

Предпосылка 1: Все люди смертны

$\forall x (\text{человек}(x) \supset \text{смертен}(x))$

$\forall x (p(x) \supset q(x))$

Предпосылка 2: Сократ - человек

$p(a)$

Заключение: Сократ - смертен

$\text{Смертен}(\text{Сократ})$

$q(a)$

Если обозначить через f функцию одного аргумента, то логическая формула для этого высказывания будет иметь вид:

$\forall x (f(x) \supset q(x))$

Алфавит логики предикатов состоит из элементов (символов):

x, y, z, u, v, w - переменные;

a, b, c, d, e - константы;

f, g, h - функциональные символы;

p, q, r, s, t - предикатные символы;

$\neg, \vee, \wedge, \supset, \forall, \exists$ - логические символы.

Запишем на языке *исчисления предикатов* некоторое выражение:

$\exists y \forall x (\text{человек}(x) \supset \text{отец}(y,x))$

Что означает записанное выражение? Ответ очевиден: "у всех людей общий отец".

Приведем пример простого доказательства на языке *исчисления предикатов*.

Даны следующие факты:

1. "Иван является отцом Михаила" - $\text{отец}(a, b)$

2. "Петр является отцом Василия" - $\text{отец}(c, d)$

3. "Иван и Петр являются братьями" -

$\exists w (\text{брат}(a,c) \supset \text{отец}(w,a) \wedge \text{отец}(w,c))$

Даны следующие определения:

4. "Брат отца является дядей" -

$\exists y (\text{дядя}(x,u) \supset \text{отец}(y,u) \wedge \text{брат}(y,x))$

5. "Сын дяди является двоюродным братом" -

$\exists x$ (дв.брат(z,u) \supset дядя(x,u) \wedge отец(x,z))

Требуется доказать, что "Михаил и Василий являются двоюродными братьями":

6. $\exists x \exists y$ (дв.брат(b, d) \supset отец(y, b) \wedge брат(y, x) \wedge отец(x, d))

Делаем подстановки $y =$ Иван, $b =$ Михаил и $x =$ Петр, $d =$ Василий, видим, что предикаты 1, 2, 3 дают правильное предложение 6.

Рассмотренный нами язык называется *исчислением предикатов* первого порядка и позволяет связывать знаком квантора переменные, соответствующие объектам из предметной области, но не предикаты или функции.

Исчисление предикатов второго порядка позволяет связывать знаком квантора не только переменные, соответствующие объектам из предметной области, но и предикаты или функции. Примером *исчисления предикатов* второго порядка может служить выражение "Единственное качество Джона - это честность", которое записывается так:

$\exists P$ (P(Джон) \wedge качество(P) \rightarrow P = честность)

Системы продукций

Под продукцией будем понимать выражение:

Если $\langle X_1, X_2 \dots X_n \rangle$ то

$\langle \{Y_1, D_1\}, \dots \{Y_m, D_m\} \rangle$,

где: X_i, Y_i - логические выражения, D_i - фактор достоверности (0,1) или фактор уверенности (0,100).

Системы продукций - это набор правил, используемый как база знаний, поэтому его еще называют базой правил. В Стэндфордской теории фактор уверенности CF (certainty factor) принимает значения от +1 (максимум доверия к гипотезе) до -1 (минимум доверия).

А.Ньюэлл и Г.Саймон отмечали в GPS, что продукции соответствуют навыкам решения задач человеком в долгосрочной памяти человека. Подобно навыкам в долгосрочной памяти эти продукции не изменяются при работе системы. Они вызываются по "образцу" для решения данной специфической проблемы. Рабочая память *продукционной системы* соответствует краткосрочной памяти, или текущей области внимания человека. Содержание рабочей области после решения задачи не сохраняется.

Работа *продукционной системы* инициируется начальным описанием (состоянием) задачи. Из продукционного множества правил выбираются правила, пригодные для применения на очередном шаге. Эти правила создают так называемое конфликтное множество. Для выбора правил из конфликтного множества существуют стратегии разрешения конфликтов, которые могут быть и достаточно простыми, например, выбор первого правила, а могут быть и сложными эвристическими правилами. Продукционная модель в чистом виде не имеет механизма выхода из тупиковых состояний в процессе поиска. Она продолжает работать, пока не будут исчерпаны все допустимые продукции. Практические реализации *продукционных систем* содержат механизмы возврата в предыдущее состояние для управления алгоритмом поиска.

Рассмотрим пример использования *продукционных систем* для решения шахматной задачи хода конем в упрощенном варианте на доске размером 3 x 3. Требуется найти такую последовательность ходов конем, при которой он ставится на каждую клетку только один раз ([рис. 2.2](#)).

Записанные на рисунке предикаты move(x,y) составляют базу знаний (базу фактов) для задачи хода конем. Продукционные правила - это факты перемещений move, первый параметр которых определяет условие, а второй параметр определяет действие (сделать ход в поле, в которое конь может перейти). Продукционное множество правил для такой задачи приведено ниже.

P1: If (конь в поле 1) then (ход конем в поле 8)

P2: If (конь в поле 1) then (ход конем в поле 6)

P3: If (конь в поле 2) then (ход конем в поле 9)

P4: If (конь в поле 2) then (ход конем в поле 7)

- P5: If (конь в поле 3) then (ход конем в поле 4)
- P6: If (конь в поле 3) then (ход конем в поле 8)
- P7: If (конь в поле 4) then (ход конем в поле 9)
- P8: If (конь в поле 4) then (ход конем в поле 3)
- P9: If (конь в поле 6) then (ход конем в поле 1)
- P10: If (конь в поле 6) then (ход конем в поле 7)
- P11: If (конь в поле 7) then (ход конем в поле 2)
- P12: If (конь в поле 7) then (ход конем в поле 6)
- P13: If (конь в поле 8) then (ход конем в поле 3)
- P14: If (конь в поле 8) then (ход конем в поле 1)
- P15: If (конь в поле 9) then (ход конем в поле 2)
- P16: If (конь в поле 9) then (ход конем в поле 4)

			<i>move(1, 8)</i>	<i>move(6, 1)</i>
1	2	3	<i>move(1, 6)</i>	<i>move(6, 7)</i>
			<i>move(2, 9)</i>	<i>move(7, 2)</i>
4	5	6	<i>move(2, 7)</i>	<i>move(7, 6)</i>
			<i>move(3, 4)</i>	<i>move(8, 3)</i>
7	8	9	<i>move(3, 8)</i>	<i>move(8, 1)</i>
			<i>move(4, 9)</i>	<i>move(9, 2)</i>
			<i>move(4, 3)</i>	<i>move(9, 4)</i>

Рис. 2.2. Шахматная доска 3x3 для задачи хода конем с допустимыми ходами
 Допустим, необходимо из исходного состояния (поле 1) перейти в целевое состояние (поле 2).
 Итерации *продукционной системы* для этого случая игры показаны в [таблице 2.1](#).

Таблица 2.1. Итерации для задачи хода конем

	№ итерации	Текущее поле	Целевое поле	Конфликтное множество	Активация правила
	1				
	1				
	2				
	1, 2				
	1				
	2				
	8				
	2				
	13, 14				
	13				
	3				
	3				
	2				
	5, 6				
	5				
	4				
	4				
	2				
	7, 8				
	7				
	5				
	9				
	2				
	15, 16				

15
6
2
2

Выход

Производственные системы могут породить бесконечные циклы при поиске решения. В *производственной системе* эти циклы особенно трудно определить, потому что правила могут активизироваться в любом порядке. Например, если в 4-й итерации выбирается правило 8, мы попадаем в поле 3 и зацикливаемся. Самая простая стратегия разрешения конфликтов сводится к тому, чтобы выбирать первое соответствующее перемещение, которое ведет в еще не посещаемое состояние. Следует также отметить, что конфликтное множество это простейшая база целей. В следующей лекции мы рассмотрим различные стратегии поиска в *производственных системах* и пути разрешения конфликтов. В заключение данного раздела лекции перечислим основные преимущества *производственных систем*:

- простота и гибкость выделения знаний;
- отделение знаний от программы поиска;
- модульность производственных правил (правила не могут "вызывать" другие правила);
- возможность эвристического управления поиском;
- возможность трассировки "цепочки рассуждений";
- независимость от выбора языка программирования;
- производственные правила являются правдоподобной моделью решения задачи человеком.

Семантические сети

Семантика в бытовом понимании означает смысл слова, художественного произведения, действия и т.д. **Семантическая сеть (СС)** - это граф, дуги которого есть отношения между вершинами (значениями). *Семантические сети* появились как *модель СПЗ* при решении задач разбора и понимания смысла естественного языка.

Пример *семантической сети* для предложения типа "Поставщик осуществил поставку изделий по заказу клиента до 1 июня 2004 года в количестве 1000 штук" приведен на [рис. 2.3](#).

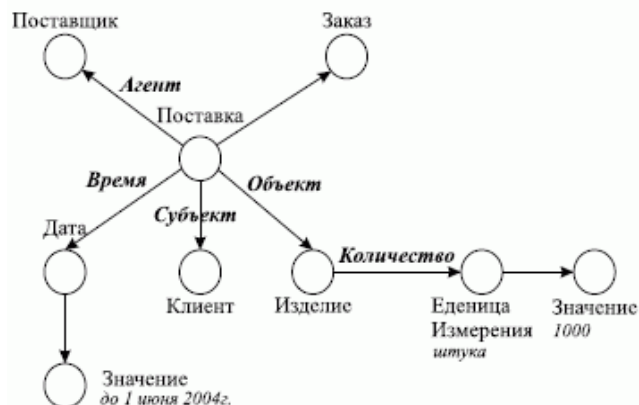


Рис. 2.3. Пример семантической сети

На этом примере видно, что между объектами Поставщик и Поставка определено отношение "агент", между объектами Изделие и Поставка определено отношение "объект" и т.д.

Число отношений, используемых в конкретных *семантических сетях*, может быть самое разное.

Агент - это то, что (тот, кто) вызывает действие. Агент часто является подлежащим в предложении, например, "**Робби** ударил мяч".

Объект - это то, на что (на кого) направлено действие. В предложении объект часто выполняет роль прямого дополнения, например, "Робби взял желтую **пирамиду**".

Инструмент - то средство, которое используется агентом для выполнения действия, например, "Робби открыл дверь **с помощью ключа**".

Соагент служит как подчиненный партнер главному агенту, например, "Робби собрал кубики **с помощью Суззи**".

Пункт отправления и пункт назначения - это отправная и конечная позиции при перемещении агента или объекта: "Робби перешел **из комнаты в библиотеку**".

Траектория - перемещение от пункта отправления к пункту назначения: "Они прошли **через дверь по ступенькам на лестницу**".

Средство доставки - то в чем или на чем происходит перемещение: "Он всегда едет домой на **метро**".

Местоположение - то место, где произошло (происходит, будет происходить) действие, например, "Он работал **за столом**".

Потребитель - то лицо, для которого выполняется действие: "Робби собрал кубики **для Суззи**".

Сырье - это, как правило, материал, из которого что-то сделано или состоит. Обычно сырье вводится предлогом из, например, "Робби собрал Суззи **из интегральных схем**".

Время - указывает на момент совершения действия: "Он закончил свою работу **поздно вечером**".

Наиболее типичный способ вывода в *семантических сетях (СС)* - это способ сопоставления частей сетевой структуры. Это видно на следующем простом примере, представленном на [рис. 2.4](#).

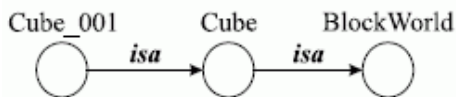


Рис. 2.4. Процедура сопоставления в СС

Куб Cube принадлежит миру BlockWorld.

Куб Cube_001 есть разновидность куба Cube.

Легко сделать вывод:

Куб Cube_001 есть часть мира BlockWorld.

Еще один пример поиска в СС. Представим вопрос "какой объект находится на желтом блоке?" в виде подсети, изображенной на [рис. 2.5](#). Произведем сопоставление вопроса с сетью, представленной на [рис. 2.6](#). В результате сопоставления получается ответ - "Пирамида".

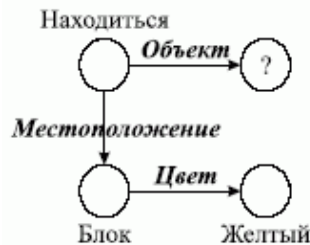


Рис. 2.5. Вопрос в виде СС



Рис. 2.6. Процедура сопоставления в СС

Нечеткая логика

При формализации знаний достаточно часто встречаются качественные знания, например, высокая температура при гриппе, слабое свечение нити накаливания, молодой дипломат и т.д. Для формального представления таких качественных знаний предложен формальный аппарат нечеткой (fuzzy) логики.

Нечеткое подмножество N множества M определяется как множество упорядоченных пар $N = \{\mu_N(x)/x\}$, где $\mu_N(x)$ - характеристическая функция принадлежности (или просто функция принадлежности), принимающая значения в интервале $[0, 1]$ и указывающая степень

(или уровень) принадлежности элемента x подмножеству N . Таким образом, нечеткое множество N можно записать как

$$N = \sum_{i=1}^n (\mu(X_i) / X_i),$$

где X_i - i -е значение базовой шкалы, а знак "+" не является обозначением операции сложения, а имеет смысл объединения.

Определим лингвистическую переменную (ЛП) как переменную, значение которой определяется набором словесных характеристик некоторого свойства. Например, ЛП "возраст" может иметь значения

ЛП = МлВ, ДВ, ОВ, ЮВ, МВ, ЗВ, ПВ, СВ ,

обозначающие возраст младенческий, детский, отроческий, юношеский, молодой, зрелый, преклонный и старый, соответственно. Множество M - это шкала прожитых человеком лет [0..120]. Функция принадлежности определяет, насколько мы уверены, что данное количество прожитых лет можно отнести к данному значению ЛП. Допустим, что неким экспертом к молодому возрасту отнесены люди в возрасте 20 лет со степенью уверенности 0,8, в возрасте 25 лет со степенью уверенности 0,95, в возрасте 30 лет со степенью уверенности 0,95 и в возрасте 35 лет со степенью уверенности 0,7. Итак:

$$\mu(X_1)=0,8; \mu(X_2)=0,95; \mu(X_3)=0,95; \mu(X_4)=0,7;$$

Значение ЛП=МВ можно записать:

$$МВ = \mu(X_1) / X_1 + \mu(X_2) / X_2 + \mu(X_3) / X_3 + \mu(X_4) / X_4 =$$

$$= 0,8 / X_1 + 0,95 / X_2 + 0,95 / X_3 + 0,7 / X_4 .$$

Таким образом, нечеткие множества позволяют учитывать субъективные мнения отдельных экспертов. Для большей наглядности покажем множество МВ графически при помощи функции принадлежности ([рис. 2.7](#)).

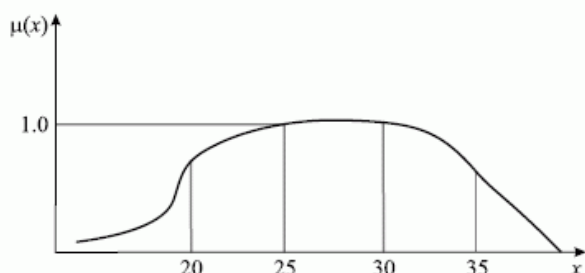


Рис. 2.7. График функции принадлежности

Для операций с нечеткими множествами существуют различные операции, например, операция "нечеткое ИЛИ" (иначе) задается в логике Заде [\[39\]](#), [\[40\]](#):

$$\mu(x) = \max(\mu_1(x), \mu_2(x))$$

и при вероятностном подходе так:

$$\mu(x) = \mu_1(x) + \mu_2(x) - \mu_1(x) \cdot \mu_2(x).$$

Существуют и другие операции над нечеткими числами, такие как расширенные бинарные арифметические операции (сложение, умножение и пр.) для нечетких чисел, определяемые через соответствующие операции для четких чисел с использованием принципа обобщения и т.д.

Как мы увидим в дальнейшем, нечеткие множества (другое название - мягкие вычисления) очень часто применяются в экспертных системах. Нечеткая логика применяется как удобный инструмент для управления технологическими и промышленными процессами, для интеллектуального домашнего хозяйства и электроники развлечения, в системах обнаружения ошибок и других экспертных системах. Разработаны специальные средства нечеткого вывода, например, инструментальное средство Fuzzy CLIPS. Развитием этого направления является реализации в *системах представления знаний* НЕ-факторов: неполнота, неточность, недоопределенность, неоднозначность, некорректность и др.

Кроме традиционных языков (LISP, PROLOG, SMALLTALK, РЕФАЛ) и инструментальных средств (LOOPS, KEE, ART) для представления знаний в настоящее время появляются новые веб-ориентированные версии ИС. Весьма популярными стали средства на базе JAVA: системы Exsys Corvid, JESS. Язык HTML явился основой для представления знаний в среде Интернет.

Методы поиска решений

Традиционными *методами поиска решений* в ИС считаются: *методы поиска в пространстве состояний* на основе различных *эвристических алгоритмов*, *методы поиска на основе предикатов* (*метод резолюции* и др.), *поиск решений в продукционных системах*, *поиск решений в семантических сетях* и т. д. Рассмотрим эти методы подробно.

Методы поиска решений в пространстве

Методы поиска решений в пространстве состояний начнем рассматривать с простой задачи о миссионерах и людоедах. Три миссионера и три людоеда находятся на левом берегу реки и им нужно переправиться на правый берег, однако у них имеется только одна лодка, в которую могут сесть лишь 2 человека. Поэтому необходимо определить план, соблюдая который и курсируя несколько раз туда и обратно, можно переправить всех шестерых. Однако если на любом берегу реки число миссионеров будет меньше, чем число людоедов, то миссионеры будут съедены. Решения принимают миссионеры, людоеды их выполняют.

Основой метода являются следующие этапы.

1. Определяется конечное число состояний, одно из состояний принимается за начальное и одно или несколько состояний определяются как искомое (конечное, или терминальное). Обозначим состояние S тройкой $S = (x, y, z)$, где x и y - число миссионеров и людоедов на левом берегу, $z = \{L, R\}$ - положение лодки на левом (L) или правом (R) берегах. Итак, начальное состояние $S_0 = (3, 3, L)$ и конечное (терминальное) состояние $S_k = (0, 0, R)$.
2. Заданы правила перехода между группами состояний. Введем понятие действия $M: [u, v]w$, где u - число миссионеров в лодке, v - число людоедов в лодке, w - направление движения лодки (R или L).
3. Для каждого состояния заданы определенные условия допустимости (оценки) состояний: $x \geq y; 3-x \geq 3-y; u+v \leq 2$.
4. После этого из текущего (исходного) состояния строятся переходы в новые состояния, показанные на [рис. 3.1](#). Два новых состояния следует сразу же вычеркнуть, так как они ведут к нарушению условий допустимости (миссионеры будут съедены).
5. При каждом переходе в новое состояние производится оценка на допустимость состояний и если при использовании правила перехода для текущего состояния получается недопустимое состояние, то производится возврат к тому предыдущему состоянию, из которого было достигнуто это текущее состояние. Эта процедура получила название бэктрекинг (bac tracing или **BACKTRACK**).

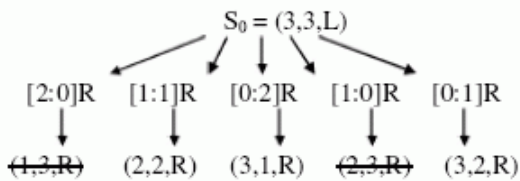


Рис. 3.1. Переходы из исходного состояния

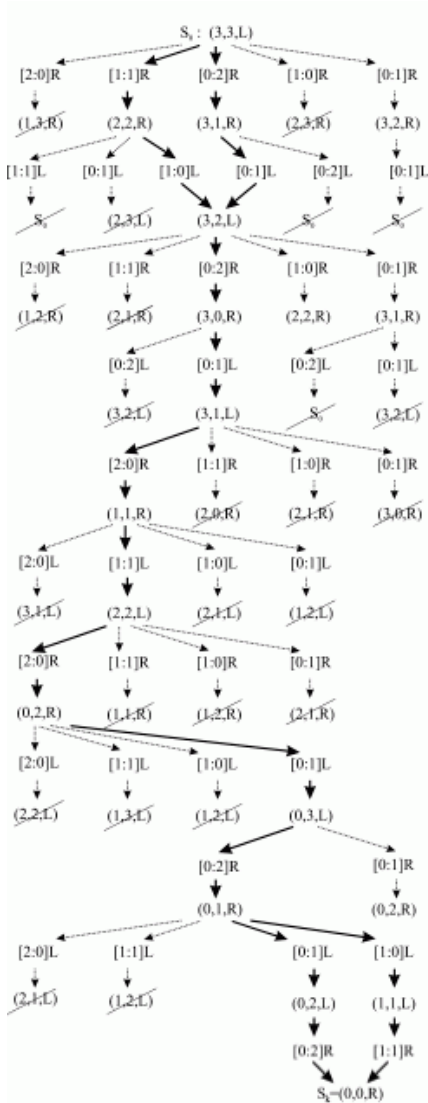


Рис. 3.2. Метод поиска в пространстве состояний

Теперь мы можем проанализировать полностью алгоритм простейшего *поиска решений в проблемном пространстве*, описанный группами состояний и переходами между состояниями на [рис. 3.2](#). Решение задачи выделено на [рис. 3.2](#) жирными стрелками. Такой *метод поиска* $S_0 \Rightarrow S_k$ называется *прямым методом поиска*. Поиск $S_k \Rightarrow S_0$ называют *обратным поиском*. Поиск в двух направлениях одновременно называют *двунаправленным поиском*.

Как уже упоминалось, фундаментальным понятием в *методах поиска* в ИС является идея рекурсии и процедура *BACKTRACK*. В качестве примера многоуровневого возвращения рассмотрим задачу размещения на доске 8 x 8 восьми ферзей так, чтобы они не смогли "съесть" друг друга.

Допустим, мы находимся на шаге размещения ферзя в 6 ряду и видим, что это невозможно. Процедура *BACKTRACK* пытается переместить ферзя в 5 строке и в 6 строке опять неудача. Только возврат к 4 строке и нахождение в ней нового варианта размещения приведет к решению задачи. Читатель сам может завершить решение этой задачи на основе процедуры *BACKTRACK*.

X							
	X						
		X					
X							
	X						



Алгоритмы эвристического поиска

В рассмотренных примерах поиска решений число состояний невелико, поэтому перебор всех возможных состояний не вызвал затруднений. Однако при значительном числе состояний время поиска возрастает экспоненциально, и в этом случае могут помочь алгоритмы эвристического поиска, которые обладают высокой вероятностью правильного выбора решения. Рассмотрим некоторые из этих алгоритмов.

Алгоритм наискорейшего спуска по дереву решений

Пример построения более узкого дерева рассмотрим на примере задачи о коммивояжере. Торговец должен побывать в каждом из 5 городов, обозначенных на карте (рис. 3.3).

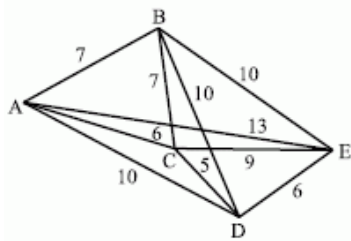


Рис. 3.3.

Задача состоит в том, чтобы, начиная с города А, найти минимальный путь, проходящий через все остальные города только один раз и приводящий обратно в А. Идея метода исключительно проста - из каждого города идем в ближайший, где мы еще не были. Решение задачи показано на рис. 3.4.

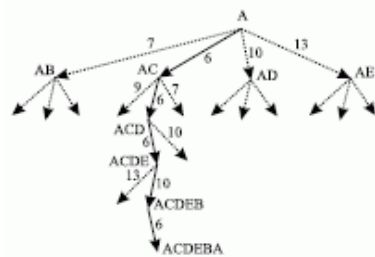


Рис. 3.4.

Такой алгоритм поиска решения получил название *алгоритма наискорейшего спуска* (в некоторых случаях - наискорейшего подъема).

Алгоритм оценочных (штрафных) функций

Умело подобранные оценочные функции (в некоторых источниках - штрафные функции) могут значительно сократить полный перебор и привести к решению достаточно быстро в сложных задачах. В нашей задаче о людоедах и миссионерах в качестве самой простой целевой функции при выборе очередного состояния можно взять число людоедов и миссионеров, находящихся "не на месте" по сравнению с их расположением в описании целевого состояния. Например, значение этой функции $f=x+y$ для исходного состояния $f_0=6$, а значение для целевого состояния $f_1=0$.

Эвристические процедуры поиска на графе стремятся к тому, чтобы минимизировать некоторую комбинацию стоимости пути к цели и стоимости поиска. Для задачи о людоедах введем оценочную функцию:

$$f(n) = d(n) + w(n)$$

где $d(n)$ - глубина вершины n на дереве поиска и $w(n)$ - число находящихся не на нужном месте миссионеров и людоедов. Эвристика заключается в выборе минимального значения $f(n)$.

Определяющим в эвристических процедурах является выбор оценочной функции.

Рассмотрим вопрос о сравнительных характеристиках оценочных целевых функций на примере функций для игры в "8" ("пятнашки"). Игра в "8" заключается в нахождении

минимального числа перестановок при переходе из исходного состояния в конечное (терминальное, целевое).

2	8	3
1	6	4
7	*	5
1	2	3
8	*	4
7	6	5

Рассмотрим две оценочные функции:

$$h_1(n) \& = Q(n)$$

$$h_2(n) \& = P(n) + 3S(n),$$

где $Q(n)$ - число фишек не на месте; $P(n)$ - сумма расстояний каждой фишки от места в ее целевой вершине; $S(n)$ - учет последовательности нецентральных фишек (штраф +2 если за фишкой стоит не та, которая должна быть в правильной последовательности; штраф +1 за фишку в центре; штраф 0 в остальных случаях).

Сравнение этих оценочных функций приведено в [таблица 3.1](#).

Таблица 3.1. Сравнение оценочных функций

Оценочная функция h	Стоимость (длина) пути L	Число вершин, открытых при нахождении пути n	Трудоемкость вычислений, необходимых для подсчета h s	Примечания
h_1 S_0	5	13	8	Поиск в ширину
S_1	>18	100-8!(=40320)		
h_2 S_0	5	11	8*2+8+1+1	Поиск в глубину
S_1	18	43		

На основе сравнения этих двух оценочных функций можно сделать выводы.

- Основу алгоритма поиска составляет выбор пути с минимальной оценочной функцией.
- Поиск в ширину, который дает функция h_1 , гарантирует, что какой-либо путь к цели будет найден. При поиске в ширину вершины раскрываются в том же порядке, в котором они порождаются.
- Поиск в глубину управляется эвристической компонентой $3S(n)$ в функции h_2 и при удачном выборе оценочной функции позволяет найти решение по кратчайшему пути (по минимальному числу раскрываемых вершин). Поиск в глубину тем и характеризуется, что в нем первой раскрывается та вершина, которая была построена самой последней.
- Эффективность поиска возрастает, если при небольших глубинах он направляется в основном в глубь эвристической компонентой, а при возрастании глубины он больше похож на поиск вширь, чтобы гарантировать, что какой-либо путь к цели будет найден. Эффективность поиска можно определить как $E=K/L*N*S$, где K и S (трудоемкость) - зависят от оценочной функции, L - длина пути, N - число вершин, открытых при нахождении пути. Если договориться, что для оптимального пути $E=1$, то $K=L^0*N^0*S^0$.

Алгоритм минимакса

Метод минимакса нашел применение в теории игр. Предположим, что противник использует оценочную функцию (ОФ), совпадающую с нашей ОФ. Выбор хода с нашей стороны определяется максимальным значением ОФ для текущей позиции. Противник стремится сделать ход, который минимизирует ОФ. Поэтому этот метод и получил название минимакса. На [рис. 3.5](#) приведен пример анализа дерева ходов с помощью *метода минимакса* (выбранный путь решения отмечен жирной линией).

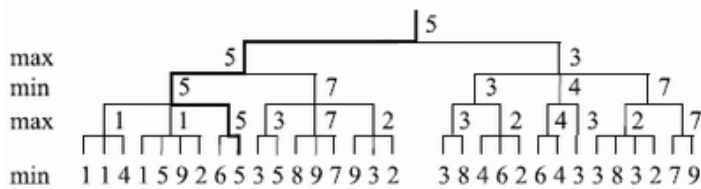


Рис. 3.5. Дерево ходов

Развивая *метод минимакса*, назначим вероятности для выполняемых действий в задаче о миссионерах и людоедах:

$$P([2 : 0]R) = 0; 8; P([1 : 1]R) = 0; 5;$$

$$P([0 : 2]R) = 0; 9;$$

$$P([1 : 0]R) = 0; 3; P([0 : 1]R) = 0; 3;$$

Интуитивно понятно, что посылать одного людоеда или одного миссионера менее эффективно, чем двух человек, особенно на начальных этапах. На каждом уровне мы будем выбирать состояние по критерию P_i . Даже такой простой подход позволит нам избежать части тупиковых состояний в процессе поиска и сократить время по сравнению с полным перебором. Кстати, этот подход достаточно распространен в экспертных продукционных системах.

Альфа-бета-процедура

Теоретически, это эквивалентная минимаксу процедура, с помощью которой всегда получается такой же результат, но заметно быстрее, так как целые части дерева исключаются без проведения анализа. В основе этой процедуры лежит идея Дж. Маккарти об использовании двух переменных, обозначенных α и β (1961 год).

Основная идея метода состоит в сравнении наилучших оценок, полученных для полностью изученных ветвей, с наилучшими предполагаемыми оценками для оставшихся. Можно показать, что при определенных условиях некоторые вычисления являются лишними. Рассмотрим идею отсечения на примере [рис. 3.6](#). Предположим, позиция А полностью проанализирована и найдено значение ее оценки α . Допустим, что один ход из позиции Y приводит к позиции Z, оценка которой по *методу минимакса* равна z. Предположим, что $z \leq \alpha$. После анализа узла Z, когда справедливо соотношение $y \leq z \leq \alpha \leq s$, ветви дерева, выходящие из узла Y, могут быть отброшены (альфа-отсечение).

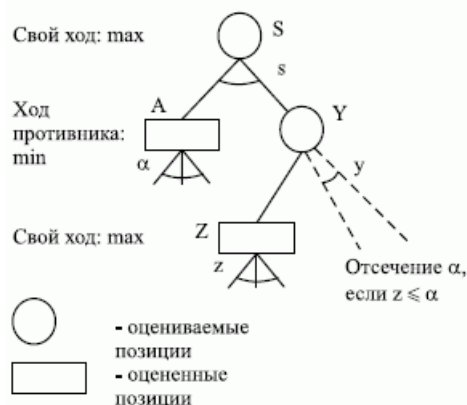


Рис. 3.6. - отсечение

Если мы захотим опуститься до узла Z, лежащего на уровне произвольной глубины, принадлежащей той же стороне, что и уровень S, то необходимо учитывать минимальное значение оценки β , получаемой на ходах противника.

Отсечение типа β можно выполнить всякий раз, когда оценка позиции, возникающая после хода противника, превышает значение β . Алгоритм поиска строится так, что оценки своих ходов и ходов противника сравниваются при анализе дерева с величинами α и β соответственно. В начале вычислений этим величинам присваиваются значения $+\infty$ и $-\infty$, а затем, по мере продвижения к корню дерева, находится оценка начальной позиции и наилучший ход для одного из противников.

Правила вычисления α и β в процессе поиска рекомендуются следующие:

1. у МАХ вершины значение α равно наибольшему в данный момент значению среди окончательных возвращенных значений для ее дочерних вершин;
2. у MIN вершины значение β равно наименьшему в данный момент значению среди окончательных возвращенных значений для ее дочерних вершин.

Правила прекращения поиска:

3. можно не проводить поиска на поддереве, растущем из всякой MIN вершины, у которой значение β не превышает значения α всех ее родительских МАХ вершин;
4. можно не проводить поиска на поддереве, растущем из всякой МАХ вершины, у которой значение α не меньше значения β всех ее родительских MIN вершин.

На рис. 3.7 показаны α - β отсечения для конкретного примера. Таким образом, α - β -алгоритм дает тот же результат, что и метод минимакса, но выполняется быстрее.

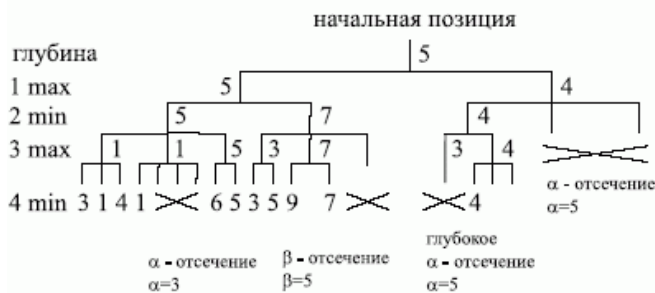


Рис. 3.7. α - β отсечение для конкретного примера

Использование алгоритмов эвристического поиска для поиска на графе И, ИЛИ выигрышной стратегии в более сложных задачах и играх (шашки, шахматы) не реален. По некоторым оценкам игровое дерево игры в шашки содержит 10^{40} вершин, в шахматах 10^{120} вершин. Если при игре в шашки для одной вершины требуется $1/3$ наносекунды, то всего игрового времени потребуется 10^{21} веков. В таких случаях вводятся искусственные условия остановки, основанные на таких факторах, как наибольшая допустимая глубина вершин в дереве поиска или ограничения на время и объем памяти.

Методы поиска решений на основе исчисления предикатов

Семантика исчисления предикатов обеспечивает основу для формализации логического вывода. Возможность логически выводить новые правильные выражения из набора истинных утверждений очень важна. Логически выведенные утверждения корректны, и они совместимы со всеми предыдущими интерпретациями первоначального набора выражений. Обсудим вышесказанное неформально и затем введем необходимую формализацию.

В исчислении высказываний основным объектом является переменное высказывание (предикат), истинность или ложность которого зависит от значений входящих в него переменных. Так, истинность предиката "x был физиком" зависит от значения переменной x. Если x - П. Капица, то предикат истинен, если x - М. Лермонтов, то он ложен. На языке исчисления предикатов утверждение $\forall x(P(x) \Rightarrow Q(x))$ читается так: "для любого x если P(x), то имеет место и Q(x)". Иногда его записывают и так: $\forall x (P(x) \rightarrow Q(x))$. Выделенное подмножество тождественно истинных формул (или правильно построенных формул - ППФ), истинность которых не зависит от истинности входящих в них высказываний, называется аксиомами.

В исчислении предикатов имеется множество правил вывода. В качестве примера приведем классическое правило отделения, или modus ponens :

$$(A, A \rightarrow B) / B$$

которое читается так "если истинна формула A и истинно, что из A следует B, то истинна и формула B". Формулы, находящиеся над чертой, называются посылками вывода, а под чертой - заключением. Это правило вывода формализует основной закон дедуктивных систем: из истинных посылок всегда следуют истинные заключения. Аксиомы и правила вывода исчисления предикатов первого порядка задают основу формальной дедуктивной системы, в

которой происходит формализация схемы рассуждений в логическом программировании. Можно упомянуть и другие правила вывода.

Modus tollendo tollens : Если из А следует В и В ложно, то и А ложно.

Modus ponendo tollens : Если А и В не могут одновременно быть истинными и А истинно, то В ложно.

Modus tollendo ponens : Если либо А, либо В является истинным и А не истинно, то В истинно.

Решаемая задача представляется в виде утверждений (аксиом) $f_1, F_2... F_n$ исчисления предикатов первого порядка. Цель задачи В также записывается в виде утверждения, справедливость которого следует установить или опровергнуть на основании аксиом и правил вывода формальной системы. Тогда решение задачи (достижение цели задачи) сводится к выяснению логического следования (выводимости) целевой формулы В из заданного множества формул (аксиом) $f_1, F_2... F_n$. Такое выяснение равносильно доказательству общезначимости (тождественно-истинности) формулы

$f_1 \& F_2 \& \dots \& F_n \rightarrow B$

или невыполнимости (тождественно-ложности) формулы

$f_1 \& F_2 \& \dots \& F_n \& \neg B$

Из практических соображений удобнее использовать доказательство от противного, то есть доказывать невыполнимость формулы. На доказательстве от противного основано и ведущее правило вывода, используемое в логическом программировании, - **принцип резолюции**.

Робинсон открыл более сильное правило вывода, чем modus ponens, которое он назвал *принципом резолюции* (или правилом резолюции). При использовании *принципа резолюции* формулы исчисления предикатов с помощью несложных преобразований приводятся к так называемой дизъюнктивной форме, то есть представляются в виде набора дизъюнктов. При этом под дизъюнктом понимается дизъюнкция литералов, каждый из которых является либо предикатом, либо отрицанием предиката.

Приведем пример дизъюнкта:

$\forall x (P(x, c_1) \supset Q(x, c_2))$.

Пусть Р - предикат уважать, c_1 - Ключевский, Q - предикат знать, c_2 - история. Теперь данный дизъюнкт отражает факт "каждый, кто знает историю, уважает Ключевского".

Приведем еще один пример дизъюнкта:

$\forall x (P(x, c_1) \& P(x, c_2))$.

Пусть Р - предикат знать, c_1 - физика, c_2 - история. Данный дизъюнкт отражает запрос "кто знает физику и историю одновременно".

Таким образом, условия решаемых задач (факты) и целевые утверждения задач (запросы) можно выразить в дизъюнктивной форме логики предикатов первого порядка. В дизъюнктах кванторы всеобщности \forall , \exists , обычно опускаются, а связки \supset , \neg , \wedge заменяются на \rightarrow импликацию.

Вернемся к *принципу резолюции*. Главная идея этого правила вывода заключается в проверке того, содержит ли множество дизъюнктов R пустой (ложный) дизъюнкт. Обычно резолюция применяется с прямым или обратным методом рассуждения. Прямой метод из посылок А, $A \rightarrow B$ выводит заключение В (правило modus ponens). Основной недостаток прямого метода состоит в его не направленности: повторное применение метода приводит к резкому росту промежуточных заключений, не связанных с целевым заключением. Обратный вывод является направленным: из желаемого заключения В и тех же посылок он выводит новое подцелевое заключение А. Каждый шаг вывода в этом случае связан всегда с первоначально поставленной целью. Существенный недостаток *метода резолюции* заключается в формировании на каждом шаге вывода множества резольвент - новых дизъюнктов, большинство из которых оказывается лишними. В связи с этим разработаны различные модификации *принципа резолюции*, использующие более эффективные *стратегии поиска* и различного рода ограничения на вид исходных дизъюнктов. В этом смысле наиболее удачной и популярной является система ПРОЛОГ, которая использует специальные виды дизъюнктов, называемых дизъюнктами Хорна.

Процесс доказательства *методом резолюции* (от обратного) состоит из следующих этапов:

1. Предложения или аксиомы приводятся к дизъюнктивной нормальной форме.
2. К набору аксиом добавляется отрицание доказываемого утверждения в дизъюнктивной форме.
3. Выполняется совместное разрешение этих дизъюнктов, в результате чего получаются новые основанные на них дизъюнктивные выражения (резольвенты).
4. Генерируется пустое выражение, означающее противоречие.
5. Подстановки, использованные для получения пустого выражения, свидетельствуют о том, что отрицание отрицания истинно.

Рассмотрим примеры применения *методов поиска решений на основе исчисления предикатов*. Итак, заданы утверждения 1-4 в левом столбце [таблица 3.2](#) Требуется ответить на вопрос: "Существует ли человек, живущий интересной жизнью?" В виде предикатов эти утверждения записаны во втором столбце таблицы. Предполагается, что $\forall X(\text{smart}(X) \rightarrow \neg \text{stupid}(X))$ и $\forall Y(\text{wealthy}(Y) \rightarrow \neg \text{poor}(Y))$. В третьем столбце таблицы записаны дизъюнкты.

Утверждения и заключение	Предикаты	Предложения(дизъюнкты)
1. Все небедные и умные люди счастливы	$\exists X(\neg \text{poor}(X) \wedge \text{smart}(X) \rightarrow \text{happy}(X))$	$\text{poor}(X) \wedge \neg \text{smart}(X) \& \text{happy}(X)$
2. Человек, читающий книги, - неглуп	$\forall Y(\text{read}(Y) \rightarrow \text{smart}(Y))$	$\neg \text{read}(Y) \& \text{smart}(Y)$
3. Джон умеет читать и является состоятельным человеком	$\text{read}(\text{John}) \wedge \neg \text{poor}(\text{John})$	3a $\text{read}(\text{John})$ 3b $\neg \text{poor}(\text{John})$
4. Счастливые люди живут интересной жизнью	$\forall Z(\text{happy}(Z) \rightarrow \text{exciting}(Z))$	$\neg \text{happy}(Z) \& \text{exciting}(Z)$
5. Заключение: Существует ли человек, живущий интересной жизнью?	$\exists W(\text{exciting}(W))$	$\text{exciting}(W)$
6. Отрицание заключения	$\neg \exists W(\text{exciting}(W))$	$\neg \text{exciting}(W)$

Отрицание заключения имеет вид (строка 6): $\neg \exists W(\text{exciting}(W))$

Одно из возможных доказательств (их более одного) дает следующую последовательность резольвент:

1. $\neg \text{happy}(Z)$ резольвента 6 и 4
2. $\text{poor}(X) \wedge \neg \text{smart}(X)$ резольвента 7 и 1
3. $\text{poor}(Y) \wedge \neg \text{read}(Y)$ резольвента 8 и 2
4. $\neg \text{read}(\text{John})$ резольвента 9 и 3b
5. NIL резольвента 10 и 3a

Символ NIL означает, что база данных выражений содержит противоречие и поэтому наше предположение, что не существует человек, живущий интересной жизнью, неверно.

В *методе резолюции* порядок комбинации дизъюнктивных выражений не устанавливался. Значит, для больших задач будет наблюдаться экспоненциальный рост числа возможных комбинаций. Поэтому в процедурах резолюции большое значение имеют также эвристики поиска и различные *стратегии*. Одна из самых простых и понятных *стратегий* - *стратегия* предпочтения единичного выражения, которая гарантирует, что резольвента будет меньше, чем наибольшее родительское выражение. Ведь в итоге мы должны получить выражение, не содержащее литералов вообще.

Среди других *стратегий* (поиск в ширину (breadth-first), *стратегия* "множества поддержки", *стратегия* линейной входной формы) *стратегия* "множества поддержки" показывает отличные результаты при поиске в больших пространствах дизъюнктивных выражений. Суть *стратегии* такова. Для некоторого набора исходных дизъюнктивных выражений S можно указать подмножество T , называемое множеством поддержки. Для реализации этой *стратегии* необходимо, чтобы одна из резольвент в каждом опровержении имела предка из множества поддержки. Можно доказать, что если S - невыполнимый набор дизъюнктов, а $S-T$ - выполнимый, то *стратегия* множества поддержки является полной в смысле опровержения. С другими *стратегиями* для поиска методом резолюции в больших пространствах дизъюнктивных выражений читатель может познакомиться в специальной литературе.

Исследования, связанные с доказательством теорем и разработкой алгоритмов опровержения резолюции, привели к развитию языка логического программирования PROLOG (Programming in Logic). PROLOG основан на теории предикатов первого порядка. Логическая программа - это набор спецификаций в рамках формальной логики. Несмотря на то, что в настоящее время удельный вес языков LISP и PROLOG снизился и при решении задач ИИ все больше используются C, C++ и Java, однако многие задачи и разработка новых *методов решения задач* ИИ продолжают опираться на языки LISP и PROLOG. Рассмотрим одну из таких задач - задачу *планирования последовательности действий* и ее решение на основе теории предикатов.

Задачи планирования последовательности действий

Многие результаты в области ИИ достигнуты при решении "задач для робота". Одной из таких простых в постановке и интуитивно понятных задач является задача *планирования последовательности действий*, или задача построения планов.

В наших рассуждениях будут использованы примеры традиционной робототехники (современная робототехника во многом основывается на реактивном управлении, а не на планировании). Пункты плана определяют атомарные действия для робота. Однако при описании плана нет необходимости опускаться до микроуровня и говорить о датчиках, шаговых двигателях и т. п. Рассмотрим ряд предикатов, необходимых для работы планировщика из мира блоков. Имеется некоторый робот, являющийся подвижной рукой, способной брать и перемещать кубики. Рука робота может выполнять следующие задания (U, V, W, X, Y, Z - переменные).

goto(X, Y, Z)перейти в местоположение X, Y, Z

pickup(W)взять блок W и держать его

putdown(W)опустить блок W в некоторой точке

stack(U, V)поместить блок U на верхнюю грань блока V

unstack(U, V)убрать блок U с верхней грани блока V

Состояния мира описываются следующим множеством предикатов и отношений между ними.

on(X, Y)блок X находится на верхней грани блока Y

clear(X)верхняя грань блока X пуста

gripping(X)захват робота удерживает блок X

gripping()захват робота пуст

ontable(W)блок W находится на столе

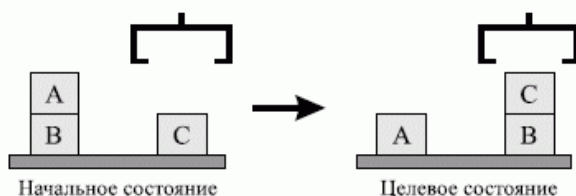


Рис. 3.8. Начальное и целевое состояния задачи из мира кубиков

Предметная область из мира кубиков представлена на [рис. 3.8](#) в виде начального и целевого состояния для решения задачи *планирования*. Требуется построить *последовательность действий робота*, ведущую (при ее реализации) к достижению целевого состояния.

Состояния мира кубиков представим в виде предикатов. Начальное состояние можно описать следующим образом:

```
start = [handempty, ontable(b),
         ontable(c), on(a,b), clear(c),
         clear(a)]
```

где: handempty означает, что рука робота Робби пуста.

Целевое состояние записывается так:

```
goal = [handempty, ontable(a),
        ontable(b), on(c,b), clear(a),
        clear(c)]
```

Теперь запишем правила, воздействующие на состояния и приводящие к новым состояниям.

$$(\forall X) (\text{pickup}(X) \rightarrow (\text{gripping}(X) \leftarrow (\text{gripping}() \wedge \text{clear}(X) \wedge \text{ontable}(X))))$$
$$(\forall X) (\text{putdown}(X) \rightarrow ((\text{gripping}() \wedge \text{ontable}(X) \wedge \text{clear}(X)) \leftarrow \text{gripping}(X)))$$
$$(\forall X) (\forall Y) (\text{stack}(X, Y) \rightarrow ((\text{on}(X, Y) \wedge \text{gripping}() \wedge \text{clear}(X)) \leftarrow (\text{clear}(Y) \wedge \text{gripping}(X))))$$
$$(\forall X) (\forall Y) (\text{unstack}(X, Y) \rightarrow ((\text{clear}(Y) \wedge \text{gripping}(X)) \leftarrow (\text{on}(X, Y) \wedge \text{clear}(X) \wedge \text{gripping}()))$$

Прежде чем использовать эти правила, необходимо упомянуть о проблеме границ. При выполнении некоторого действия могут изменяться другие предикаты и для этого могут использоваться аксиомы границ - правила, определяющие инвариантные предикаты. Одно из решений этой проблемы предложено в системе STRIPS.

Поиск решений в системах продукций

Поиск решений в системах продукций наталкивается на проблемы выбора правил из *конфликтного множества*, как это указывалось в предыдущей лекции. Различные варианты решения этой проблемы рассмотрим на примере ЭСО CLIPS, на которой нам предстоит в 7 лекции разработать исследовательский прототип ЭС. Правила в ЭС, кроме фактора уверенности эксперта, имеют приоритет выполнения (*salience*). **Конфликтное множество (KM)** - это список всех правил, имеющих удовлетворенные условия при некотором, текущем состоянии списка фактов и объектов и которые еще не были выполнены. Как отмечалось ранее, *конфликтное множество* это простейшая база целей. Когда активизируется новое правило с определенным приоритетом, оно помещается в список правил KM ниже всех правил с большим приоритетом и выше всех правил с меньшим приоритетом. Правила с высшим приоритетом выполняются в первую очередь. Среди правил с одинаковым приоритетом используется определенная *стратегия*.

CLIPS поддерживает семь *стратегий* разрешения конфликтов.

Стратегия глубины (*depth strategy*) является *стратегией* по умолчанию (*default strategy*) в CLIPS. Только что активизированное правило помещается поверх всех правил с таким же приоритетом. Это позволяет реализовать поиск в глубину.

Стратегия ширины (*breadth strategy*) - только что активизированное правило помещается ниже всех правил с таким же приоритетом. Это, в свою очередь, реализует поиск в ширину.

LEX стратегия - активация правила, выполненная более новыми образцами (фактами), располагается перед активацией, осуществленной более поздними образцами. Например, как это указано в [таблица 3.3](#) ниже.

МЕА стратегия - сортировка образцов не производится, а осуществляется только упорядочение правил по первым образцам, как это показано в столбце 3 [таблица 3.3](#).

Таблица 3.3. Результаты применения LEX и MEA стратегий

Исходный набор правил	Правила, отсортированные LEX	Правила, отсортированные MEA
rule-6: f-1,f-4	rule-6: f-4,f-1	rule-2: f-3,f-1
rule-5: f-1,f-2,f-3	rule-5: f-3,f-2,f-1	rule-3: f-2,f-1
rule-1: f-1,f-2,f-3	rule-1: f-3,f-2,f-1	rule-6: f-1,f-4
rule-2: f-3,f-1	rule-2: f-3,f-1	rule-5: f-1,f-2,f-3
rule-4: f-1,f-2	rule-4: f-2,f-1	rule-1: f-1,f-2,f-3
rule-3: f-2,f-1	rule-3: f-2,f-1	rule-4: f-1,f-2

*Стратегия*упрощения (simplicity strategy) - среди всех правил с одинаковым приоритетом только что активизированное правило располагается выше всех правил с равной или большей определенностью (specificity). Определенность правила задается количеством сопоставлений в левой части правил плюс количество вызовов функций. Логические функции не увеличивают определенность правила.

*Стратегия*усложнения (complexity strategy) - среди всех правил с одинаковым приоритетом только что активизированное правило располагается выше всех правил с равной или большей определенностью.

Случайная *стратегия* (random strategy) - каждой активации назначается случайное число, которое используется для определения местоположения среди активаций с определенным приоритетом.

Основы принятия решений и ситуационного моделирования

Принятие решения и целеполагающая ресурсоориентированная деятельность человека в социальной, экономической, политической, идеологической, военной сферах тесно связаны. В них крайне нежелательны ошибки, которые могут привести к пагубным последствиям. Но из-за ограниченных информационных возможностей человека ошибки всегда возможны. Поэтому есть настоятельная необходимость применения научного подхода к обоснованию и *принятию решений*.

Принятие решений, наряду с прогнозированием, планированием, ситуационным анализом обстановки, исполнением решений, контролем и учетом является функцией управления. Все функции управления направлены так или иначе на формирование или реализацию решений, и любую функцию управления технологически можно представить в виде последовательности каких-либо связанных общей целью решений.

При прогнозировании и планировании принимаются решения, связанные с выбором методов и средств, организацией работы, оценкой достоверности информации, выбором наиболее достоверного варианта прогноза и наилучшего варианта плана. Таким образом, функция *принятия решений* является с методологической и технологической точек зрения более общей, чем другие функции управления. Для лица, принимающего решение (*ЛПР*), *принятие решений* является основной задачей, которую он обязан исполнять в процессе управления. Поэтому знание методов, технологий и средств решений этой задачи является необходимым элементом квалификации руководителя, базой для дальнейшего управления.

Конечным результатом любой задачи *принятия решений* становится решение, конструктивное предписание к действию. Решение является одним из видов мыслительной деятельности и имеет следующие признаки: имеется выбор из множества возможностей; выбор ориентирован на сознательное достижение целей; выбор основан на сформировавшейся установке к действию. Основной характеристикой решения является его эффективность, т.е. степень, темп достижения целей и затраты ресурсов для *принятия* и реализации *решения*. Решение тем эффективнее, чем больше степень достижения целей и меньше стоимость затрат.

Принятие решения - это выбор одного из множества рассматриваемых допустимых вариантов. Обычно их число конечно, а каждый вариант выбора определяет некоторый

результат (экономический эффект, прибыль, выигрыш, полезность, надежность и т.д.), допускающий количественную оценку. Такой результат обычно называется полезностью решения. Таким образом, ищется вариант с наибольшим значением полезности решения. Возможен и подход с минимизацией противоположной оценки, например, отрицательной величины полезности. Часто на практике встречается ситуация, когда каждому варианту решения соответствует единственный результат (детерминированность выбора решения), хотя возможны и другие случаи, например, когда каждому варианту i и условию j , характеризующему полезность, соответствует результат решения x_{ij} . Таким образом, можно говорить о матрице решений $||x_{ij}||$, $i=1,2,...,m$; $j=1,2,...,n$. Чтобы оценить решение, необходимо уметь оценивать все его последствия. Существуют различные подходы для такой оценки. Например, если решения альтернативные, то можно последствия каждого из них характеризовать суммой его наибольшего и наименьшего результатов, максимумом из возможных таких сумм, максимумом из максимумов по всем вариантам (оптимистическая позиция выбора), максимумом из среднего арифметического (нейтральная позиция выбора), максимумом из минимума (пессимистическая позиция) и другие.

Классические модели *принятия решений*, как правило, являются оптимизационными, ставящими целью максимизировать выгоду и на основе этих моделей получить практическую прибыль. Так как теоретиков больше интересует первая сторона, а практиков - вторая, то при разработке и использовании таких моделей необходимо их тесное сотрудничество. Практические рекомендации (решения) могут быть получены, если при построении модели *принятия решений* придать большее значение учету существенных структурных элементов моделируемой системы, т.е. разработке имитационной модели *принятия решений*, с привлечением экспериментальных, полужэкспериментальных и теоретических методов. Кроме классических, оптимизационных *процедур принятия решений* существуют и ряд базовых неклассических (неоклассических) *процедур, технологий принятия решений*, некоторые из которых мы рассмотрим.

Классификация задач *принятия решений* проводится по различным признакам. Наиболее существенными являются: степень определенности информации; использование эксперимента для получения информации; количество лиц, принимающих решения; содержание решений; направленность решений.

На процесс *принятия решения* часто воздействуют различные случайные (стохастические) параметры, усложняющие процедуру. Недостаток информации об их распределении (сложность их измерения) приводит к необходимости принятия каких-то гипотез как об области их изменения, так и о характере их распределения (о функции распределения вероятностей). Правильность используемых гипотез необходимо проверять с помощью методов оценки статистических гипотез. При отсутствии достаточной информации для такой процедуры приходится привлекать большое число типов распределения. Проблемы *принятия решений* с недетерминированными параметрами называют проблемами *принятия решений* в условиях недостатка информации. Чем меньше информации у нас, тем больше может оказаться различие между ожидаемым и действительным результатами принимаемых решений в целом. Мера влияния информации (параметров) на результат решения называется релевантностью. Особо важно в социально-экономической сфере *принятие решения* при наличии рисков (неплатежей, невозвратов кредитов, ухудшения условий жизни и т.д.).

Формализуемые решения принимаются на основе соответствующих математических методов (алгоритмов). Математическая модель задачи оптимизации формализуемого решения включает следующие элементы:

1. заданную оптимизируемую целевую функцию (критерий управляемости):
 $\Phi = F(x_1, x_2, \dots, x_n)$, где x_j ($j=1, 2, \dots, n$) - параметры, учитываемые при *принятии решения* (отражающие ресурсы *принятия решений*);
2. условия, отражающие ограниченность ресурсов и действий ЛПР при *принятии решений*: $g_i(x_j) < a_i$, $k_i(x_j) = b_i$; $c_j < x_j < d_i$, $i=1, 2, \dots, m$; $j=1, 2, \dots, n$.

Непременным требованием для решения задачи оптимизации является условие $n > m$.

В зависимости от критерия эффективности, стратегий и факторов управления выбирается тот или иной метод (алгоритм) оптимизации.

Основными являются следующие классы методов:

1. методы линейного и динамического программирования (*принятия решения* об оптимальном распределении ресурсов);
2. методы теории массового обслуживания (*принятие решения* в системе со случайным характером поступления и обслуживания заявок на ресурсы);
3. методы имитационного моделирования (*принятие решения* путем проигрывания различных ситуаций, анализа откликов системы на различные наборы задаваемых ресурсов);
4. методы теории игр (*принятие решений* с помощью определения стратегии в тех или иных состязательных задачах);
5. методы теории расписаний (*принятие решений* с помощью разработки календарных расписаний выполнения работ и использования ресурсов);
6. методы сетевого планирования и управления (*принятие решений* с помощью оценки и перераспределения ресурсов при выполнении проектов, изображаемых сетевыми графиками);
7. методы многокритериальной (векторной) оптимизации (*принятие решений* при условии существования многих критериев оптимальности решения)

и другие методы.

Выбор решения - заключительный и наиболее ответственный этап процесса *принятия решений*. Здесь *ЛПР* должно осмыслить полученную на этапах постановки задачи и формирования решений информацию и использовать ее для обоснования выбора. В реальных задачах *принятия решений* к началу этапа выбора решения еще сохраняется большая неопределенность, поэтому сразу осуществить выбор единственного решения из множества допустимых решений практически очень сложно. Поэтому используется принцип последовательного уменьшения неопределенности, который заключается в последовательном трехэтапном (обычно) сужении множества решений. На первом этапе исходное множество альтернативных решений Y сужается (используя ограничения на ресурсы) до множества приемлемых или допустимых решений $Y_1 \subseteq Y$. На втором этапе множество допустимых решений Y_1 сужается (учитывая критерий оптимальности) до множества эффективных решений $Y_2 \subseteq Y_1$. На третьем этапе осуществляется выбор (на основе критерия выбора и дополнительной информации, в том числе и экспертной) единственного решения $Y^* \in Y_2$.

Система принятия решений - совокупность организационных, методических, программно-технических, информационно-логических и технологических обеспечений *принятия решений* для достижения поставленных целей.

Общая *процедура принятия решений* может состоять из следующих этапов:

- анализ проблемы и среды (цели *принятия решения*, их приоритеты, глубина и ограничения рассмотрения, элементы, связи, ресурсы среды, критерии оценки);
- постановка задачи (определение спецификаций задачи, альтернатив и критериев выбора решения);
- выбор (адаптация, разработка) метода решения задачи;
- выбор (адаптация, разработка) метода оценки решения;
- решение задачи (математическая и компьютерная обработка данных, имитационные и экспертные оценки, уточнение и модификация, если это необходимо);
- анализ и интерпретация результатов.

Задачи *принятия решений* могут быть поставлены и решены в условиях детерминированных (определенности, формализованности и единственности целевой функции, ее количественной оцениваемости), риска (возможные решения, исходы распределены вероятностно) и недетерминированных (неопределенности, неточности, плохой формализуемости информации).

В моделях *принятия решений* используются различные процедуры. В частности, наиболее просты и эффективны следующие:

- методы математического программирования;
- методы кривых безразличия;
- многокритериального выбора альтернатив на основе четкого или же нечеткого отношения предпочтения;
- последовательной оценки и последующего исключения вариантов;
- многомерного ранжирования (шкалирования) объектов и другие.

При выборе рационального решения необходимо принимать во внимание внешнюю среду и побочные явления, динамическую изменчивость критериев оценок решения, необходимость ранжирования аспектов и приоритетов решения, их неполноту и разнородность (а иногда и конфликтность).

Продемонстрируем *ситуационное моделирование* на примере моделирования деятельности банка. Банковская система является одной из подсистем современной экономической системы, наиболее подверженной информатизации. Развитие банковской системы сопровождается постоянным поиском адекватных оптимальных методов и инструментов управления, *принятия решений* на основе экономико-математического анализа и моделирования деятельности банков. При этом необходимо учитывать тот факт, что финансовые операции имеют еще и стохастические составляющие, усложняющие и без того сложные процессы начисления процентных ставок, взносов и выплат, регулирования и управления, инвестиций и др. Эти процессы сложны не только динамически, но и вычислительно, логически. Кроме того, от таких прогнозов зависят и прогноз, анализ темпов инфляции, структуры активов и пассивов банка, доходности акций, курсов валют, процентная ставка и др.

Ситуационный анализ денежных потоков состоит в основном - часто на имитационном моделировании - анализе эффективности того или иного набора финансовых операций и процедур (из множества возможных и допустимых) путем сравнения результатов их воздействия на финансовые, денежные потоки с величиной финансовых, денежных активов без учета их воздействия. Следовательно, ситуационный анализ денежных потоков является динамическим процессом, использующим методы оптимизации и критерии оптимальности. При ситуационном анализе некоторых базовых значений величины активов (соответствующих определенным финансовым условиям и обязательствам, например, величине уставного капитала), можно по некоторым критериям оптимальности (целевым функциям оптимизации), выбрать оптимальный набор возможных, допустимых финансовых операций, обеспечивающих, например, наибольшую доходность. Возможно построение целевой функции максимизации с учетом ликвидности. Возможно также получение решения задач, свидетельствующего об отсутствии роста (или малого роста) каких-либо финансовых параметров, например, активов, из которого можно сделать вывод о невозможности проведения оптимизирующих операций (процедур).

Пусть d_t - средний уровень доходности, получаемый в результате проведения некоторых инвестиционных мероприятий, а P_t - процентная ставка на момент времени $t=0, 1, 2, \dots, T$. Тогда рост активов A будет осуществляться по закону

$$A = \sum_{t=1}^T P_t (1 + d_t)^t$$

и можно использовать при ситуационном анализе критерий эффективности:

$$\sum_{t=1}^T P_t (1 + d_t)^t \Rightarrow \max.$$

Соотношение между доходностью активов и ценой пассивов коммерческого банка является важнейшим показателем, который отражает эффективность денежно-финансовой политики банка.

Ситуационный анализ соотношения осложняется рядом факторов:

1. структура активов и пассивов могут отражаться ссудами различной длительности, а также различными схемами размещения и привлечения обязательств и ценных бумаг, например, возврат денег может быть осуществлен по схеме ежемесячного отчисления процентов и уплаты кредита в конце либо по схеме единовременного возврата суммы долга и процентов в конце промежутка кредитования;
2. необходимостью учета (прогноза) инфляционного ожидания и "увеличения" или "очистки" тех или иных составляющих активов и пассивов в зависимости от инфляции;
3. различными параметрами и факторами, влияющими на степень риска, затрудненностью оценки величины риска.

Различные структуры и схемы размещения и привлечения финансовых ресурсов определяют и различные динамические модели.

Например, если схема предусматривает возврат долга с процентами одновременно, реальная ставка рублевого кредита d может быть определена по формуле

$$d = (z - a) / (1 + a/100) (\%),$$

где z - номинальная ставка рублевого кредита (%), a - инфляция за период кредитования (%). Для валютного кредита, очищенного от инфляции, с учетом внутренней конвертируемости рубля:

$$d = [(1 + z/100)(1 + g/100) - (1 + a/100)] / (1 + a/100) 100 (\%),$$

где z - номинальная ставка валютного кредита (%), g - рост курса валюты за период кредитования (%).

Если же договор размещения кредитов предусматривает учет динамики возврата долга (части долга) и уплаты процентов, то реальная ставка может определяться следующей процедурой:

1. определяется динамика срочных выплат (части долга и процентов), гарантирующая полное выполнение обязательств за период кредитования, т.е. обеспечивающая выполнение условий

$$\sum_{t=1}^T g_t (1 + z)^{-t} = S$$

где g_t - ежемесячные (ежеквартальные, ежегодные) выплаты, t - номер месяца (квартала, года), в конце которого происходит выплата, S - размер ссуды, выданной в начале договора кредитования, T - количество дней (месяцев, кварталов, лет) кредитования;

2. задается динамика инфляции, например, дискретная функция $a_t = a(t)$, $t = 1, 2, \dots, T$;
3. определяется реальная ставка d - решение уравнения:

$$\sum_{t=1}^T g_t / \prod_{i=1}^t (1 + a_i / 100) (1 + d / 100) = S;$$

если кредит - валютный, то необходимо дополнить этапы 1-3 этой процедуры следующими этапами:

4. осуществляется прогноз роста курса валюты, т.е. определяется (задается) дискретная функция $g_t = g(t)$, $t = 1, 2, \dots, T$;
5. реальная ставка определяется из уравнения вида (S - ссуда в валюте):

$$\sum_{t=1}^T g_t / \prod_{i=1}^t (1 + a_i / 100) (1 + g_i / 100) (1 + d / 100) = S.$$

В долговременных финансово-кредитных операциях проценты либо выплачиваются сразу после их начисления, либо их реинвестируют, применяя сложные проценты. Исходная сумма S (база) увеличивается по принятому (кредитором и дебитором) соглашению, а для простых процентов база постоянная и равна начальной сумме S . Присоединение начисленных процентов к базовой сумме называется капитализацией процентов, $t = 0, \dots, T$.

Важнейшим показателем при ситуационном анализе и моделировании деятельности и жизнеспособности банка является надежность, банковский или кредитный риск. Надежность банка - не просто вероятность быть надежным банком в данный момент, а вероятность банка сохранять надежностные характеристики и отношения на некотором допустимом промежутке их варьирования и для определенного промежутка времени.

Пусть $x=(x_1, x_2, \dots, x_n) \in \Omega$ - вектор, характеризующий надежность банка, а Ω - некоторое множество его допустимых изменений. В качестве меры надежности можно взять условную вероятность $p=p(P/\Omega)$, где P - оценка (степень) надежности, P/Ω - оценка при условии изменения $x \in \Omega$.

Пример. Пусть $\Omega=\Omega(x_1, x_2, x_3)$ - информационные ресурсы, доступные объекту (субъекту), который производит анализ надежности банка, а $x=(x_1, x_2, x_3)$, где x_1 - активы банка, x_2 - пассивы банка, x_3 - дебиторская задолженность банку. Пусть, например, мы хотим оценить надежность банка, но не имеем о банке информации (или имеем нулевую информацию). Тогда значение $p(P/\Omega)$ можно получить, только исходя из двух возможных равновероятных состояний - банк либо надежен, либо не надежен, т.е. $p(P/\Omega)=0,5$. Результат мало информативен и может быть применен к любому банку при любых условиях Ω . Пусть теперь известно, что существует лишь 30 % надежных банков, т.е. мы при оценке надежности банка используем эту информацию. В этом случае можно оценить надежность банка как $0 \leq p(P/\Omega) \leq 0,3$. В то же время, как и для предыдущего случая, такая оценка надежности будет малоинформативной, так как здесь мы имеем, как и в первом случае, два возможных состояния ($p \leq 0,3$ и $p > 0,3$) и по формуле Шеннона количество информации в том и в другом случае равно

$$I = \log_2 N = \log_2 2 = 1 \text{ (бит)}.$$

Чем более точной информацией о банке владеет вкладчик (дебитор), тем проще ему можно принимать верные решения, т.е. тем чаще и ближе будут оценки вероятности (надежности) p к $p=0$ и $p=1$. Чем меньше информации, тем сложнее принять однозначное решение, тем чаще и ближе будет оценка вероятности к $p=0,5$ ("пятьдесят на пятьдесят").

Величину $p(P/\Omega)$ принято называть апостериорной вероятностью (a posteriori - после опыта).

Под опытом здесь подразумевается процесс получения информации Ω , следовательно, $p(P/\Omega)$ - вероятность быть надежным банком с учетом полученной в результате опыта информации.

При определении надежности (например, экспертами) могут допускаться ошибки, в том числе и субъективного характера. Это - вероятность "ложной классификации". Пусть p_1 - вероятность отнесения (априори) надежного банка в класс ненадежных, а p_2 - вероятность отнесения (априори) ненадежного банка в класс надежных банков. Если не учитывать гипотез о степени их предпочтения (рейтинг банка), то показатель качества классификации - сумма вероятностей совершения ошибок, т.е. $p=p_1+p_2$. Можно снабдить их весами (предпочтения) a_1 и a_2 , например, если $a_1=1$, $a_2=2$, то вероятность p_2 в 2 раза важнее p_1 (иначе говоря, в 2 раза опаснее относить ненадежный банк в группу надежных, чем надежный банк в группу ненадежных). Тогда итоговый показатель является средневзвешенной суммой вероятностей:

$$p = a_1 q_1 + a_2 q_2,$$

где $a_1, a_2 \geq 0$, $q_1, q_2 \geq 0$, q_1, q_2 - вероятности ошибок, $q_1 = 1 - p_1$, $q_2 = 1 - p_2$.

Показатель p называют байесовским риском. Чем больше p , тем хуже произведена классификация, а чем она ближе к нулю, тем классификация ближе к реальной или априорной классификации.

Для ситуационного анализа необходимо иметь адекватные модели потока платежей. Как правило, этот поток - дискретный. Рассмотрим одну из простых подмоделей модели ситуационного анализа, дополняющую выше приведенную процедуру.

Пусть в момент времени $t_0=0$ имеется капитал $x(0)$ (денежных единиц), а в момент времени $t=t_1, t_2, \dots, t_n$ имеются транзакции (приход, расход) $y(t_i)$, $i=1, 2, \dots, n$. Рассмотрим, как это бывает на практике, одинаковые промежутки времени (год, месяц, день) $[t_0; t_1], [t_1; t_2], \dots, [t_{n-1}; t_n]$, т.е. $t_i - t_{i-1} = \text{const}$ и векторы $t=(0, t_1, t_2, \dots, t_n)$, $x=(x(0), x(t_1), x(t_2), \dots, x(t_n))$, $y=(0, y(t_1), y(t_2), \dots, y(t_n))$, $v=(0,$

$v(t_1), v(t_2), \dots, v(t_n)$), где $v(t_i)$ - коэффициент дисконта на промежутке времени $(0; t_i]$, т.е. коэффициент относительной скидки или отношения приращения ссуды (капитала) за срок от 0 до t_i к наращенной сумме. Тогда потоки доходов и расходов будут, соответственно, равны

$$P = x(0) + \sum_{i=1}^n x(t_i)v(t_i),$$

$$R = \sum_{i=1}^n y(t_i)v(t_i).$$

Будем считать доходы кредитора (инвестора) отрицательными величинами (отдает), а поступления - положительными. Тогда $z(0)=-x(0)$ - начальный доход (начальная величина инвестиций), а $z(t_i)=y(t_i)-x(t_i)$ - поступление на его счет, $i=1,2,\dots, n$.

Чистая стоимость потока $Q=R-P$ равна:

$$Q = \sum_{i=1}^n y(t_i)v(t_i) - x(0) - \sum_{i=1}^n x(t_i)v(t_i) = -x(0) + \sum_{i=1}^n [y(t_i) - x(t_i)]v(t_i) = -x(0) + \sum_{i=1}^n z(t_i)v(t_i)$$

Аналогично, чистое наращенное значение потока на момент времени $t_i > 0$ равно (вводя $a(t_j, t_i)$ - коэффициент наращивания на $(t_j; t_i], j=1, \dots, n-1$)

$$Q_i = \sum_{j=1}^i z(t_j)a(t_j, t_i).$$

Наращенное значение всех платежей к моменту времени $t_n=T$ равно Q_n .

Одним из эффективных механизмов принятия деловых решений (в проблемах инвестирования, выработки стратегии поведения, развития и т.д.) является использование ИСПР (просто СПР) - информационных систем поддержки решений (Decision Support Systems), сочетающих современные средства аналитической обработки и средства визуализации информации и технологии поддержки деятельности экспертной группы.

Пример. В области организационного управления наибольший интерес имеют так называемые ситуационные (эмерджентные) комнаты (центры), позволяющие быстро "погрузить" ЛПР в рассматриваемую проблемную ситуацию, обстановку, помочь разобраться в проблеме и принять локально-оптимальное (не обязательно глобально-оптимальное) решение. Например, президент США имеет несколько таких комнат. Существуют ситуационные центры Президента РФ, Совета Безопасности, МЧС. Ситуационные комнаты - это специальное место для поддержки построения, проигрывания проблемной ситуации и *принятия решений* одним человеком или группой людей. Эффект от использования ситуационной комнаты зависит от корректности поставленной проблемы, полноты и достоверности используемых данных, сценария обсуждения, технологий интеллектуальной и компьютерной поддержки (например, использования экспертных систем), временного интервала прогноза и др. Простое использование автоматизированной системы обработки документов, поисковых систем, средств визуализации и мультимедиа - недостаточные условия для функционирования ситуационной комнаты. Основная функция СПР - поддержка умственной, эвристической и творческой деятельности ЛПР. СПР может работать в следующих режимах:

1. проблемный мониторинг и актуализация информации (СМИ, органов власти, объектов управления и пр.) с целью текущего информирования и предупреждения о накапливающихся небольших негативных явлениях;
2. плано-аналитический режим - плановое заслушивание и обсуждение аналитических докладов по проблемной ситуации с целью поддержки и принятия заслушиваемого решения по заранее фиксированному сценарию подачи, демонстрации материала для анализа "вширь" и "вглубь";
3. чрезвычайный режим - оперативный мониторинг информации, принятие и контроль исполнения решений по непредвиденным, чрезвычайным проблемам с целью

уменьшения негативных факторов, влияющих на обычное в таких ситуациях совмещение построения сценария, обсуждения и *принятия решений*.

В базовом варианте, ситуационная комната может включать экран коллективного доступа; компьютер (обычно, ноутбук) с возможностью отображения на экран коллективного доступа; средства доступа к базе данных (знаний), в том числе - с целью сохранения сценария обсуждения, систему подготовки презентаций.

Задачи и упражнения

1. Требуется *принять решение* о том, когда необходимо проводить профилактический ремонт ЭВМ, чтобы минимизировать потери из-за неисправности. В случае, если ремонт будет производиться слишком часто, затраты на обслуживание будут большими при малых потерях из-за случайных поломок. Так как невозможно предсказать заранее, когда возникнет неисправность, необходимо найти вероятность того, что ПЭВМ выйдет из строя в период времени t . ЭВМ ремонтируется индивидуально, если она остановилась из-за поломки. Через t интервалов времени выполняется профилактический ремонт всех n ПЭВМ. Построить *процедуру принятия решения* о ремонте (исходя из различных ситуаций, в которые помещено ЛПП).
2. Интенсивность спроса x (спрос в единицу времени) на некоторый товар задается непрерывной функцией распределения $f(x)$. Если запасы в начальный момент невелики, возможен дефицит товара. В противном случае к концу рассматриваемого периода запасы нерезализованного товара могут оказаться большими. Потери возможны и в том, и в другом случае. Предложите *процедуру принятия решения* о необходимом запасе товаров.
3. При работе на ЭВМ необходимо периодически проверять наличие вирусов. Приостановка в обработке информации приводит к определенным экономическим издержкам. Если же вирус вовремя не будет обнаружен, возможна и потеря информации, и затраты на восстановление. Варианты решения таковы: E_1 - полная проверка; E_2 - минимальная проверка (проверка каталога); E_3 - отказ от проверки. ЭВМ может находиться в состояниях: F_1 - вирус отсутствует; F_2 - вирус есть, но он не успел активизироваться; F_3 - некоторые файлы испорчены вирусом и нуждаются в восстановлении. Предложите *процедуру принятия решения*. Организуйте группу и руководство по *ситуационному моделированию* для решения этой проблемы (для *принятия решений* по проблеме).

Модели знаний

Такие понятия как "интеллект", "интеллектуальность", у специалистов различного профиля (системного анализа, информатики, нейропсихологии, психологии, философии и др.) могут несколько различаться, причем это не несет в себе никакой опасности.

Примем, не обсуждая ее положительные и отрицательные стороны, следующую "формулу интеллекта":

"Интеллект = цель + факты + способы их применения",

или, в несколько более "математическом", формализованном виде:

"Интеллект = цель + аксиомы + правила вывода из аксиом".

При поиске наиболее удобных, рациональных средств и форм информационного обмена человек чаще всего сталкивается с проблемой компактного, однозначного и достаточно полного *представления знаний*.

Знания - система понятий и отношений для такого обмена. Можно условно классифицировать *знания* в предметной области на *понятийные, конструктивные, процедурные, фактографические знания* и *метазнания*.

Понятийные знания - набор понятий, используемых при решении данной задачи, например, в фундаментальных науках и теоретических областях наук, т.е. это понятийный аппарат науки.

Конструктивные знания - наборы структур, подсистем системы и взаимодействий между их элементами, например, в технике.

Процедурные знания - методы, процедуры (алгоритмы) их реализации и идентификации, например, в прикладных науках.

Фактографические - количественные и качественные характеристики объектов и явлений, например, в экспериментальных науках.

Метазнания - знания о порядке и правилах применения знаний (знания о знаниях).

Представление знаний есть процесс, конечная цель которого - представление информации (семантического смысла, значения) в виде информативных сообщений (синтаксических форм): фраз устной речи, предложений письменной речи, страниц книги, понятий справочника, объектов географической карты, мазков и персонажей картины и т.п.

Для этого необходимо пользоваться некоторой конструктивной системой правил для их представления и восприятия (прагматического смысла). Назовем такую систему правил **формализмом представления знаний**. Неформализуемые знания - это знания, получаемые с применением неизвестных (неформализуемых) правил, например, *эвристик*, интуиции, здравого смысла и принятия решений на их основе.

Человек пользуется естественным *формализмом* - языком, письменностью. Язык, языковые конструкции развиваются благодаря тому, что человеческие знания постоянно нуждаются в языковом представлении, выражении, сжатии, хранении, обмене. Мысль, которую нельзя выразить в языковой конструкции, не может быть включена в информационный обмен. Язык - форма *представления знаний*. Чем многообразнее язык народа, чем больше *знаний* он может отражать, тем богаче культура народа. В то же время, предложения и слова языка должны иметь однозначный семантический смысл. Особую роль играет язык математики как язык наук (не только точных, но и гуманитарных), формализации *знаний*, основа изложения системы *знаний* в естественных науках. Свой язык имеют химия, физика, экономика, информатика и т.д. Языки наук часто пересекаются и взаимообогащаются при исследовании междисциплинарных проблем.

Использование языковых систем и диалектов повышает надежность информационного обмена, снижая возможность неправильного истолкования передаваемой информации и уровень шумов в сообщениях. Главное назначение языка науки - создавать и использовать типовые, "стандартные" формы изложения, сжатия и хранения *знаний*, ликвидация полисемии (смысловой многозначности) естественного языка. Полисемия, обогащая естественный язык, делая его богаче и выразительнее, тем не менее, является в информационном обмене источником семантического шума, смысловой неоднозначности, а часто - и алогичности, неалгебраичности.

Пример. Найдем и формализуем закономерность в последовательности 1, 10, 11, 100, 111, 1000, 1111, 10000, Из сравнения членов $A[i]$ ($i=1,2,\dots$) последовательности, стоящих на четных местах и на нечетных местах, видно, что: 1) элемент на нечетном месте получается из элемента на предыдущем нечетном месте добавлением единицы справа к нему; 2) каждый элемент на четном месте получается из элемента на предыдущем четном месте добавлением справа к нему нуля. Это словесно описанное (неформализованное) правило можно записать на математическом языке, в аналитическом виде. Получим для случаев 1) и 2): $A[2n]=10A[2n-2]$, $A[2n-1]=10A[2n-1]+1$, $n=1, 2, \dots$. Можно записать формулу, объединяющую обе эти формулы: $A[2n+m]=10A[2n+n-2]+m$, где $m=0$ или $m=1$. Лучшая форма (с меньшей полисемией): $A[2n+\text{mod}(n,2)]=10A[2n+\text{mod}(n,2)-2]+\text{mod}(n,2)$.

Пример. Формализуем закон формирования последовательности: АВ, ААВ, АВВ, АААВ, АВВВ, Словесное описание правила имеет вид: к слову, стоящему на очередном нечетном месте, добавляется с конца символ "В", а к слову, стоящему на очередном четном месте слева, добавляется символ "А". "Формульная" запись правила: $X_{2n+1}=X_{2n-1}+В$, $X_{2n}=А+X_{2n-2}$, $n=1, 2, 3, \dots$. Здесь операция "+" означает конкатенацию (присоединение текста к тексту справа), а X_n - элемент последовательности на n-м месте.

Одной из важных форм (методов) формализации *знаний* является их представление классом (*классификация*).

Классификация - выделение некоторого критерия (некоторых критериев) распределения и группировка систем или процессов таким образом, что в одну группу попадают лишь те системы (процессы), которые удовлетворяют этому критерию (значению критерия). *Классификация* - это метод научной систематики, особенно важный на начальном этапе формирования базовых *знаний* научного направления. *Классификация*, установление эквивалентности объектов, систем позволяет решать такие важные задачи информатики как фиксация *знаний*, поиск по образцу, сравнение и др.

Пример. Такими системами являются классификационная система К. Линнея в ботанике, систематика живых организмов, таблица элементов Д. Менделеева, систематика экономических систем, механизмов, "табель о рангах", введенная Петром Первым в 1722 г. Эта табель подразделяла чины на 14 рангов. Каждому чину соответствовала определенная должность. Первые 6 рангов статской и придворной служб и первый обер-офицерский чин в армии давали право на получение потомственного дворянства, что способствовало формированию дворянской бюрократии. Таким образом, "табель о рангах" выполняла социально-экономическую *классификацию* определенной (определяющей) части общества, социально-экономическое стимулирующее упорядочивание.

Указанные выше классификационные системы - иерархические структуры (модели) *представления знаний*. Отдельные понятия, факты, *знания*, связаны между собой отношениями дедуктивного (от частного к общему), индуктивного (от общего к частному) или индуктивно-дедуктивного вывода и формализуются соответствующими формальными структурами: древовидными, морфологическими, реляционными и др.

Пример. Рассмотрим систему "Фирма". Опишем всех сотрудников фирмы в лексикографически упорядоченном списке с именем "Сотрудники", указывая табельный номер, ФИО, год рождения, образование, специальность, разряд, стаж работы. Этот список дает нам *знание* о коллективе, его возрастных и профессиональных качествах и др. Составим другой список - "Заработная плата", где укажем для каждого сотрудника условия оплаты, величину их заработка (стоимости единицы времени их работы). Этот список дает нам *знания* о системе оплаты фирмы, ее финансового состояния и др. Оба списка содержат необходимый объем *знаний* о трудовом коллективе, если цель исследования этой системы - начисление заработной платы. Здесь мы наблюдаем и древовидные, и морфологические, и реляционные модели *представления знаний*.

Для более строгой формализации (сложных и динамических) *знаний* в последнее время используют такой перспективный инструментарий, как *категории* и *функторы*. Впрочем, математическая сложность такого аппарата не дает применять его на первоначальных этапах формализации *знаний* и он чаще используется лишь тогда, когда *знания* получили достаточно полную математическую форму описания.

Появление и развитие объектно-ориентированных технологий и объектно-ориентированного проектирования, использующих близкие по духу идеи, тем не менее, актуализируют аппарат *категорий* и *функторов*, поэтому введем основные начальные понятия.

Категория $K = \langle S, M \rangle$ - это совокупность S элементов (компоненты, характеристики, параметры, свойства и другие параметры исследуемой системы), называемых объектами *категории*, и совокупность преобразований, морфизмов M - специального типа преобразований, которые позволяют описывать (определять), например, эквивалентность, инвариантность и другие свойства. Объекты и морфизмы связаны между собой так, что:

1. каждой упорядоченной паре объектов $A, B \in S$ сопоставлено множество $M(A, B)$ морфизмов из M ;
2. каждый морфизм $m \in M$ принадлежит только одному из множеств $M(A, B)$;
3. в классе морфизмов M введен закон композиции морфизмов: произведение $a \circ b$ морфизма $a \in M(A, B)$ на морфизм $b \in M(C, D)$ определено и принадлежит $M(A, B)$ тогда и

только тогда, когда объект $b \in X$ совпадает с объектом $c \in X$, причем композиция морфизмов ассоциативна: $(a \circ b) \circ c = a \circ (b \circ c)$;

4. в каждом множестве $M(A, A)$ содержится единичный или тождественный морфизм I_A :

$$\forall a \in M(X, A), \forall b \in M(A, Y), \forall A, X, Y \in F, \exists I_A: a \circ I_A = a, I_A \circ b = b.$$

Категории, их использование для представления знаний адекватны мыслительным процедурам человека, учитывающим опыт, интуицию, понимание мира в терминах *категорий*, которым мы затем приписываем реальные оболочки, конкретные структуры. Объекты *категории* могут быть связаны между собой, влиять друг на друга, даже если у них нет общего (формального) сходства, а свойства *категорий* отражают сущность способностей человека, его поведения в окружении.

Функтор - обобщение понятия *категории*. Для введения преобразования между *категориями* используем понятие *функтора*. *Функтор* - аналог семантической операции, т.е. преобразования информации, приводящего к появлению некоторого смыслового (семантического) содержания.

Функтор определяется парой отображений, которые сохраняют композицию морфизмов и тождественные отображения (сохраняют смысл информации при преобразованиях): одно отображение преобразует объекты S (грубо говоря, - информацию), а другое - преобразует морфизмы M (грубо говоря, - семантический смысл).

Самый плохо формализуемый в информатике процесс - это процесс образования семантического смысла. Строгая математическая основа аппарата *категорий* и *функторов* позволяет исследовать семантический смысл математически корректно (путем построения семантических сетей, анализа *фреймов*, продукционных правил и др.), что является необходимым условием формализации *знаний*, разработки баз *знаний* и систем интеллектуальной поддержки принятия решений.

Категорийно-функторный подход к проблеме формализации *знаний* позволяет формализовать многие интуитивно используемые понятия.

Пример. Формализуем, например, понятия "формула", "теория". Формула F_i - запись вида $R_i(k)(x_1, \dots, x_k)$, которую следует читать так: k переменных x_1, \dots, x_k удовлетворяют отношению $R_i(k)$. В каждой i -ой формуле F_i может быть различное число свободных (не связанных) переменных. Понятие "(формальная) теория" можно определить как кортеж $T = \langle S, F \rangle$, где S - сигнатура (множество определенных, разрешенных операции), а F - множество формул без свободных переменных (аксиом теории). Если дополнительно определено и множество правил вывода P , то $T = \langle S, F, P \rangle$. Отсюда видно, что формальная теория базируется на конкретной предметной области, определяемой сигатурой.

Для компьютерного представления и обработки *знаний* и данных о предметной области (об объектах, процессах, явлениях, их структуре и взаимосвязях), они должны быть формализованы и представлены в определенном формализованном виде.

При традиционном способе реализации математической модели, *знаний*, заложенных в ней, строится моделирующий алгоритм (моделирующая программа), т.е. *знания* процедурно зависят от метода (алгоритма) обработки. В интеллектуальных системах (в системах искусственного интеллекта, в частности) *знания* о предметной области представлены в виде декларативной (описательной) модели формирования базы *знаний* и соответствующих правил вывода из нее и явно не зависят от процедуры их обработки. Для этого используются специальные модели *представления знаний*, например, продукционные, *фреймовые*, сетевые и логические. При обработке модели *знаний* используются процедуры логического вывода, называемые также механизмом или машиной вывода. Обычно в базе *знаний* зафиксированы общие закономерности, правила, описывающие проблемную среду и предметную область.

Процедуры вывода позволяют на основании общих правил вывести решение для заданной конкретной ситуации, описываемой некоторыми исходными данными. Цепочка логического вывода строится по мере приближения к решению, в зависимости от выведенных на каждом шаге данных и выведенных к этому шагу новых *знаний*. Конкретные формы

организации дедуктивного вывода зависят от того, в какой форме представлены *знания* в базе *знаний* (на каком языке *представления знаний*).

Продукционная модель представления знаний наиболее распространена в приложениях.

Модель реализуется правилами-продукциями:

если <условие> то <заключение>.

В качестве условия может выступать любая совокупность суждений, соединенных логическими связками и (\wedge), или (\vee).

Пример. Продукцией будет следующее правило:

если (курс доллара-растет) \vee (сезон-осень) \wedge (число продавцов-убывает)

то (прогноз цен на рынке жилья - рост рублевых цен на квартиры).

Такого рода правила и *знания* о ценах, предложении и спросе на рынке жилья могут стать базой для базы *знаний* о рынке жилья и экспертной системы для риэлторской группы (фирмы).

Существуют две основные стратегии вывода на множестве правил-продукций:

1. прямой вывод (вывод от исходных данных-фактов, аксиом - к цели, по пути вывода пополняя исходную базу *знаний* новыми полученными истинными фактами; процесс заканчивается лишь тогда, когда выведен факт, эквивалентный искомому);
2. обратный вывод (вывод от целевого факта к данным, на очередном шаге отыскивается очередной факт, в заключительной части содержится факт, эквивалентный исходному факту; процесс заканчивается тогда, когда для каждого факта, выведенного на очередном шаге, не будет найдено правило, имеющее этот факт в качестве заключения, а посылками - исходные или выведенные на предыдущих шагах факты).

Обе приведенные стратегии вывода имеют недостатки, достоинства и модификации.

Пример. Если все множество правил-продукций разбито на группы по некоторому признаку (структурировано), то вместо полного или случайного перебора всех правил при прямом и обратном выводе осуществляется целенаправленный переход от одной группы правил к другой. Используются также смешанные стратегии вывода, сочетающие прямой и обратный вывод.

Продукционные модели удобны для представления логических взаимосвязей между фактами, так как они более формализованы и достаточно строгие (теоретические), модульные (продукции явно между собой не связаны, поэтому их можно модифицировать по модульной технологии), соответствуют долговременной памяти человека.

Представление знаний в виде *семантической сети* является одной из основных моделей *представления знаний*.

Семантическая сеть - это ориентированная графовая структура, каждая вершина которой отображает некоторое понятие (объект, процесс, ситуацию), а ребра графа соответствуют отношениям типа "это есть", "принадлежать", "быть причиной", "входить в", "состоять из", "быть как" и аналогичным между парами понятий. На *семантических сетях* используются специальные процедуры вывода: пополнение сети, наследование свойств, поиск по образцу и др.

Пример. Рассмотрим факт: "причиной неритмичной работы предприятия является старое оборудование, а причиной последнего - отсутствие оборотных средств". *Семантическая сеть* может содержать вершины "оборотные средства", "старое оборудование", соединяемые ребрами - отношениями типа "быть причиной".

Достоинство *семантических сетей* - наглядность *представления знаний*, с их помощью удобно представлять причинно-следственные связи между элементами (подсистемами), а также структуру сложных систем. Недостаток таких сетей - сложность вывода, поиска подграфа, соответствующего запросу.

Характерная особенность *семантических сетей* - наличие трех типов отношений:

1. класс - элемент класса (часть - целое, класс - подкласс, элемент - множество и т.п.);
2. свойство - значение (иметь свойство, иметь значение и т.п.);
3. пример элемента класса (элемент за, элемент под, раньше, позже и др.).

Фреймовая модель представления знаний задает остов описания класса объектов и удобна для описания структуры и характеристик одноптипных объектов (процессов, событий) описываемых **фреймами** - специальными ячейками (шаблонами понятий) фреймовой сети (знания).

Фрейм - концентратор знаний и может быть активизирован как отдельный автономный элемент и как элемент сети. **Фрейм** - это модель кванта знаний (абстрактного образа, ситуации), активизация **фрейма** аналогична активизации этого кванта знаний - для объяснения, предсказания и т.п. Отдельные характеристики (элементы описания) объекта называются **слотами фрейма**. **Фреймы** сети могут наследовать **слоты** других **фреймов** сети.

Различают фреймы-образцы (прототипы), хранящиеся в базе знаний, и фреймы-экземпляры, создаваемые для отображения реальных ситуаций для конкретных данных.

Фреймовое представление данных достаточно универсальное. Оно позволяет отображать знания с помощью:

- фрейм-структур - для обозначения объектов и понятий;
- фрейм-ролей - для обозначения ролевых обязанностей;
- фрейм-сценариев - для обозначения поведения;
- фрейм-ситуаций - для обозначения режимов деятельности, состояний.

Пример. Фрейм-структурами являются понятия "заем", "вексель", "кредит". Фрейм-роли - "кассир", "клиент", "сервер". Фрейм-сценарии - "страхование", "банкинг", "банкротство". Фрейм-ситуации - "эволюция", "функционирование", "безработица".

Пример. Например, возьмем такое понятие, как "функция". Различные функции могут отличаться друг от друга, но существует некоторый набор формальных характеристик для описания любой функции (**фрейм** "Функция"): тип и допустимое множество изменений аргумента (область определения функции), тип и допустимое множество значений функции (множество значений функции), аналитическое правило связи аргумента со значением функции. Соответственно, могут быть определены **фреймы** "Аргумент", "Значение функции", "Закон соответствия". Далее можно определить **фреймы** "Тип аргумента", "Вычисление значения функции", "Операция" и др. Пример **слотов** для **фрейма** "Закон соответствия": аналитический способ задания закона; сложность вычисления (реализации). Чтобы описать конкретное значение **фрейма**, необходимо каждому **слоту** придать конкретное значение, например, таким образом:

Имя **фрейма** - Функция;

Аргумент - x ;

Значение функции - y ;

Закон соответствия - квадратичный.

Слоты:

Значения аргумента - R ;

Способ задания функции - $y=ax^2+bx+c$;

Сложность вычисления - 7.

Пример. **Фрейм** "Задача вычислительного типа" - на рис. [14.1](#).



Рис. 14.1. Структура фрейма "Задача вычислительного типа"

Фреймовое представление наглядно и структурировано (модульно) и позволяет получать описание системы в виде связанных, иерархических структур (модулей - *фреймов*, единиц представления знаний).

Логическая (предикатная) **модель представления знаний** основана на алгебре высказываний и предикатов, на системе аксиом этой алгебры и ее правилах вывода. Из предикатных моделей наибольшее распространение получила модель предикатов первого порядка, базирующаяся на термах (аргументах предикатов - логических констант, переменных, функций), предикатах (выражениях с логическими операциями). Предметная область описывается при этом с помощью предикатов и системы аксиом.

Пример. Возьмем утверждение: "Инфляция в стране превышает прошлогодний уровень в 2 раза". Это можно записать в виде *логической модели*: $r(\text{InfNew}, \text{InfOld}, n)$, где $r(x,y)$ - отношение вида " $x=ny$ ", InfNew - текущая инфляция в стране, InfOld - инфляция в прошлом году. Тогда можно рассматривать истинные и ложные предикаты, например, $r(\text{InfNew}, \text{InfOld}, 2)=1$, $r(\text{InfNew}, \text{InfOld}, 3)=0$ и т.д. Очень полезные операции для логических выводов - операции импликации, эквиваленции и др.

Логические модели удобны для представления логических взаимосвязей между фактами, они формализованы, строги (теоретические), для их использования имеется удобный и адекватный инструментарий, например, язык логического программирования Пролог.

Модель предметной области можно определить упрощенно в виде:

$\langle \text{модель предметной области} \rangle = \langle \text{понятийные знания} \rangle + \langle \text{конструктивные знания} \rangle$.

При реализации указанных выше моделей используются **эвристики** - эмпирические или полуэмпирические правила, с помощью которых эксперт (экспертная группа) в отсутствие алгоритма (например, задача плохо структурируема) пытается найти решение, моделируя возможный ход рассуждений эксперта на основе эвристической информации, получаемых в результате опыта, наблюдения, сбора и анализа статистики.

Пример. Сбор эвристической информации у представителей рынка приводит к следующим **знаниям**, которые можно представить, например, *семантической сетью* или продукциями:

1. нужно рекламировать свой товар активно в начальный период;
2. нужно поднимать цены в условиях отсутствия конкуренции;

3. нужно опускать цены в условиях жесткой конкуренции;

4. нужно стараться быть монополистом на рынке и др.

Многие *знания*, особенно находящиеся на стыке наук, трудно формализовать и описать формальными моделями, исследовать аналитически. В таких случаях часто применяют *эвристики*, эвристические процедуры, использующие аналоги, опыт поиска нового, исследования родственных задач, перебор вариантов с учетом интуиции.

Пример. Такими процедурами учат компьютер играть в шахматы. Шахматная программа - один из самых ранних примеров невычислительного применения ЭВМ. Если в 50-х годах она "играла" на уровне "разрядника", то за 40-50 лет она "научилась играть" на уровне чемпиона мира.

Задачи и упражнения

1. Формализуйте понятия "Решить задачу", "Решение задачи", "Метод решения задачи", "Алгоритм решения задачи".
2. Постройте одну *продукционную* и одну семантическую модели *знаний* по специальности.
3. Постройте одну *фреймовую* и одну *логическую модели знаний* по специальности.

Экспертные системы

Об экспертных системах (ЭС) можно говорить много и сложно. Но наш разговор очень упростится, если мы будем исходить из следующего определения экспертной системы. **Экспертная система** — это программа (на современном уровне развития человечества), которая заменяет эксперта в той или иной области.

Отсюда вытекает простой вывод — все, что мы изучаем в курсе "Основы проектирования систем с ИИ", конечной целью ставит разработку ЭС. В этой лекции мы остановимся только на некоторых особенностях их построения, которые не затрагиваются в остальных лекциях.

ЭС предназначены, главным образом, для решения практических задач, возникающих в слабо структурированной и трудно формализуемой предметной области. ЭС были первыми системами, которые привлекли внимание потенциальных потребителей продукции искусственного интеллекта.

С ЭС связаны некоторые распространенные заблуждения.

Заблуждение первое: ЭС будут делать не более (а скорее даже менее) того, чем может эксперт, создавший данную систему. Для опровержения данного постулата можно построить самообучающуюся ЭС в области, в которой вообще нет экспертов, либо объединить в одной ЭС знания нескольких экспертов, и получить в результате систему, которая может то, чего ни один из ее создателей не может.

Заблуждение второе: ЭС никогда не заменит человека-эксперта. Уже заменяет, иначе зачем бы их создавали?

Экспертные системы, методика построения

В настоящее время сложилась определенная технология разработки ЭС, которая включает следующие шесть этапов: идентификация, концептуализация, формализация, выполнение, тестирование и опытная эксплуатация.

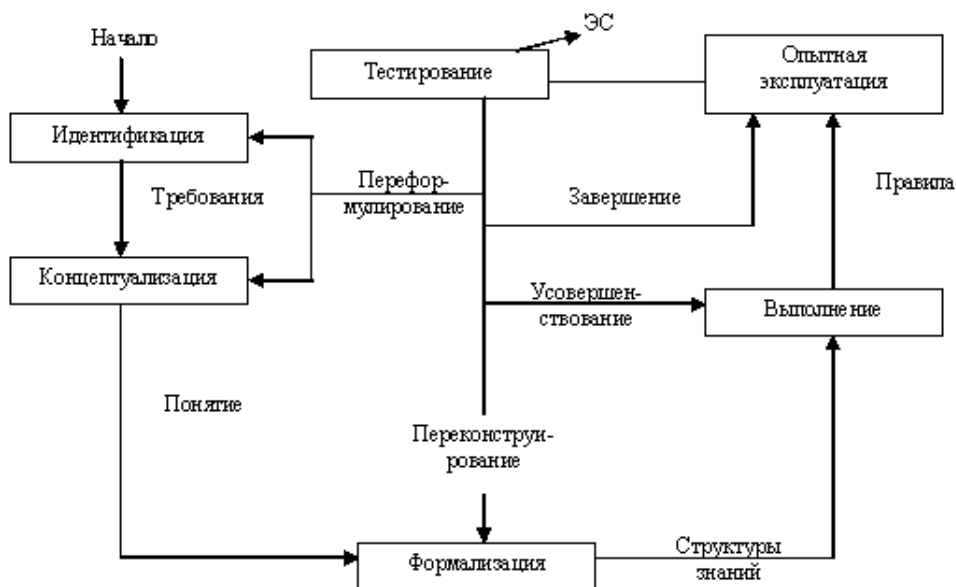


Рис. 9.1. Методика (этапы) разработки ЭС

Этап идентификации

Этап идентификации связан, прежде всего, с осмыслением тех задач, которые предстоит решить будущей ЭС, и формированием требований к ней. Результатом данного этапа является ответ на вопрос, что надо сделать и какие ресурсы необходимо задействовать (идентификация задачи, определение участников процесса проектирования и их роли, выявление ресурсов и целей).

Обычно в разработке ЭС участвуют не менее трех-четырех человек — один эксперт, один или два инженера по знаниям и один программист, привлекаемый для модификации и согласования инструментальных средств. Также к процессу разработки ЭС могут по мере необходимости привлекаться и другие участники. Например, инженер по знаниям может пригласить других экспертов, чтобы убедиться в правильности своего понимания основного эксперта, представительности тестов, демонстрирующих особенности рассматриваемой задачи, совпадения взглядов различных экспертов на качество предлагаемых решений. Кроме того, для сложных систем считается целесообразным привлекать к основному циклу разработки несколько экспертов. Однако в этом случае, как правило, требуется, чтобы один из экспертов отвечал за непротиворечивость знаний, сообщаемых коллективом экспертов.

Идентификация задачи заключается в составлении неформального (вербального) описания, в котором указываются: общие характеристики задачи; подзадачи, выделяемые внутри данной задачи; ключевые понятия (объекты), их входные (выходные) данные; предположительный вид решения, а также знания, относящиеся к решаемой задаче.

В процессе идентификации задачи инженер по знаниям и эксперт работают в тесном контакте. Начальное неформальное описание задачи экспертом используется инженером по знаниям для уточнения терминов и ключевых понятий. Эксперт корректирует описание задачи, объясняет, как решать ее и какие рассуждения лежат в основе того или иного решения. После нескольких циклов, уточняющих описание, эксперт и инженер по знаниям получают окончательное неформальное описание задачи.

При проектировании ЭС типичными ресурсами являются источники знаний, время разработки, вычислительные средства и объем финансирования. Для эксперта источниками знаний служат его предшествующий опыт по решению задачи, книги, известные примеры решения задач, а для инженера по знаниям — опыт в решении аналогичных задач, методы представления знаний и манипулирования ими, программные инструментальные средства. При определении времени разработки обычно имеется в виду, что сроки разработки и внедрения ЭС составляют, как правило, не менее года (при трудоемкости 5 чел.-лет). Определение объема финансирования оказывает существенное влияние на процесс

разработки, так как, например, при недостаточном финансировании предпочтение может быть отдано не разработке оригинальной новой системы, а адаптации существующей.

При идентификации целей важно 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.

Этап концептуализации

На данном этапе проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач. Этот этап завершается созданием модели предметной области (ПО), включающей основные концепты и отношения. На этапе **концептуализации** определяются следующие особенности задачи: типы доступных данных; исходные и выводимые данные, подзадачи общей задачи; применяемые стратегии и гипотезы; виды взаимосвязей между объектами ПО, типы используемых отношений (иерархия, причина — следствие, часть — целое и т.п.); процессы, применяемые в ходе решения; состав знаний, используемых при решении задачи; типы ограничений, накладываемых на процессы, которые применены в ходе решения; состав знаний, используемых для обоснования решений.

Существует два подхода к процессу построения модели предметной области, которая является целью разработчиков ЭС на этапе концептуализации. Признаковый или атрибутивный подход предполагает наличие полученной от экспертов информации в виде троек объект—атрибут—значение атрибута, а также наличие обучающей информации. Этот подход развивается в рамках направления, получившего название "формирование знаний" или "машинное обучение" (machine learning).

Второй подход, называемый структурным (или когнитивным), осуществляется путем выделения элементов предметной области, их взаимосвязей и семантических отношений.

Для атрибутивного подхода характерно наличие наиболее полной информации о предметной области: об объектах, их атрибутах и о значениях атрибутов. Кроме того, существенным моментом является использование дополнительной обучающей информации, которая задается группированием объектов в классы по тому или иному содержательному критерию. Тройки объект—атрибут—значение атрибута могут быть получены с помощью так называемого метода реклассификации, который основан на предположении что задача является объектно-ориентированной и объекты задачи хорошо известны эксперту. Идея метода состоит в том, что конструируются правила (комбинации значений атрибутов), позволяющие отличить один объект от другого. Обучающая информация может быть задана на основании прецедентов правильных экспертных заключений, например, с помощью метода извлечения знаний, получившего название "анализ протоколов мыслей вслух".

При наличии обучающей информации для формирования модели предметной области на этапе концептуализации можно использовать весь арсенал методов, развиваемых в рамках задачи распознавания образов. Таким образом, несмотря на то, что здесь атрибутивному подходу не уделено много места, он является одним из потребителей всего того, что было указано в лекции, посвященной распознаванию образов и автоматического группирования данных.

Структурный подход к построению модели предметной области предполагает выделение следующих когнитивных элементов знаний: 1. Понятия. 2. Взаимосвязи. 3. Метапонятия. 4. Семантические отношения.

Выделяемые понятия предметной области должны образовывать систему, под которой понимается совокупность понятий, обладающая следующими свойствами: уникальностью (отсутствием избыточности); полнотой (достаточно полным описанием различных процессов, фактов, явлений и т.д. предметной области); достоверностью (валидностью — соответствием выделенных единиц смысловой информации их реальным наименованиям) и непротиворечивостью (отсутствием омонимии).

При построении системы понятий с помощью "метода локального представления" эксперта просят разбить задачу на подзадачи для перечисления целевых состояний и описания общих категорий цели. Далее для каждого разбиения (локального представления) эксперт формулирует информационные факты и дает им четкое наименование (название). Считается, что для успешного решения задачи построения модели предметной области число таких информационных фактов в каждом локальном представлении, которыми человек способен одновременно манипулировать, должно быть примерно равно семи.

"Метод вычисления коэффициента использования" основан на следующей гипотезе. Элемент данных (или информационный факт) может являться понятием, если он:

- используется в большом числе подзадач;
- используется с большим числом других элементов данных;
- редко используется совместно с другими элементами данных по сравнению с общим числом его применения во всех подзадачах (это и есть коэффициент использования).

Полученные значения могут служить критерием для классификации всех элементов данных и, таким образом, для формирования системы понятий.

"Метод формирования перечня понятий" заключается в том, что экспертам (желательно, чтобы их было больше двух) дается задание составить список понятий, относящихся к исследуемой предметной области. Понятия, выделенные всеми экспертами, включаются в систему понятий, остальные подлежат обсуждению.

"Ролевой метод" состоит в том, что эксперту дается задание обучить инженера по знаниям решению некоторых задач предметной области. Таким образом, эксперт играет роль учителя, а инженер по знаниям — роль ученика. Процесс обучения записывается на магнитофон. Затем третий участник прослушивает магнитофонную ленту и выписывает на бумаге все понятия, употребленные учителем или учеником.

При использовании метода "составления списка элементарных действий" эксперту дается задание составить такой список при решении задачи в произвольном порядке.

В методе "составление оглавления учебника" эксперту предлагается представить ситуацию, в которой его попросили написать учебник. Необходимо составить на бумаге перечень предполагаемых глав, разделов, параграфов, пунктов и подпунктов книги.

"Текстологический метод" формирования системы понятий заключается в том, что эксперту дается задание выписать из руководств (книг по специальности) некоторые элементы, представляющие собой единицы смысловой информации.

Группа методов установления взаимосвязей предполагает установление семантической близости между отдельными понятиями. В основе установления взаимосвязей лежит психологический эффект "свободных ассоциаций", а также фундаментальная категория близости объектов или концептов.

Эффект свободных ассоциаций заключается в следующем. Испытуемого просят отвечать на заданное слово первым пришедшим на ум словом. Как правило, реакция большинства испытуемых (если слова не были слишком необычными) оказывается одинаковой. Количество переходов в цепочке может служить мерой "смыслового расстояния" между двумя понятиями. Многочисленные опыты подтверждают гипотезу, что для двух любых слов (понятий) существует ассоциативная цепочка, состоящая не более чем из семи слов.

"Метод свободных ассоциаций" основан на психологическом эффекте, описанном выше. Эксперту предъявляется понятие с просьбой назвать как можно быстрее первое пришедшее на ум понятие из сформированной ранее системы понятий. Далее производится анализ полученной информации.

В методе "сортировка карточек" исходным материалом служат выписанные на карточки понятия. Применяются два варианта метода. В первом эксперту задаются некоторые глобальные критерии предметной области, которыми он должен руководствоваться при раскладывании карточек на группы. Во втором случае, когда сформулировать глобальные

критерии невозможно, эксперту дается задание разложить карточки на группы в соответствии с интуитивным пониманием семантической близости предъявляемых понятий.

"Метод обнаружения регулярностей" основан на гипотезе о том, что элементы цепочки понятия, которые человек вспоминает с определенной регулярностью, имеют тесную ассоциативную взаимосвязь. Для эксперимента произвольным образом отбирается 20 понятий. Эксперту предъявляется одно из числа отобранных. Процедура повторяется до 20 раз, причем каждый раз начальные концепты должны быть разными. Затем инженер по знаниям анализирует полученные цепочки с целью нахождения постоянно повторяющихся понятий (регулярностей). Внутри выделенных таким образом группировок устанавливаются ассоциативные взаимосвязи.

Кроме рассмотренных выше неформальных методов для установления взаимосвязей между отдельными понятиями применяются также формальные методы. Сюда в первую очередь относятся методы семантического дифференциала и репертуарных решеток.

Выделенные понятия предметной области и установленные между ними взаимосвязи служат основанием для дальнейшего построения системы метапонятий — осмысленных в контексте изучаемой предметной области системы группировок понятий. Для определения этих группировок применяют как неформальные, так и формальные методы.

Интерпретация, как правило, легче дается эксперту, если группировки получены неформальными методами. В этом случае выделенные классы более понятны эксперту. Причем в некоторых предметных областях совсем не обязательно устанавливать взаимосвязи между понятиями, так как метапонятия, образно говоря, "лежат на поверхности".

Последним этапом построения модели предметной области при концептуальном анализе является установление семантических отношений между выделенными понятиями и метапонятиями. Установить семантические отношения — это значит определить специфику взаимосвязи, полученной в результате применения тех или иных методов. Для этого необходимо каждую зафиксированную взаимосвязь осмыслить и отнести ее к тому или иному типу отношений.

Существует около 200 базовых отношений, например, "часть — целое", "род — вид", "причина — следствие", пространственные, временные и другие отношения. Для каждой предметной области помимо общих базовых отношений могут существовать и уникальные отношения.

"Прямой метод" установления семантических отношений основан на непосредственном осмыслении каждой взаимосвязи. В том случае, когда эксперт затрудняется дать интерпретацию выделенной взаимосвязи, ему предлагается следующая процедура. Формируются тройки: понятие 1 — связь — понятие 2. Рядом с каждой тройкой записывается короткое предложение или фраза, построенное так, чтобы понятие 1 и понятие 2 входили в это предложение. В качестве связок используются только содержательные отношения и не применяются неопределенные связки типа "похож на" или "связан с".

Для "косвенного метода" не обязательно иметь взаимосвязи, достаточно лишь наличие системы понятий. Формулируется некоторый критерий, для которого из системы понятий выбирается определенная совокупность концептов. Эта совокупность предъявляется эксперту с просьбой дать вербальное описание сформулированного критерия. Концепты предъявляются эксперту все сразу (желательно на карточках). В случае затруднений эксперта прибегают к разбиению отобранных концептов на группы с помощью более мелких критериев. Исходное количество концептов может быть произвольным, но после разбиения на группы в каждой из таких групп должно быть не более десяти концептов. После того как составлены описания по всем группам, эксперту предлагают объединить эти описания в одно.

Следующий шаг в косвенном методе установления семантических отношений — это анализ текста, составленного экспертом. Концепты заменяют цифрами (это может быть исходная нумерация), а связки оставляют. Тем самым строится некоторый граф, вершинами которого служат концепты, а дугами — связки (например, "ввиду", "приводит к", "выражаясь с одной стороны", "обуславливая", "сочетаясь", "определяет", "вплоть до" и т.д.) Этот метод

позволяет устанавливать не только базовые отношения, но и отношения, специфические для конкретной предметной области.

Рассмотренные выше методы формирования системы понятий и метапонятий, установления взаимосвязей и семантических отношений в разных сочетаниях применяются на этапе концептуализации при построении модели предметной области.

Этап формализации

Теперь все ключевые понятия и отношения выражаются на некотором формальном языке, который либо выбирается из числа уже существующих, либо создается заново. Другими словами, на *данном этапе* определяются состав средств и способы представления декларативных и процедурных знаний, осуществляется это представление и в итоге формируется описание решения задачи ЭС на предложенном (инженером по знаниям) формальном языке.

Выходом этапа формализации является описание того, как рассматриваемая задача может быть представлена в выбранном или разработанном формализме. Сюда относится указание способов представления знаний (фреймы, сценарии, семантические сети и т.д.) и определение способов манипулирования этими знаниями (логический вывод, аналитическая модель, статистическая модель и др.) и интерпретации знаний.

Этап выполнения

Цель *этого этапа* — создание одного или нескольких прототипов ЭС, решающих требуемые задачи. Затем на данном этапе по результатам тестирования и опытной эксплуатации создается конечный продукт, пригодный для промышленного использования. Разработка прототипа состоит в программировании его компонентов или выборе их из известных инструментальных средств и наполнении базы знаний.

Главное в создании прототипа заключается в том, чтобы этот прототип обеспечил проверку адекватности идей, методов и способов представления знаний решаемым задачам. Создание первого прототипа должно подтвердить, что выбранные методы решений и способы представления пригодны для успешного решения, по крайней мере, ряда задач из актуальной предметной области, а также продемонстрировать тенденцию к получению высококачественных и эффективных решений для всех задач предметной области по мере увеличения объема знаний.

После разработки первого прототипа ЭС-1 круг предлагаемых для решения задач расширяется, и собираются пожелания и замечания, которые должны быть учтены в очередной версии системы ЭС-2. Осуществляется развитие ЭС-1 путем добавления "дружественного" интерфейса, средств для исследования базы знаний и цепочек выводов, генерируемых системой, а также средств для сбора замечаний пользователей и средств хранения библиотеки задач, решенных системой.

Выполнение экспериментов с расширенной версией ЭС-1, анализ пожеланий и замечаний служат отправной точкой для создания второго прототипа ЭС-2. Процесс разработки ЭС-2 — итеративный. Он может продолжаться от нескольких месяцев до нескольких лет в зависимости от сложности предметной области, гибкости выбранного представления знаний и степени соответствия управляющего механизма решаемым задачам (возможно, потребуется разработка ЭС-3 и т.д.). При разработке ЭС-2, кроме перечисленных задач, решаются следующие:

- анализ функционирования системы при значительном расширении базы знаний;
- исследование возможностей системы в решении более широкого круга задач и принятие мер для обеспечения таких возможностей;
- анализ мнений пользователей о функционировании ЭС;
- разработка системы ввода-вывода, осуществляющей анализ или синтез предложений ограниченного естественного языка, позволяющей взаимодействовать с ЭС-2 в форме, близкой к форме стандартных учебников для данной области.

Если ЭС-2 успешно прошла этап тестирования, то она может классифицироваться как промышленная экспертная система.

Этап тестирования

В ходе данного этапа производится оценка выбранного способа представления знаний в ЭС в целом. Для этого инженер по знаниям подбирает примеры, обеспечивающие проверку всех возможностей разработанной ЭС.

Различают следующие источники неудач в работе системы: тестовые примеры, ввод-вывод, правила вывода, управляющие стратегии.

Показательные тестовые примеры являются наиболее очевидной причиной неудачной работы ЭС. В худшем случае тестовые примеры могут оказаться вообще вне предметной области, на которую рассчитана ЭС, однако чаще множество тестовых примеров оказывается слишком однородным и не охватывает всю предметную область. Поэтому при подготовке тестовых примеров следует классифицировать их по подпроблемам предметной области, выделяя стандартные случаи, определяя границы трудных ситуаций и т.п.

Ввод-вывод характеризуется данными, приобретенными в ходе диалога с экспертом, и заключениями, предъявленными ЭС в ходе объяснений. Методы приобретения данных могут не давать требуемых результатов, так как, например, задавались неправильные вопросы или собрана не вся необходимая информация. Кроме того, вопросы системы могут быть трудными для понимания, многозначными и не соответствующими знаниям пользователя. Ошибки при вводе могут возникать также из-за неудобного для пользователя входного языка. В ряде приложения для пользователя удобен ввод не только в печатной, но и в графической или звуковой форме.

Выходные сообщения (заключения) системы могут оказаться непонятны пользователю (эксперту) по разным причинам. Например, их может быть слишком много или, наоборот, слишком мало. Также причиной ошибок может являться неудачная организация, упорядоченность заключений или неподходящий пользователю уровень абстракций с непонятной ему лексикой.

Наиболее распространенный источник ошибок в рассуждениях находится в правилах вывода. Важная причина здесь часто кроется в отсутствии учета взаимозависимости сформированных правил. Другая причина заключается в ошибочности, противоречивости и неполноте используемых правил. Если неверна посылка правила, то это может привести к употреблению правила в неподходящем контексте. Если ошибочно действие правила, то трудно предсказать конечный результат. Правило может быть ошибочно, если при корректности его условия и действия нарушено соответствие между ними.

Нередко к ошибкам в работе ЭС приводят применяемые управляющие стратегии. Изменение стратегии бывает необходимо, например, если ЭС анализирует сущности в порядке, отличном от "естественного" для эксперта. Последовательность, в которой данные рассматриваются ЭС, не только влияет на эффективность работы системы, но и может приводить к изменению конечного результата. Так, рассмотрение правила А до правила В способно привести к тому, что правило В всегда будет игнорироваться системой. Изменение стратегии бывает также необходимо и в случае неэффективной работы ЭС. Кроме того, недостатки в управляющих стратегиях могут привести к чрезмерно сложным заключениям и объяснениям ЭС.

Критерии оценки ЭС зависят от точки зрения. Например, при тестировании ЭС-1 главным в оценке работы системы является полнота и безошибочность правил вывода. При тестировании промышленной системы превалирует точка зрения инженера по знаниям, которого в первую очередь интересует вопрос оптимизации представления и манипулирования знаниями. И, наконец, при тестировании ЭС после опытной эксплуатации оценка производится с точки зрения пользователя, заинтересованного в удобстве работы и получении практической пользы.

Этап опытной эксплуатации

На этом этапе проверяется пригодность ЭС для конечного пользователя. Пригодность ЭС для пользователя определяется в основном удобством работы с ней и ее полезностью. Под полезностью ЭС понимается ее способность в ходе диалога определять потребности

пользователя, выявлять и устранять причины неудач в работе, а также удовлетворять указанные потребности пользователя (решать поставленные задачи). В свою очередь, удобство работы с ЭС подразумевает естественность взаимодействия с ней (общение в привычном, не утомляющем пользователя виде), гибкость ЭС (способность системы настраиваться на различных пользователей, а также учитывать изменения в квалификации одного и того же пользователя) и устойчивость системы к ошибкам (способность не выходить из строя при ошибочных действиях неопытного пользователя).

В ходе разработки ЭС почти всегда осуществляется ее модификация. Выделяют следующие виды модификации системы: переформулирование понятий и требований, переконструирование представления знаний в системе и усовершенствование прототипа.

Экспертные системы, параллельные и последовательные решения

Как мы можем заметить, в большинстве алгоритмов распознавания образов подразумевается, что к началу работы алгоритма уже известна вся входная информация, которая перерабатывается параллельно. Однако ее получение зачастую требует определенных усилий. Да и наши наблюдения за реальными экспертами подтверждают, что зачастую они задают два-три вопроса, после чего делают правильные выводы. Представьте себе, если бы врач (эксперт в области медицины) перед постановкой диагноза "ангина" заставлял бы пациента пройти полное обследование вплоть до кулоноскопии и пункции позвоночника (я не пробовал ни то, ни другое, но думаю, что это малоприятные вещи, а также значительная потеря времени).

Соответственно большинство алгоритмов модифицируются, чтобы обеспечить выполнение следующих условий:

- алгоритмы должны работать в условиях неполной информации (последовательно);
- последовательность запроса информации должна быть оптимальна по критериям скорости получения результата и (или) наименьшей трудоемкости (болезненности, стоимости и т.д.) получения этой информации.

Одной из возможных стратегий для оптимизирования запросов является стратегия получения в первую очередь той информации, которая подтверждает либо опровергает наиболее вероятный на текущий момент результат. Другими словами мы пытаемся подтвердить или опровергнуть наши догадки (обратный вывод).

Пример ЭС, основанной на правилах логического вывода и действующую в обратном порядке

Допустим, вы хотите построить ЭС в области медицинской диагностики. В этом случае вам вряд ли нужно строить систему, использующую обучение на примерах, потому что имеется большое количество доступной информации, позволяющей непосредственно решать такие проблемы. К сожалению, эта информация приведена в неподходящем для обработки на компьютере виде.

Возьмите медицинскую энциклопедию и найдите в ней статью, например, о гриппе. Вы обнаружите, что в ней приведены все симптомы, причем они бесспорны. Другими словами, при наличии указанных симптомов всегда можно поставить точный диагноз.

Но чтобы использовать информацию, представленную в таком виде, вы должны обследовать пациента, решить, что у него грипп, а потом заглянуть в энциклопедию, чтобы убедиться, что у него соответствующие симптомы. Что-то здесь не так. Ведь необходимо, чтобы вы могли обследовать пациента, решить, какие у него симптомы, а потом по этим симптомам определить, чем он болен. Энциклопедия же, похоже, не позволяет сделать это так, как надо. Нам нужна не болезнь со множеством симптомов, а система, представляющая группу симптомов с последующим названием болезни. Именно это мы сейчас и попробуем сделать.

Идеальной будет такая ситуация, при которой мы сможем в той или иной области предоставить машине в приемлемом для нее виде множество определений, которые она

сможет использовать примерно так же, как человек-эксперт. Именно это и пытаются делать такие программы, как PUFF, DENDRAL, PROSPECTOR.

С учетом байесовской системы логического вывода примем, что большая часть информации не является абсолютно точной, а носит вероятностный характер. Итак, начнем программирование:

№	Симптомы
1	Симптом_1
2	Симптом_2
N	Симптом_N

Полученный формат данных мы будем использовать для хранения симптомов. При слове "симптомы" создается впечатление, что мы связаны исключительно с медициной, хотя речь может идти о чем угодно. Суть в том, что компьютер задает множество вопросов, содержащихся в виде символьных строк <Симптом_1>, <Симптом_2> и т.д.

Например, Симптом_1 может означать строку "Много ли вы кашляете?", или, если вы пытаетесь отремонтировать неисправный автомобиль, — строку "Ослаб ли свет фар?".

Теперь оформим болезни:

№	Болезнь	p	[j, py, pn]
1	Болезнь_1	p1	[j, py, pn]1
2	Болезнь_2	p2	[j, py, pn]2
N	Болезнь_N	pn	[j, py, pn]n

В таком виде мы будем хранить информацию о болезнях. Это не обязательно должны быть болезни — могут быть любые результаты, и каждый оператор содержит один возможный исход и всю информацию, относящуюся к нему.

Поле "болезнь" характеризует название возможного исхода, например "Грипп". Следующее поле — p — это априорная вероятность такого исхода P(H), т.е. вероятность исхода в случае отсутствия дополнительной информации. После этого идет ряд повторяющихся полей из трех элементов. Первый элемент — j — это номер соответствующего симптома (свидетельства, переменной, вопроса, если вы хотите назвать его по-другому). Следующие два элемента — P(E : H) и P(E : не H) — соответственно вероятности получения ответа "Да" на этот вопрос, если возможный исход верен и неверен. Например:

2010	Грипп	0.01	(1, 0.9, 0.01); (2, 1, 0.01); (3, 0, 0.01)
------	-------	------	--

Здесь сказано существует априорная вероятность P(H)=0.01, что любой наугад взятый человек болеет гриппом.

Допустим, программа задает вопрос 1 (симптом 1). Тогда мы имеем P(E : H)=0.9 и P(E : не H)=0.01, а это означает, что если у пациента грипп, то он в девяти случаях из десяти ответит "да" на этот вопрос, а если у него нет гриппа, он ответит "да" лишь в одном случае из ста. Очевидно, ответ "да" подтверждает гипотезу о том, что у него грипп. Ответ "нет" позволяет предположить, что человек гриппом не болеет.

Так же и во второй группе симптомов (2, 1, 0.01). В этом случае P(E : H)=0.9, т.е. если у человека грипп, то этот симптом должен присутствовать. Соответствующий симптом может существовать и при отсутствии гриппа (P(E : не H)=0.01), но это маловероятно.

Вопрос 3 исключает грипп при ответе "да", потому что P(E : H)=0. Это может быть вопрос вроде такого: "наблюдаете ли вы такой симптом на протяжении большей части жизни?" — или что-нибудь вроде этого.

Нужно подумать, — а если вы хотите получить хорошие результаты, то и провести исследование, — чтобы установить обоснованные значения для этих вероятностей. И если быть честным, то получение такой информации — вероятно, труднейшая задача, в решении которой компьютер также сможет существенно помочь Вам. Если вы напишите программу общего назначения, ее основой будет теорема Байеса, утверждающая:

$$P(H : E) = P(E : H) * P(H) / (P(E : H) * P(H) + P(E : не H) * P(не H)).$$

Вероятность осуществления некой гипотезы H при наличии определенных подтверждающих свидетельств E вычисляется на основе априорной вероятности этой гипотезы без подтверждающих свидетельств и вероятности осуществления свидетельств при условиях, что гипотеза верна или неверна.

Поэтому, возвращаясь к нашим болезням, увидим:

$$P(H : E) = p_y * p / (p_y * p + p_n * (1 - p)) .$$

В данном случае мы начинаем с того, что $P(H) = p$ для всех болезней. Программа задает соответствующий вопрос и в зависимости от ответа вычисляет $P(H : E)$. Ответ "да" подтверждает вышеуказанные расчеты, ответ "нет" тоже, но с $(1 - p_y)$ вместо p_y и $(1 - p_n)$ вместо p_n . Сделав так, мы забываем об этом, за исключением того, что априорная вероятность $P(H)$ заменяется на $P(H : E)$. Затем продолжается выполнение программы, но с учетом постоянной коррекции значения $P(H)$ по мере поступления новой информации.

Описывая алгоритм, мы можем разделить программу на несколько частей.

Часть 1.

Ввод данных.

Часть 2.

Просмотр данных на предмет нахождения априорной вероятности $P(H)$. Программа вырабатывает некоторые значения массива правил и размещает их в массиве RULEVALUE. Это делается для того, чтобы определить, какие вопросы (симптомы) являются самыми важными, и выяснить, о чем спрашивать в первую очередь. Если вы вычислите для каждого вопроса $RULEVALUE[I] = RULEVALUE[I] + ABS(P(H : E) - P(H : \text{не } E))$, то получите значения возможных изменений вероятностей всех болезней, к которым они относятся.

Часть 3.

Программа находит самый важный вопрос и задает его. Существует ряд вариантов, что делать с ответом: вы можете просто сказать: "да" или "нет". Можете попробовать сказать "не знаю", — изменений при этом не произойдет. Гораздо сложнее использовать шкалу от -5 до $+5$, чтобы выразить степень уверенности в ответе.

Часть 4.

Априорные вероятности заменяются новыми значениями при получении новых подтверждающих свидетельств.

Часть 5.

Подсчитываются новые значения правил. Определяются также минимальное и максимальное значения для каждой болезни, основанные на существующих в данный момент априорных вероятностях и предположениях, что оставшиеся свидетельства будут говорить в пользу гипотезы или противоречить ей. Важно выяснить: стоит ли данную гипотезу продолжать рассматривать или нет? Гипотезы, которые не имеют смысла, просто отбрасываются. Те же из них, чьи минимальные значения выше определенного уровня, могут считаться возможными исходами. После этого возвращаемся к части 3.

Методология построения экспертных систем

Экспертные системы: Определения и классификация

Одним из наиболее значительных достижений искусственного интеллекта стала разработка мощных компьютерных систем, получивших название «экспертных» или основанных на «знаниях» систем. В современном обществе при решении задач управления сложными многопараметрическими и сильносвязанными системами, объектами, производственными и технологическими процессами приходится сталкиваться с решением неформализуемых либо трудноформализуемых задач. Такие задачи часто возникают в следующих областях: авиация, космос и оборона, нефтеперерабатывающая промышленность и транспортировка нефтепродуктов, химия, энергетика, металлургия, целлюлозно-бумажная промышленность, телекоммуникации и связь, пищевая промышленность, машиностроение, производство цемента, бетона и т.п. транспорт, медицина и фармацевтическое производство,

административное управление, прогнозирование и *мониторинг*. Наиболее значительными достижениями в этой области стало создание систем, которые ставят диагноз заболевания, предсказывают месторождения полезных ископаемых, помогают в проектировании электронных устройств, машин и механизмов, решают задачи управления реакторами и другие задачи.

Под *экспертной системой (ЭС)* будем понимать программу, которая использует знания специалистов (*экспертов*) о некоторой конкретной узко специализированной предметной области и в пределах этой области способна принимать решения на уровне *эксперта-профессионала*.

Осознание полезности систем, которые могут копировать дорогостоящие или редко встречающиеся человеческие знания, привело к широкому внедрению и расцвету этой технологии в 80-е, 90-е годы прошлого века. Основу успеха ЭС составили два важных свойства, отмечаемые рядом исследователей:

- в ЭС знания отделены от данных, и мощность *экспертной системы* обусловлена в первую очередь мощностью базы знаний и только во вторую очередь используемыми методами решения задач;
- решаемые ЭС задачи являются неформализованными или слабоформализованными и используют эвристические, экспериментальные, субъективные знания *экспертов* в определенной предметной области.

Основными категориями решаемых ЭС задач являются: диагностика, управление (в том числе технологическими процессами), интерпретация, прогнозирование, проектирование, отладка и ремонт, планирование, наблюдение (*мониторинг*), обучение.

Обобщенная схема ЭС приведена на [рис. 6.1](#). Основу ЭС составляет подсистема логического вывода, которая использует информацию из базы знаний (БЗ), генерирует рекомендации по решению искомой задачи. Чаще всего для представления знаний в ЭС используются системы продукций и семантические сети. Допустим, БЗ состоит из фактов и правил (если <посылка> то <заключение>). Если ЭС определяет, что посылка верна, то правило признается подходящим для данной консультации и оно запускается в действие. Запуск правила означает принятие заключения данного правила в качестве составной части процесса консультации.

Обязательными частями любой ЭС являются также модуль приобретения знаний и модуль отображения и объяснения решений. В большинстве случаев, реальные ЭС в промышленной эксплуатации работают также на основе баз данных (БД). Только одновременная работа со знаниями и большими объемами информации из БД позволяет ЭС получить неординарные результаты, например, поставить сложный диагноз (медицинский или технический), открыть месторождение полезных ископаемых, управлять ядерным реактором в реальном времени.

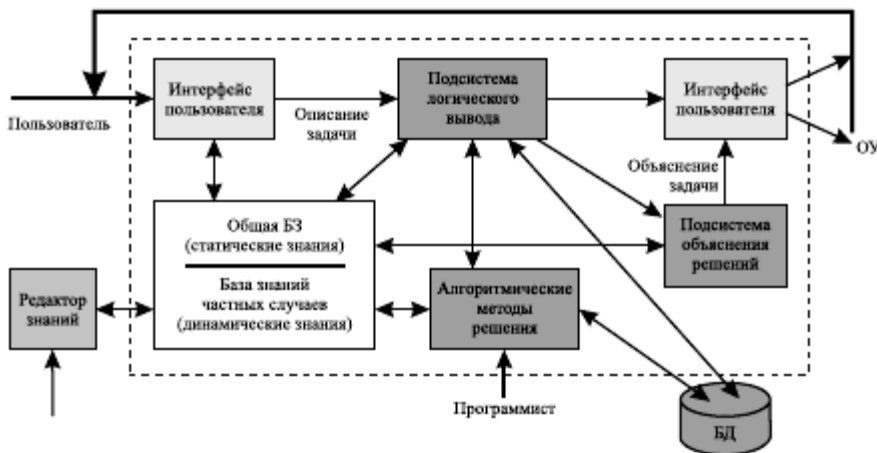


Рис. 6.1. Структура экспертной системы

Важную роль при создании ЭС играют инструментальные средства. Среди инструментальных средств для создания ЭС наиболее популярны такие языки

программирования, как LISP и PROLOG, а также *экспертные системы-оболочки* (ЭСО): KEE, CENTAUR, G2 и GDA, CLIPS, AT_ТЕХНОЛОГИЯ, предоставляющие в распоряжение разработчика - инженера по знаниям широкий набор для комбинирования систем представления знаний, языков программирования, объектов и процедур [81], [82].

Рассмотрим различные способы *классификации ЭС*.

По назначению ЭС делятся на:

- ЭС общего назначения.
- Специализированные ЭС:
 1. проблемно-ориентированные для задач диагностики, проектирования, прогнозирования
 2. предметно-ориентированные для специфических задач, например, контроля ситуаций на атомных электростанциях.

По степени зависимости от внешней среды выделяют:

- Статические ЭС, не зависящие от внешней среды.
- Динамические, учитывающие динамику внешней среды и предназначенные для решения задач в реальном времени. Время реакции в таких системах может задаваться в миллисекундах, и эти системы реализуются, как правило, на языке C++.

По типу использования различают:

- Изолированные ЭС.
- ЭС на входе/выходе других систем.
- Гибридные ЭС или, иначе говоря, ЭС интегрированные с базами данных и другими программными продуктами (приложениями).

По сложности решаемых задач различают:

- Простые ЭС - до 1000 простых правил.
- Средние ЭС - от 1000 до 10000 структурированных правил.
- Сложные ЭС - более 10000 структурированных правил.

По стадии создания выделяют:

- Исследовательский образец ЭС, разработанный за 1-2 месяца с минимальной БЗ.
- Демонстрационный образец ЭС, разработанный за 2-4 месяца, например, на языке типа LISP, PROLOG, CLIPS
- Промышленный образец ЭС, разработанный за 4-8 месяцев, например, на языке типа CLIPS с полной БЗ.
- Коммерческий образец ЭС, разработанный за 1,5-2 года, например, на языке типа C++, Java с полной БЗ.

Трудности при разработке экспертных систем

Разработка ЭС связана с определенными трудностями, которые необходимо хорошо знать, так же как и способы их преодоления. Рассмотрим подробнее эти проблемы.

1. Проблема *извлечения знаний экспертов*. Ни один специалист никогда просто так не раскроет секреты своего профессионального мастерства, свои сокровенные знания в профессиональной области. Он должен быть заинтересован материально или морально, причем хорошо заинтересован. Никто не хочет рубить сук, на котором сидит. Часто такой специалист опасается, что, раскрыв все свои секреты, он будет не нужен компании. Вместо него будет работать *экспертная система*. Избежать эту проблему поможет выбор высококвалифицированного *эксперта*, заинтересованного в сотрудничестве.
2. Проблема *формализации знаний экспертов*. *Эксперты*-специалисты в определенной области, как правило, не в состоянии формализовать свои знания. Часто они принимают правильные решения на интуитивном уровне и не могут аргументированно объяснить, почему принято то или иное решение. Иногда *эксперты* не могут прийти к взаимопониманию (фраза «встретились два геолога, у них было три мнения» - не шутка, а реальная жизнь). В таких ситуациях поможет выбор *эксперта*, умеющего ясно формулировать свои мысли и легко объяснять другим свои идеи.

3. Проблема нехватки времени у *эксперта*. Выбранный для разработки *эксперт* не может найти достаточно времени для выполнения проекта. Он слишком занят. Он всем нужен. У него есть проблемы. Чтобы избежать этой ситуации, необходимо получить от *эксперта*, прежде чем начнется проект, согласие тратить на проект время в определенном фиксированном объеме.
4. Правила, формализованные *экспертом*, не дают необходимой точности. Проблему можно избежать, если решать вместе с *экспертом* реальные задачи. Не надо придумывать «игрушечных» ситуаций или задач. В условиях задач нужно использовать реальные данные, такие как лабораторные данные, отчеты, дневники и другую информацию, взятую из практических задач. Постарайтесь говорить с *экспертом* на одном языке, используя единую терминологию. *Эксперт*, как правило, легче понимает правила, записанные на языке, близком к естественному, а не на языке типа LISP или PROLOG.
5. Недостаток ресурсов. В качестве ресурсов выступают персонал (инженеры знаний, разработчики инструментальных средств, *эксперты*) и средства построения ЭС (средства разработки и средства поддержки). Недостаток благожелательных и грамотных администраторов порождает скептицизм и нетерпение у руководителей. Повышенное внимание в прессе и преувеличения вызвали нереалистические ожидания, которые приводят к разочарованию в отношении *экспертных систем*. ЭС могут давать не самые лучшие решения на границе их применимости, при работе с противоречивыми знаниями и в рассуждениях на основе здравого смысла. Могут потребоваться значительные усилия, чтобы добиться небольшого увеличения качества работы ЭС. *Экспертные системы* требуют много времени на разработку. Так, создание системы PUFF для интерпретации функциональных тестов легких потребовало 5 человеко-лет, на разработку системы PROCPECTOR для разведки рудных месторождений ушло 30 человеко-лет, система XCON для расчета конфигурации компьютерных систем на основе VAX 11/780 потребовала 8 человеко-лет. ЭС последних лет разрабатываются более быстрыми темпами за счет развития технологий ЭС, но проблемы остались. Удвоение персонала не сокращает время разработки наполовину, потому что процесс создания ЭС - это процесс со множеством обратных связей. Все это необходимо учитывать при планировании создания ЭС.
6. Неадекватность инструментальных средств решаемой задаче. Часто определенные типы знаний (например, временные или пространственные) не могут быть легко представлены на одном ЯПЗ, так же как и разные схемы представления (например, фреймы и продукции) не могут быть достаточно эффективно реализованы на одном ЯПЗ. Некоторые задачи могут быть непригодными для решения по технологии ЭС (например, отдельные задачи анализа сцен). Необходим тщательный анализ решаемых задач, чтобы определить пригодность предлагаемых инструментальных средств и сделать правильный выбор.

Методология построения экспертных систем

Рассмотрим методику *формализации экспертных знаний* на примере создания *экспертных диагностических систем (ЭДС)*.

Целью создания ЭДС является определение состояния объекта диагностирования (ОД) и имеющихся в нем неисправностей.

Состояниями ОД могут быть: исправно, неисправно, работоспособно. Неисправностями, например, радиоэлектронных ОД являются обрыв связи, замыкание проводников, неправильное функционирование элементов и т.д.

Число неисправностей может быть достаточно велико (несколько тысяч). В ОД может быть одновременно несколько неисправностей. В этом случае говорят, что неисправности кратные.

Введем следующие определения. Разные неисправности ОД проявляются во внешней среде информационными параметрами. Совокупность значений информационных параметров

определяет «информационный образ» (ИО) неисправности ОД. ИО может быть полным, то есть содержать всю необходимую информацию для постановки диагноза, или, соответственно, неполным. В случае неполного ИО постановка диагноза носит вероятностный характер.

Основой для построения эффективных ЭДС являются знания *эксперта* для постановки диагноза, записанные в виде информационных образов, и система представления знаний, встраиваемая в *информационные системы* обеспечения функционирования и контроля ОД, интегрируемые с соответствующей технической аппаратурой.

Для описания своих знаний *эксперт* с помощью инженера по знаниям должен выполнить следующее.

1. Выделить множество всех неисправностей ОД, которые должна различать ЭДС.
2. Выделить множество *информативных (существенных) параметров*, значения которых позволяют различить каждую неисправность ОД и поставить диагноз с некоторой вероятностью.
3. Для выбранных параметров следует выделить информативные значения или информативные диапазоны значений, которые могут быть как количественными, так и качественными. Например, точные количественные значения могут быть записаны: задержка 25 нсек, задержка 30 нсек и т.д. Количественный диапазон значений может быть записан: задержка 25--40 нсек, 40--50 нсек, 50 нсек и выше. Качественный диапазон значений может быть записан: индикаторная лампа светится ярко, светится слабо, не светится.

Для более удобного дальнейшего использования качественный диапазон значений может быть закодирован, например, следующим образом:

- светится ярко $P1 = +++$ (или $P1 = 3$),
- светится слабо $P1 = ++$ (или $P1 = 2$),
- не светится $P1 = +$ (или $P1 = 1$).

Процедура получения информации по каждому из параметров определяется индивидуально в каждой конкретной системе диагностирования. Эта процедура может заключаться в автоматическом измерении параметров в ЭДС, в ручном измерении параметра с помощью приборов, качественном определении параметра, например, светится слабо, и т.д.

4. Процедура создания полных или неполных ИО каждой неисправности в алфавите значений информационных параметров может быть определена следующим образом. Составляются диагностические правила, определяющие вероятный диагноз на основе различных сочетаний диапазонов значений выбранных параметров ОД. Правила могут быть записаны в различной форме. Ниже приведена форма записи правил в виде таблицы.

Таблица 6.1. Диагностические правила

Номер	P1	P2	P3	Диагноз	Вероятность диагноза	Примечания
1		+++		Неисправен блок А1	0.95	
2	12-15	+		Неисправен блок А2	0.80	

Для записи правил с учетом изменений по времени следует ввести еще один параметр P0 - время (еще один столбец в таблице). В этом случае диагноз может ставиться на основе нескольких строк таблицы, а в графе Примечания могут быть указаны использованные тесты. Диагностическая таблица в этом случае представлена в [таблице 6.1](#).

Таблица 6.2. Динамические диагностические правила

Номер	P0	P1	P2	P3	Диагноз	Вероятность диагноза	Примечания
1	12:00	+	+	+			тест Т1
2	12:15	++	++	+	Неисправен блок А3	0.90	

Для записи последовательности проведения тестовых процедур и задания ограничений (если они есть) на их проведение может быть предложен аналогичный механизм. Механизм записи

последовательности проведения тестовых процедур в виде правил реализуется, например, следующим образом:

ЕСЛИ: P2 = 1

ТО: тест = T1, T3, T7

где T1, T3, T7 - тестовые процедуры, подаваемые на ОД при активизации (срабатывании) соответствующей продукции.

В современных ЭДС применяются различные стратегии поиска решения и постановки диагноза, которые позволяют определить необходимые последовательности тестовых процедур. Однако приоритет в ЭС отдается прежде всего знаниям и опыту, а лишь затем логическому выводу.

Примеры экспертных систем



Рис. 6.2. Общая структура экспертной системы диагностирования

Среди современных коммерческих систем хочется выделить *экспертную систему* - оболочку G2 американской фирмы Gensym (USA) как непревзойденную экспертную коммерческую систему для работы с динамическими объектами. Работа в реальном времени с малыми временами ответа часто необходима при анализе ситуаций в корпоративных информационных сетях, на атомных реакторах, в космических полетах и множестве других задач. В этих задачах необходимо принимать решения в течение миллисекунд с момента возникновения критической ситуации. ЭС G2, предназначенная для решения таких задач, отличается от большинства динамических ЭС такими характерными свойствами, как:

- работа в реальном времени с распараллеливанием процессов рассуждений;
- структурированный естественно-языковый интерфейс с управлением по меню и автоматической проверкой синтаксиса;
- обратный и прямой вывод, использование метазнаний, сканирование и фокусирование;
- интеграция подсистемы моделирования с динамическими моделями для различных классов объектов;
- структурирование БЗ, наследование свойств, понимание связей между объектами;
- библиотеки знаний являются ASCII-файлами и легко переносятся на любые платформы и типы ЭВМ;
- развитый редактор для сопровождения БЗ без программирования, средства трассировки и отладки БЗ;
- управление доступом с помощью механизмов авторизации пользователя и обеспечения желаемого взгляда на приложение;
- гибкий интерфейс оператора, включающий графики, диаграммы, кнопки, пиктограммы и т.п.;

- интеграция с другими приложениями (по TCP/IP) и базами данных, возможность удаленной и многопользовательской работы.

В качестве примера быстродействующей системы для отслеживания состояния корпоративной информационной сети (КИС) можно привести основанную на знаниях *систему мониторинга OMEGAMON* фирмы *Candle* (IBM с 2004 г.) . *OMEGAMON* - типичный представитель современных экспертных мультиагентных динамических *систем*, работающих в реальном времени. *OMEGAMON* позволяет за считанные минуты ввести и отладить правила *мониторинга* внештатных ситуаций для объектов КИС. Правило (situation) записывается как продукция. Логический вывод в такой ЭС реализован при помощи механизма *policy*, обеспечивающего построение цепочек логического вывода на основе *situations*. На [рис. 6.3](#) приведен один из интерфейсов для заполнения БЗ в ЭС *OMEGAMON*. На этом рисунке показана ситуация, определяющая критическое количество сообщений в очередях транспортной системы IBM *WebSphere MQ (MQSeries)*.

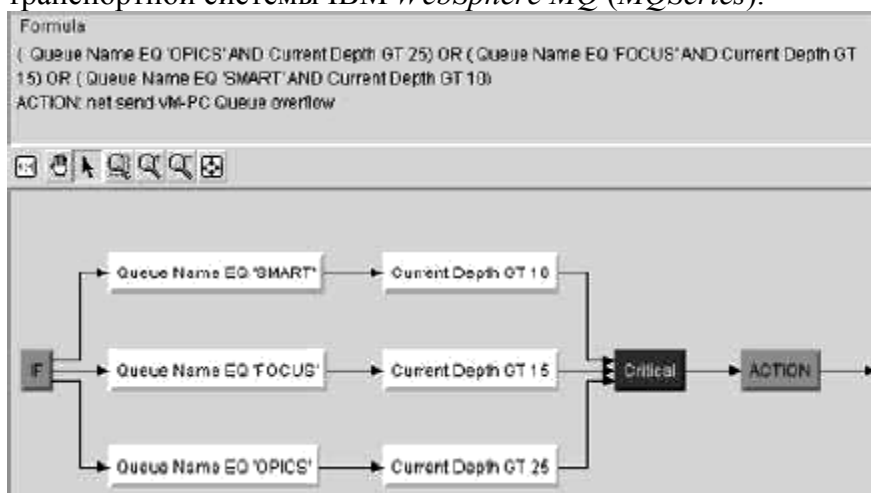


Рис. 6.3. Интерфейс OMEGAMON для заполнения БЗ

На [рис. 6.4](#) показаны основные компоненты системы *OMEGAMON*:

- сервер сбора информации от агентов *CandleManagementServer (CMS)*;
- сервер отображения результатов, оповещения пользователей и *настройки мониторинга* КИС *CandleNetPortal Server (CNP)* со своими клиентами;
- *Candle Management Workstation (CMW)* - рабочая станция администратора *OMEGAMON*;
- *Managed Systems* - компьютеры КИС, на которых работают агенты.

Агенты *OMEGAMON* работают на контролируемых системах (*Managed Systems*), как первоклассные шпионы: они незаметны с точки зрения использования CPU и оперативны при *мониторинге* с точки зрения времени поставки своих донесений в центр (*CMS*). Они фиксируют критическую ситуацию и обеспечивают реакцию (*ACTION*) менее чем за 1 секунду. Все определяется тем интервалом *мониторинга*, который задается *экспертом* на основе своих интуитивных знаний. В качестве *ACTION* при определении ситуаций можно использовать различные типы действий: посылку почтовых сообщений и sms специалистам сопровождения, посылку информации в другие системы, выполнение системных команд и т.д. Количество объектов *мониторинга* (компьютеров КИС) может достигать нескольких сотен, и на каждом объекте может быть несколько сотен контролируемых параметров. Количество платформ (типов операционных систем), на которых работают агенты, превышает 30, начиная от OS/390, OS/400, далее различные UNIX-платформы (HP_UX, AIX, Solaris) и заканчивая Windows. На одном сервере может работать несколько агентов, например, для *мониторинга WebSphere MQ (MQSeries)*, *WebSphere Application Server*, DB-2 и HP_UNIX одновременно.

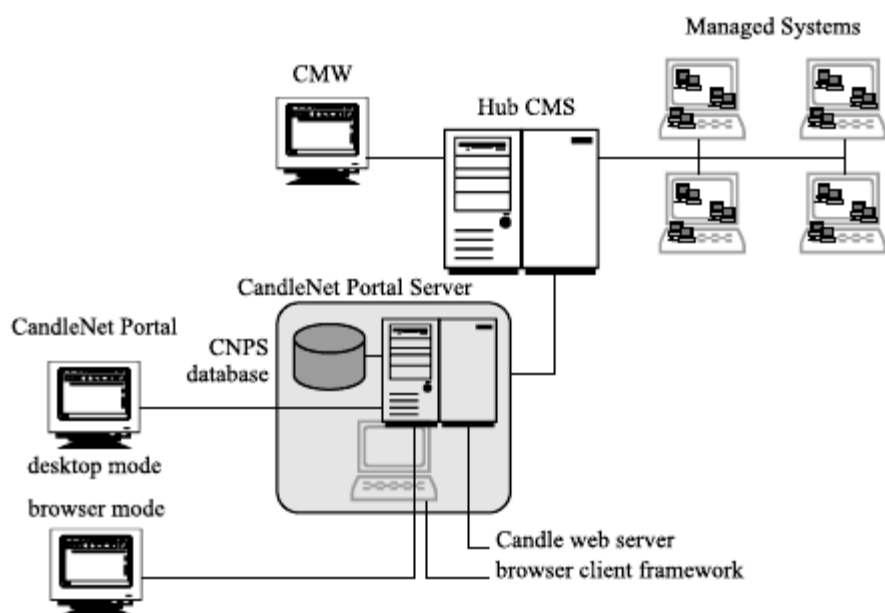


Рис. 6.4. Структурная схема ЭС OMEGAMON

Серверы CMS и CNP-servers могут работать на одном выделенном сервере, как правило, на базе операционной системы Windows. Настройка ситуаций (situations) и механизмов логического вывода (policy) производится на рабочем компьютере администратора через CNP-client. Для только что созданной ситуации вы нажимаете кнопку Apply и моментально видите отображение ACTION через CNP-client, через почту и т.д.

Следует подчеркнуть, что основанная на знаниях система *мониторинга OMEGAMON* - это весьма эффективная система управления вычислительными ресурсами, надежный и незаменимый помощник в поисках решений по оперативному устранению критических и трудных для диагностирования ситуаций, при анализе информационных потоков, анализе *производительности и настройке* КИС.

Новые технологии проектирования и анализа систем

Обзор и классификация новых информационных технологий, наиболее актуальных для анализа и моделирования систем, примеры, тенденции развития технологий.

Цель лекции: содержательное введение в ряд наиболее важных для системного анализа и моделирования новых информационных технологий, в основные тенденции развития новых информационных технологий.

Процесс извлечения (получения) информации строится на основе упорядоченных последовательных действий по сбору, накоплению, отражению, преобразованию, актуализации данных; такие процессы в информатике называются **информационными технологиями**, и их основными элементами являются технические средства и устройства, например, в телеграфе - телетайпное устройство, в телевидении - телевизор и т.д.

Новые информационные технологии - это *информационные технологии*, базирующиеся на новых, инфологических и компьютерных средствах получения, хранения, актуализации информации, знаний.

Высокие технологии - это технологии качественного изменения состава, характера, методов решаемых задач, технологии эволюции, а не функционирования.

Пример. Обычная ("старая") технология вычислений ставит основную цель - найти решение задачи за приемлемое время и стоимость. Новая технология использования математических компьютерных пакетов ставит новую цель - найти решение достаточно быстро, точно и экономично. *Высокая технология* распределенных, квантовых вычислений ставит цель - найти решение задачи, не решаемой (труднорешаемой) обычными технологиями.

В узком понимании, *новая информационная технология* - использование вычислительной техники и систем связи для создания, сбора, передачи, хранения, обработки информации; она - часть информационного бизнеса.

Любая технология базируется на научно-теоретическом, инженерно-техническом, программном обеспечении. Само по себе это ядро еще не образует технологию. Для этого оно должно быть интегрировано и поддерживаемо сетевыми пространственно-временными, организационно-людскими связями и отношениями. Должна быть система, сеть поддержки технологических отношений (TSN).

Пример. TSN системы дистанционного обучения состоит из инфраструктуры - компьютерных сетей, протоколов их взаимодействия и т.д. Хаб (маршрутизатор) - элемент этой системы, но он управляется своим программным обеспечением (например, программа переключения), своим электрическим обеспечением. Хаб сам по себе - не технология. Программа Word - сама по себе не технология (хотя ее часто называют технологией подготовки и редактирования документов), а элемент технологии, определяемой как MS Office - технологии автоматизированного, компьютеризованного делопроизводства, автоматизации работ в офисе.

Традиционная (классическая) *информационная технология*, как правило, строится на базе хорошо формализуемых, структурируемых интеллектуальных процедур. *Новая информационная технология*, как правило, строится на основе плохо формализованных и структурированных интеллектуальных процедур.

Цивилизация возможна только при наличии информации, информационных потоков и обменов в обществе.

Информация делает народы человечеством.

Сначала *информационные технологии* использовались, из-за их сложности, уникальности и дороговизны, только в научных центрах и крупных промышленных компаниях. По мере совершенствования, распространения и удешевления, *информационные технологии* проникли в разные отрасли и стали развивать их и развиваться сами, что привело к развитию потребностей общества.

Новые *информационные технологии* бывают следующих базовых типов:

- *когнитивные технологии*, направленные большей частью на получение, хранение и актуализацию знаний, принятие интеллектуальных решений;
- *инструментальные технологии*, направленные большей частью на использование в качестве инструментария, среды для построения других технологий и для обслуживания их;
- *прикладные технологии*, направленные большей частью на решение проблем некоторой проблемной области (или областей);
- *коммуникативные технологии*, направленные большей частью на решение проблем связи, коммуникаций, общения.

Отметим, что такое деление - весьма условное - и технология может с успехом быть и *прикладной*, и *когнитивной*, и *инструментальной*, и *коммуникативной*.

Пример. Такова, например, технология компьютерного моделирования, *гипермедиа*. Возможно деление (также условное) *информационных технологий* и по сфере использования, например:

- *информационные технологии* в науке;
- *информационные технологии* в образовании;
- *информационные технологии* в проектировании и производстве;
- *информационные технологии* в управлении;
- *информационные технологии* в сфере услуг;
- *информационные технологии* в сфере быта.

Можно также условно разбить все новые технологии на две группы - технологии корпоративной работы и технологии индивидуальной работы.

Рассмотрим новые *информационные технологии*, ограничиваясь содержательным простым их обзором, с учетом того, что наиболее важные *информационные технологии* анализа и синтеза систем - математическое и компьютерное, имитационное моделирование - уже были нами рассмотрены выше. Отметим лишь, что математическое моделирование - "старая" *информационная технология*, в отличие от компьютерного моделирования, являющегося новой технологией.

1.Технология баз данных (*БД*) и систем управления *БД* (*СУБД*). **БД** - достаточно большие наборы структурированных данных некоторой предметной области, представленные на машинных носителях и имеющие общую и удобную структуру, единые организационно-методические, программно-технические и языковые средства обеспечения использования данных различными программами пользователей. В зависимости от способа и технологии представления данных, различают иерархические, сетевые или реляционные базы данных, табличные или страничные. В любой *БД* задается порядок (отношение порядка) на множестве записей (полей записи), например, ключевыми полями, содержимое которых нумеруемо, лексикографически упорядочено. Таких полей может быть несколько, и при сортировке (выборке, модификации) данных записи ищутся сперва по одному ключу, затем - по другому и т.д., пока не будет совпадения или несовпадения требуемых полей. Остальные поля при этом не сравниваются. Такой процесс называется сортировкой или поиском, сравнением по ключу (ключам). Кроме поиска по ключу, можно искать и по значению, перебирая все записи *БД*, но этот процесс более длителен и часто требует построения дополнительных вспомогательных индексных таблиц для хранения подходящих по поисковому образцу значения записей (если такие есть).

В последнее время распространяется технология удаленных *БД*. Она базируется на коллективном доступе пользователей к информационным ресурсам, сосредоточенным на едином компьютере, или хост-компьютере, в диалоговом режиме по сетям передачи данных. Информационными продуктами здесь выступают *БД* разных предметных областей, а также различные директории, рубрикаторы и другие данные, облегчающие пользователю поиск по *БД*. Информационные услуги предоставляются благодаря наличию разнообразных средств поиска, обработки и выдачи информации. Информационные продукты и программные средства служат главными элементами банков данных или автоматизированных банков данных (*АБД*) - основной организационной формы, в которой развиваются современные технологии коммерческого распространения информации. Основными особенностями данной технологии, определяющими ее достоинства и ее недостатки, являются:

- предоставление пользователю только информационных услуг, а не непосредственно информационных продуктов, в результате чего он получает (оплачивает) только действительно нужную информацию;
- полнота информации, связанная с загрузкой на мощные хост-компьютеры больших массивов данных;
- высокая скорость обновления, модификации и перемещения информации;
- развитое программное обеспечение, позволяющее не только находить и получать информацию, но и при необходимости осуществлять ее графическую, наукометрическую и эконометрическую обработку.

Интерактивные услуги *АБД* могут предоставляться в режимах:

- локальном, когда работа пользователя осуществляется с терминала, подключенного к хост-компьютеру;
- удаленном, когда работа пользователя осуществляется с физически удаленного от хост-компьютера терминала по сетям связи.

Пример. В локальном режиме работают читатели библиотеки, осуществляющие поиск в *АБД*, который расположен на ее вычислительном центре, с терминалов по всему помещению библиотеки. В удаленном режиме можно работать, например, с библиотекой Конгресса США. **СУБД** (*DBMS - DataBase Management System*) - программная система, обеспечивающая общение (интерфейс) программ пользователя и данных из *БД*. Это общение происходит на

специальном непроцедурном языке логического представления данных и структур данных; сами данные описываются средствами также специального языка представления данных, программы пользователя при этом могут быть написаны на языке программирования. СУБД должна иметь средства, позволяющие сформулировать запрос к БД (поиск, сортировка и т.д.) на языке, близком к естественному и понятному для пользователя, но в то же время формальном, реализованном на ЭВМ языке. Такие языки называются языками запросов к базам данных и относятся к языкам непроцедурного типа.

Основные функции СУБД:

- управление данными во внешней памяти - обеспечение необходимых структур внешней памяти для хранения данных и манипулирования ими;
- управление буферными областями памяти - обеспечение копирования необходимой части БД в области (буфере) оперативной памяти, а также использование определенных правил манипулирования с буферами;
- управление транзакциями, т.е. последовательностями операций над БД, рассматриваемыми СУБД как одна макрооперация; каждая транзакция не изменяет БД, а, следовательно, можно выполнять различные транзакции, т.е. организовывать многопользовательскую работу с БД через СУБД, в том числе и параллельную;
- поддержание надежности хранения данных в БД через избыточность данных и журнал (часть БД, недоступная пользователям СУБД и тщательно копируемая; в нее поступают записи обо всех изменениях БД) с целью сохранения данных при сбоях аппаратуры или программы;
- поддержка языков БД (языков определения логической структуры БД, языков манипулирования данными) или единого интегрированного языка, содержащего необходимые средства для работы - от проектирования БД до обеспечения базового пользовательского интерфейса с БД.

Пример. База данных ГИБДД всех владельцев автотранспорта, из которой по запросам сотрудников ГИБДД можно оперативно извлечь, например, данные о владельце машины по номеру ее госрегистрации.

2. Технологии хранилищ данных и интеллектуального анализа данных. Хранилище данных - очень большая специализированная БД и программная система, предназначенная для извлечения, коррекции (чистка, правка) и загрузки данных из источников в БД с многомерной структурой, включая средства упрощения доступа, анализа с целью принятия решения. Интеллектуальный анализ данных (Data Mining) - автоматический поиск скрытых ("не лежащих на поверхности") в больших базах данных взаимоотношений и связей с помощью математического и инфологического анализа, выделения трендов, кластеризации (кластерного анализа), классификации и распознавания (таксономии), шкалирования и т.д. Специальные модели и алгоритмы анализа извлекают из больших баз данных (или из других хранилищ данных, например, электронных таблиц) знания, позволяющие агрегировать, интегрировать и детализировать эти данные и, самое главное, принимать на их основе решения. Это, по сути, идентификация скрытых в них зависимостей.

Пример. Хранилища данных собирают и централизуют текущую информацию о состоянии дел корпорации, о ее услугах, клиентах, поставщиках, и предоставляют аналитические и отчетные инструменты. С помощью анализа финансовых отчетов фирм, можно разбить их на классы по финансовой устойчивости, по вероятности банкротства, что поможет банку-кредитору осуществлять политику их кредитования более эффективно. Интеллектуальный анализ данных в геоинформационных системах может помочь обнаружить и визуализировать участки земной коры с залежами нефти, газа, сейсмоопасные. В бизнесе такой анализ может осуществляться для оценки надежности клиентов, выявления мошенничества, интерактивного маркетинга, анализ трендов и др. т.е. для Business Intelligence.

3. Технология баз знаний (БЗ) и экспертных систем (ЭС). БЗ - накопление, структурирование и хранение с помощью ЭВМ знаний, сведений из различных областей таким организованным

способом, что можно иметь доступ к этим знаниям, расширять их, получать, выводить новые знания и т.д.

Пример. *БЗ* по хирургическим операциям брюшной полости, из которой молодой и неопытный хирург в экстренной хирургической ситуации может извлечь необходимую информацию об операции; сама же *БЗ* разработана на основе знаний высокопрофессиональных и опытных хирургов.

ЭС - накопление опыта, знаний, умений, навыков высокого уровня профессионалов-экспертов, структурирование и хранение, актуализация с помощью ЭВМ с целью получения экспертных суждений по различным проблемам данной области.

Пример. Примером *ЭС* "Хирург" может быть экспертная система, построенная на основе приведенного выше примера *БЗ*. *БЗ* и *ЭС* тесно связаны. Примером другой *ЭС* может быть система "Таможня", которая дает возможность анализировать документацию о финансовых сделках, находить и выдавать подозрительные факты, исследовать их связи и давать рекомендации финансовым инспекторам.

4. Технология *электронной почты* и телекоммуникационного доступа к удаленной от пользователя информации, носителю информации, собеседнику - человеку или компьютеру.

Электронная почта - система передачи сообщений с помощью компьютера отправителя и приема их с помощью компьютера получателя. При этом сообщение отправителя преобразуется из цифровых кодов, например, с помощью модема, в коды электромагнитных колебаний, передаваемых по телефонным каналам, а ЭВМ адресата производит обратное преобразование. Развитие сетей связи - виртуальные локальные вычислительные сети, объединяющие пользователей не по территориальному принципу, а по профессиональным интересам. *Телеконференция* - обмен сообщениями (докладами) между участниками (подписчиками) конференции, анонсированной на специальной доске объявлений в сети, в частности, на электронной доске объявлений. *Телеконференция* представляет собой технологию на базе программных средств интерактивного доступа к ресурсам сети и предназначена для обсуждения какой-либо тематики. С помощью *телеконференций* можно проводить консалтинг, обучение, совещание, автоматизацию офиса и др. Базовая система проведения видеоконференций обычно включает: мощную рабочую мультимедийную станцию; видеокамеру и специальную плату для сжатия видеoinформации; микрофон и видеомаягнитофон; средства сопряжения с используемой для проведения конференции сетью. *Телеконференции* могут проводиться как в режиме обмена письмами по *электронной почте* (режим почтового подключения), так и в режиме терминального интерактивного подключения через телекоммуникационные сети. В режиме терминального подключения пользователь может иметь доступ (подписку) к целой системе *телеконференций*, но, в отличие от режима почтового подключения, можно подключаться к заявленной конференции непосредственно в сети, с помощью специальных программ, управляющих работой пользователей с *телеконференциями*. Эти программы позволяют выполнять следующие манипуляции: найти конференцию; подписаться на конференцию (зарегистрироваться); перейти в конференцию; послать отклик (доклад); получить отклик (доклад); закрыть подписку и другие.

Пример. Рассмотрим медицинские видеоконференции (один из наиболее убедительных и ярких социально-экономических примеров использования *телеконференции*). В крупных больницах и клиниках сейчас имеется современное медицинское оборудование - томографы, эхокардиографы и др., а также достаточно высококвалифицированный медицинский персонал, с помощью которых в режиме видеодиалога (конференции) врачи из региональных (вплоть до районных) медицинских учреждений могут обсудить результаты диагностики больного, диагноза, методов и стратегий лечения. Проблема "приближения" этих средств и кадров особенно актуальна для нашей страны, с ее большой территорией. Основные направления использования медицинских *телеконференций*:

- первичное консультирование дооперационных больных и уточнение предварительного диагноза, анализ обследования;
- постоперационные консультации и наблюдение больных;

- срочные неотложные консультации больных в критических ситуациях;
- консилиумы и консультации, обмен мнениями врачей.

Технология проведения медицинской видеоконференции:

- согласование времени проведения видеоконференции (сеанса связи);
- подготовка информации о пациенте (файлов записей из историй болезни, статических данных, например, рентгенограммы, эхокардиограммы и др., и динамических, например, видеозаписи операций и результатов анализа);
- предварительная пересылка данных по *электронной почте*;
- обсуждение в режиме видеодиалога информации о больном и диагноза;
- принятие решения, а также документирование результатов обсуждения.

В Научном центре сердечно-сосудистой хирургии им. А.Н. Бакулева РАМН проведено множество плановых и экстренных видеоконсультаций. Экономическая и социально-медицинская выгода от таких видеоконференций в 6-10 раз выше, чем от классической технологии проведения консультаций с выездом в клинику (что иногда невозможно).

5. Технология (использования) автоматизированных систем (АС) и автоматизированных рабочих мест (АРМ). АС - это человеко-машинная система для исполнения ежедневных, часто рутинных, профессионально выполняемых на рабочем месте сотрудника работ - с целью уменьшения затрат времени, сокращения числа ошибок и обеспечения оперативной связи с другими сотрудниками; интеллектуальные системы имеют также способность к перестройке технологической цепочки, они способны и к обучению.

Возможны различные системные цели автоматизации (в зависимости от типа организации, структуры): эффективное управление потоками материальных, трудовых, информационных, энергетических ресурсов, повышение социальных, экономических и технических показателей системы и других систем (для управляющих систем); минимизация риска невыполнения планов и максимизация качества принятых решений, повышение конкурентоспособности, рентабельности (для производственных систем); получение новых знаний, повышение престижа в области инноваций, расширение сферы использования результатов исследования, создание "ноу-хау", обеспечение экономической эффективности (для научно-исследовательских организаций); минимизация риска невыполнения заказа или услуг, повышение конкурентоспособности, повышение экономической эффективности функционирования (для обслуживающих организаций); повышение престижа, совершенствование учебного процесса, переход к новым формам обучения, к дистанционному образованию (для образовательных организаций).

В последние годы распространилась концепция корпоративных и распределенных систем в народном хозяйстве, в которых широко используются локальные информационные системы. Для реализации идеи распределенного управления необходимо создание автоматизированных рабочих мест на базе профессиональных компьютеров (рабочих станций).

АРМ - предметно-ориентированная инструментальная АС, устанавливаемая непосредственно на рабочем месте специалиста и предназначенная для автоматизации профессиональной деятельности (сидящего за этим рабочим столом сотрудника). Можно их определить как автоматизированные системы локального характера, соответствующие некоторому функциональному назначению. Несмотря на различный характер задач, решаемых с помощью АРМ, принципы создания любых АРМ должны быть общими:

- системность, во-первых, подхода к проектированию и решению задач и, во-вторых, возможность работы в составе сети, системы;
- гибкость, приспособляемость, адаптируемость к изменениям задач;
- устойчивость, надежность в работе, восстанавливаемость при сбоях;
- эффективность (по затратам, повышению производительности труда);
- быстрота отклика - минимум времени на каждый шаг диалога с пользователем;
- полнота выполняемых функций, решаемых профессиональных задач;

- интерактивность - возможность вмешиваться в диалог, выбирать следующий шаг диалога, например, в форме команд на специальном командном языке, в форме выбора объектов, в форме "меню", в смешанной форме;
- функциональность, дружелюбность, эргономические характеристики и удобство использования, в частности, ориентация на непрофессионала в области компьютерной подготовки

и др.

Пользовательский интерфейс *АРМ* часто организуется с помощью понятия рабочего стола на экране. Экран делится на три части (три объекта). Первая (обычно верхняя часть) - строка меню, с ее помощью осуществляется доступ к другим объектам. Вторая часть (обычно нижняя часть) называется строкой состояния, с ее помощью быстро вызываются наиболее используемые объекты или отображается важная текущая информация. Третья часть (основная, средняя часть экрана) называется рабочей поверхностью (поверхностью стола), с ее помощью отображаются все объекты, вызываемые из меню или из строки состояния. Такая форма организации диалога человека и машины наиболее удобна, и многие программы используют именно ее. Программные средства *АРМ* - часть инструментального программного обеспечения.

Пример. *АРМ* секретаря-референта должен включать редактор текстов, электронную таблицу, переводчики, органайзер и др. *АРМ* студента-экономиста должен иметь электронные учебники по изучаемым дисциплинам, обучающие программы и среды, электронные справочники и энциклопедии, переводчики, органайзер и др. *АРМ* администратора базы данных должен быть оснащен *СУБД*, электронным журналом администратора и др. *АРМ* управляющего должен обладать средствами описания управленческой деятельности в виде сетевого графика, системой контроля исполнения, системой согласования документов, системой электронной подписи, системой ведения совещания и др. *АРМ* статистика (работника статистической службы) должен включать информационно-справочные базы и материалы, средства его обслуживания и поддержки, средства анализа статистических данных для оперативного составления докладов, справок и отчетов, в частности, для анализа динамических рядов и проверки статистических гипотез. *АРМ* банковского служащего и банковские системы, - наиболее развиваемые системы. Они содержат программное и техническое обеспечение как специального назначения (например, для банковских расчетов и операций с банкоматами), так и для обеспечения безопасности таких систем. Банковские системы используют локальные вычислительные сети, специализированные бизнес-серверы, ЭВМ общего назначения, технологию "*клиент-сервер*" и, часто, ОС UNIX, объединение несколько локальных сетей, межсетевой обмен и удаленный доступ филиалов банка к ресурсам центрального офиса банка для выполнения электронных платежей, транзакций. Банковские системы должны иметь средства адаптации к конкретным условиям эксплуатации. Для поддержки оперативной работы банка, банковская система должна функционировать в режиме реального времени OLTP (On-Line Transaction Processing, Онлайновая Обработка Состыкровок). Основные их функции:

- автоматизация всех ежедневных внутрибанковских операций, ведение бухгалтерии и составление сводных отчетов;
- обеспечение связей с филиалами и отделениями;
- автоматизированное взаимодействие с клиентами (система "банк-клиент");
- анализ деятельности банка и выбор оптимальных в данной ситуации решений;
- автоматизация розничных операций - применение банкоматов и кредитных карточек;
- межбанковские расчеты;
- автоматизация деятельности банка на рынке ценных бумаг (мониторинг курсов, объемов сделок и т.д.);
- оперативная информационная поддержка финансовой и кредитной политики банка.

Необходимо отметить, что автоматизация учреждения, организации заключается не только в обеспечении работников *АРМ* локальной сетью, *электронной почтой* и т.п., но и в

создании новой технологии совместной работы и управления с целью эффективной работы всего учреждения. Это своего рода информационная, экономическая (корпоративная) культура совместной работы, использования (актуализации) профессиональных знаний, умений и инноваций. При этом знания каждого сотрудника могут быть актуализированы другими сотрудниками, обсуждены (проголосованы) и применены ими.

6. Технологии *компьютерного* (компьютеризированного) *офиса* коллективной работы в офисе. **Компьютерный офис** - офис, в котором имеется высокий уровень компьютеризации, внедрения АРМ, систем делопроизводства, так, что вся профессиональная деятельность офиса может быть успешно автоматизирована.

Пример. *Компьютерный офис* - это, например, офис, где работа осуществляется с использованием локальных сетей связи и интегрированной программной среды Microsoft Office, которая включает в себя все основные программные пакеты для выполнения типовых и регулярно выполняемых операций, работ в офисе, в частности, ведение делопроизводства, контроль исполнения и др. Microsoft Office имеет встроенный простой язык программирования - Visual Basic for Applications (VBA). Этот язык позволяет создавать новые приложения или корректировать и связывать старые, выполняемые в среде Microsoft Office, а также расширять возможности офиса, его используемых приложений. Стандартное ядро Microsoft Office включает:

- редактор текстов Microsoft Word (функции редактора - набор, именованное и сохранение текста, модификация, переименование и перемещение текста или его отдельных фрагментов, вставка различных формул, графиков, таблиц, диаграмм и др.);
- электронную таблицу Excel (функции - обработка, хранение и модификация в произвольных таблицах чисел, строк, столбцов, формул, по которым динамически изменяются числа, строки и столбцы);
- систему для презентаций (презентационный пакет) PowerPoint (функции - создание и проецирование на большом экране электронных презентаций, слайд-шоу, ярких пленок для проектора, раздаточных печатных материалов);
- систему управления базами данных Access (реляционная СУБД, доступная любому пользователю и позволяющая быстро и эффективно организовывать, анализировать, перемещать, вести поиск и т.д. для больших массивов информации, без дублирования информации в них), например, по шаблонам создания базы данных: Адресная книга - создает базу данных типа адресной книги, Библиотека - создает базу данных типа библиотеки, Контакты - создает базу данных типа контактных связей и др.

Более совершенные версии Microsoft Office-97, -2000, кроме приведенных стандартных приложений, имеют и следующие приложения:

- Office Assistant - помощник для подсказок;
- HTML- и Web-поддержку (Internet Assistants);
- различные программы-помощники (Graph - графическое представление данных, Organization Chart - создатель штатного расписания офиса, Equation Editor - редактор формул, WordArt - создатель логотипов, заголовков, ClipArt Gallery - для просмотра рисунков) и др.

В состав Microsoft Office-2000 входят, и в ее дальнейшие модификации будут входить, возможности одновременного показа презентаций по локальной сети, использование возможностей системы распознавания речи, визуальные среды разработки различных офисных приложений (например, заполнения платежных поручений), сайты *рабочих групп*, системы визуализации данных, система сканирования и ввода данных и др.

Технология "**Рабочая группа**" - технология совместной работы нескольких связанных между собой общими информационными ресурсами компьютеров ("*рабочей группы*"), объединенных для решения какой-либо общей задачи.

Пример. Типы *рабочих групп*: "Дирекция", "Бухгалтерия", "Канцелярия". Компьютерная сеть организации может объединять несколько *рабочих групп*. У каждого компьютера *рабочей группы* имеется идентификатор, имя в группе, например, по ФИО человека, на нем

работающего. В *рабочей группе* "Бухгалтерия" может существовать компьютер (рабочее место) "Главбух" или "Иванов Сергей Николаевич".

Рабочая группа может быть и временной - для работы над конкретным проектом в пределах определенного промежутка времени.

Пример. Можно организовать *рабочую группу* "Презентация фирмы", которая состоит из компьютеров сотрудников фирмы, подготавливающих презентацию своей фирмы, или "Годовой отчет" - для подготовки годового финансового отчета фирмы. Все эти люди могут работать в разных отделах, но они составляют временную *рабочую группу*, чтобы было легко обмениваться информацией общего доступа при работе над отчетом.

Обмен информацией может происходить и между *рабочими группами*. Для этого не нужно физически перемещать компьютеры: чтобы сформировать *рабочую группу*, достаточно присвоить всем компьютерам, входящим в состав группы, ее имя.

Пример. Операционная система Windows for Workgroups позволяет выделение компьютеров в *рабочие группы* при ее инсталляции. Изменять состав и структуру *рабочей группы* затем можно из "Панели управления", запустив прикладную программу Network (сеть). При этом все компьютеры одной сети, независимо от их объединения в *рабочие группы*, имеют доступ к общим принтерам и общим файлам, а такие приложения как Mail (*Электронная почта*), Schedule+ (Ежедневник), работают только в пределах одной *рабочей группы*. Передача почты через Mail возможна только в пределах одной *рабочей группы*. Как правило, в небольших фирмах имеется одна *рабочая группа*.

Технология (модель взаимодействия) "**Клиент-сервер**" - это технология взаимодействия компьютеров в сети, в которой каждый из компьютеров имеет свое рабочее назначение. Один, более мощный, компьютер (сервер) в сети владеет и распоряжается информационными и аппаратными ресурсами (процессор, файловая система, почтовая служба, база данных и др.), другой, менее мощный ("клиент"), имеет доступ к этим ресурсам лишь через сервер.

Этот принцип распространяется и на взаимодействие программ и информационных сред. Программа (среда), выполняющая предоставление соответствующего набора услуг - "сервер", а программа (среда), пользующаяся этими услугами - "клиент". Технология традиционной модели "**клиент-сервер**" модернизируется и совершенствуется.

Пример. Сейчас говорят уже о принципиально иной концепции взаимодействия между элементами сети peer-to-peer (P2P), позволяющей отдельным компьютерам работать друг с другом напрямую.

7. Технологии использования интегрированных пакетов прикладных программ (**ППП**) - технологии на базе специальным образом организованных комплексов программ для решения различных классов однотипных и часто встречающихся задач из различного типа предметных областей. Современные **ППП** имеют диалоговую, интерактивную обратную связь с пользователем в процессе постановки задачи, решения и анализа результатов. При решении задач применяют обычно используемый в предметной области интерфейс. "Интеллектуальность" **ППП** - возможность постановки задачи содержательно, не указывая алгоритма ее решения. Построение алгоритма решения и сборка целевой программы производятся автоматически и скрыто от пользователя. Предметное обеспечение **ППП** - база знаний о методах, алгоритмах решения задачи и о самих задачах. Программирование осуществляется в терминах предметной области, ЭВМ используется уже на этапе постановки задачи, решение задач - с помощью автоматического построения цепочки программ, по ходу накопления знаний о решаемой задаче, т.е. возможно пополнение базы знаний **ППП**. Используются инструментальные **ППП**, которые ускоряют и упрощают процесс создания **ППП** и снижают его стоимость. Для этого метода характерна высокая интеграция: наполнение **ППП** само состоит из **ППП** различного назначения.

Пример. В качестве примера интегрированного **ППП** приведем пакет MathCAD, предназначенный как для сложных математических вычислений, так и для несложных (в режиме инженерного калькулятора).

8. Технологии машинной графики и визуализации - технологии, базирующиеся на системах рисования и черчения различных графических объектов и образов с помощью ЭВМ и устройств рисования (например, плоттеров), а также их визуального, наглядного представления. Особо следует отметить средства **анимации** - "оживления" изображений на экране, т.е. методы и средства создания динамических изображений, иначе говоря - компьютерных мультфильмов.

Пример. Примером средств машинной графики может служить программный комплекс изображения пространственных объектов и их динамической актуализации - пакет "3D-Studio". Этот пакет позволяет не только создавать трехмерные сцены, но и использовать их при реализации компьютерных анимационных ситуаций (мультипликаций) с использованием различных графических файлов разных форматов, что дает возможность применять при разработке мультфильмов известные графические пакеты: CorelDraw, PhotoPaint и др. 3D-Studio имеет модульную структуру, состоящую из пяти модулей, за каждым из которых закреплены задачи конкретного типа, решаемые в строгой последовательности. Первый модуль (2D-Shaper) является основным инструментом создания и редактирования плоских фигур, а также снабжения других модулей особыми геометрическими структурами, формами и траекториями. Для преобразования плоских фигур в трехмерные каркасные объекты имеется модуль 3D-Lofter, в который включены мощные средства генерации сложных пространственных форм и структур. Подготовленные двумерные планы моделей отображаются ("выдавливаются") в третье измерение по специально заданным траекториям. Модуль 3D-Lofter снабжен средствами деформации, например, по осям, что позволяет создавать трехмерные объекты более сложных форм. Можно построить 3D-фигуру по трем проекциям на координатные плоскости.

Современные технологии 3D-графического моделирования позволяют строить полные трехмерные объекты по их эскизам.

Пример. Программная среда LightWave Modeler, позволяющая, используя соответствующие графические примитивы, формировать графический персонаж, который легко анимируется по желанию (масштаб, направление, цвет и т.п.) пользователя.

9. Гипертекстовые технологии. **Гипертекст** (Hypertext - "сверхтекстовая, надтекстовая".) - эта технология на базе средств обработки больших, глубоко вложенных, структурированных, связанных семантически и понятийно текстов, информации, которые организованы в виде фрагментов (текста), которые относятся к одной и той же системе объектов, расположенных в вершинах некоторой сети и выделяемых обычно цветом; они дают возможность при машинной реализации быстро, нажатием нескольких клавиш, вызывать и помещать в нужное место просматриваемого или организуемого нового текста заданные фрагменты *гипертекста*, т.е. тексты, "привязанные" к выделенным по цвету ключевым словам или словосочетаниям; гипертекстовая технология позволяет определять, выбирать вариант актуализации информации *гипертекста* в зависимости от информационных потребностей пользователя и его возможностей, уровня подготовки, т.е. жестко и заранее не определяет сценарии диалога. При работе с гипертекстовой системой пользователь может просматривать документы (страницы текста) в том порядке, в котором ему это больше нравится, а не последовательно, как это принято при чтении книг, т.е. *гипертекст* - нелинейная структура. Достигается это путем создания специального механизма связи различных страниц текста при помощи гипертекстовых ссылок, т.е. кроме линейных ссылок обычного текста типа "текст-предшественник - текст-преемник", у *гипертекста* можно построить еще сколь угодно много других динамических ссылок, ассоциированных с документом в целом или только с отдельными его фрагментами, т.е. контекстные ссылки.

Пример. Примерами *гипертекстов* могут быть электронные журналы.

10. Средства и системы *мультимедиа* (multimedia) и *гипермедиа* (hypermedia). Медиа - "среда или носитель информации". Мультимедийность, многосредность - актуализация различных сред и чувств восприятия информации: средства озвучивания, оживления - мультипликации,

графического и наглядного представления входных и выходных данных задачи и сценариев решения или даже самого решения.

Пример. Примерами средств *мультимедиа* могут служить звуковые карты (Sound Blaster) для генерирования на ЭВМ широкого диапазона звуков, активные звуковые колонки для их передачи и устройства считывания информации с компакт-дисков - CD-ROM, позволяющие считывать большие объемы информации, например, некоторую сложную и длительную музыкальную композицию, а затем воспроизводить с использованием предыдущих двух средств *мультимедиа*.

Средства *гипермедиа* - средства на основе синтеза концепции *гипертекста* и *мультимедиа*, т.е. в гипертекстовые фрагменты могут быть "встроены" мультимедийное сопровождение, мультимедийные приложения: hypermedia=hypertext+multymedia.

Пример. Глобальной гипермедийной системой является WWW (World Wide Web - "Всемирная Паутина") - система навигации, поиска и доступа к гипертекстовым и мультимедийным ресурсам Интернет в реальном масштабе времени. Глобальной ее можно считать потому, что, в отличие от обычного (локального) *гипертекста*, ссылка на документ в нем (осуществляемая одним или несколькими щелчками мыши) может привести не только к другому документу (как в локальном *гипертексте*), но и к другому компьютеру (WWW-серверу), возможно, в другом полушарии. Работа ведется с помощью универсальной программы-клиента, которая позволяет объединить в единое целое клиента и сервер. Для доступа к WWW-серверу (информации на нем) необходимо знать адрес сервера, например, адрес <http://www.mark-itt.ru> - сервер со списком российских WWW-серверов, http (HyperText Transfer Protocol) - протокол работы с *гипертекстом*. Имеется система автоматического поиска по определенным ключам (запросам, разделам). Информация в WWW представлена в виде гипертекстового документа, включающего в себя различные типы данных (текст, графика, видео, аудио, ссылки на другие гипертекстовые документы и т.д.). Такие документы называют WWW-страницами (WWW-pages). Эти страницы просматриваются с помощью браузеров, специальных программ для навигации по сети. Страницы хранятся на компьютерах-узлах сети, которые называют сайтами (site). Каждый компьютер имеет свой уникальный IP-адрес URL (Uniform Resource Locator - универсальный локатор ресурсов), с помощью которого браузер знает, где находится информация и что надо с ней делать. Страница - основной элемент WWW. На них находится та информация, которую мы ищем в сети, или ссылки на эту информацию. Страницы, *гипертекст* - это легкая и быстрая в использовании, чрезвычайно мощная система связанных ключевых слов и фраз (ссылок), позволяющая ссылаться на другие ключевые слова и фразы других страниц. Эти ссылки обычно выделены другим цветом, и достаточно просто щелкнуть мышкой по выделенной ссылке, чтобы перейти к информации, на которую отсылает эта ссылка. Для создания гипертекстовых приложений (например, личной WWW-страницы) используется специальный язык HTML (HyperText Markup Language), позволяющий создавать гипертекстовый документ в любом текстовом редакторе формата ASCII, с подключением графических файлов двух основных форматов GIF, JPEG.

По мнению ряда исследователей, следующей после Web формой коллективного сосуществования компьютеров будет Grid, которая даст пользователям больше возможностей для работы с удаленными машинами. Если World Wide Web можно сравнить с аналоговой телефонной сетью, способной передавать тексты, аудио и видео, то Grid подобна современной системе электроснабжения, предоставляющей потребителям столько ресурсов, сколько им необходимо. В настоящее время в ЦЕРНе ведутся работы по определению стандартов для Grid. Так же, как и WWW, новая концепция, в первую очередь, будет востребована в исследовательских кругах. Ученые с помощью Grid будут получать доступ к ресурсам, необходимым для решения их задач. Архитектура Grid трехслойна: интерфейс, слой приложений и операционная система Grid, позволяющая подключить пользователей к распределенным ресурсам.

11. Нейро-математические и нейро-информационные технологии и сети. **Нейротехнологии** - технологии на базе моделей, методов, алгоритмов, программ, моделирующих, имитирующих нейронные сети и процессы решения задач *искусственного интеллекта*; позволяют эффективно реализовывать параллелизм, самообучение, распознавание и классификацию, адаптивность, перестройку структуры, топологии.

Пример. Идентификация личности в криминологии; выбор управляющих воздействий в сложных системах; геологоразведка; диагностика в сейсмологии. Нейросистема VNS-736, например, позволяет обрабатывать (распознавать) изображения размером 512_512 элементов.

Пример. Одним из распространенных зарубежных нейросистем является пакет Brain Maker. Пусть необходимо решить задачу прогноза цены закрытия на сегодняшних торгах по валютным тысячедолларовым трехмесячным фьючерсным контрактам. Пусть нас устраивает точность прогноза, при которой правильно указывается ценовой тренд (подъем, спад) и изменение цены с точностью не ниже 90% от последнего скачка. Применение нейронной сети начинается с подготовки входных данных: курс доллара, индекс инфляции, ставка межбанка, биржевые индексы, объем торгов, количество сделок, максимальные и минимальные цены и др. После предварительной настройки сети начинается итерационный процесс обучения, в результате которой нейросеть настраивает свою логическую структуру для точной реакции рынка на те или иные воздействия. Для этого в пакете Brain Maker предусмотрен мощный аналитический блок, который позволяет увидеть, какие параметры оказывают позитивное влияние на ситуацию, а какие - негативное. Затем сеть снова обучается и далее тестируется на качество и адекватность, и после удачного тестирования используется для прогнозов. За десять биржевых дней сеть ни разу не ошиблась в знаке отклонения фьючерсных котировок, а девять дней из десяти отклонение прогноза от реальной цены составило менее 10 рублей. BrainMaker - это программа, с которой началась история применения нейронных сетей в России. В этом пакете на профессиональном уровне реализована классическая многослойная нейронная сеть. Это единственная программа, в которой есть возможность настройки всех параметров нейронных сетей и алгоритмов обучения. В последнее время BrainMaker чаще всего используется не как самостоятельная программа, а как надстройка к программе TradeStation для анализа в режиме реального времени. NeuroShell, хотя и является универсальной программой, но благодаря тому, что она была первой русифицированной нейросетевой программой с удобным интерфейсом, ей удалось завоевать широкое распространение на российском рынке. Для решения финансовых задач NeuroShell имеет модуль рыночных индикаторов, позволяющий использовать более 20 индикаторов технического анализа при работе с нейросетью.

Пример. Известное семейство российских программ NeuroScalp построено по модульному принципу. Базовым модулем является модуль классического технического анализа, в который интегрируются дополнительные модули, реализующие различные методы анализа финансовых рынков. В настоящее время доступны следующие дополнительные модули: "Экспертный модуль, российский рынок акций" - модуль, содержащий готовые нейросети для трех российских акций: РАО Газпром, РАО ЕЭС России, НК Лукойл; "Модуль Нейронных сетей" - эмулятор классических многослойных нейронных сетей с использованием генетических алгоритмов; "Модуль Карты Кохонена" - модуль, реализующий карты Кохонена в приложении к финансовым рынкам; "Модуль Статистика" - модуль статистической обработки финансовой информации и анализа рынка. NeuroScalp имеет удобный интерфейс и реализует необходимое множество методов, требуемое для реализации различных идей пользователя.

12. Технология виртуальной реальности, виртуальная реальность - технологии актуализации различных гипотетических сред и ситуации, не существующих реально и возможных как варианты развития реальных аналогов систем реального мира; эти технологии и системы позволяют управлять виртуальным объектом, системой путем моделирования законов пространства, времени, взаимодействия, инерции и др.

Высшая форма развития *компьютерного офиса - виртуальный офис и виртуальная корпорация* - офисы и корпорации, не существующие в обычном, классическом виде ("имеющих вывеску, штат, здание"), а созданные воображаемо, распределенно - как в пространстве, так и во времени (отделы и сотрудники могут находиться даже на различных континентах, общаясь по работе с помощью ЭВМ и сетей связи). Они являются высшей ступенью делового сотрудничества и в корне меняют организацию работ и систему информационного обеспечения сотрудников.

Виртуализация свойств и атрибутов корпорации, динамическое (а иногда и виртуальное!) выделение их общих фундаментальных (родовых, классовых) свойств, их описание и использование в рамках единой технологии, позволяет сократить промежуток между прогнозируемым (или имитируемым) состоянием корпорации и его реальным состоянием. У корпорации общие интересы появляются, актуализируются, виртуализируются на тот период, когда они служат общей цели. Успех *виртуальной корпорации* (ее разработки, внедрения и сопровождения) зависит от полноты и качества информационных потоков между объектами корпорации. При этом новые сотрудники корпорации (или сотрудники новой корпорации) имеют дело, в первую очередь, с информационными моделями, например, с моделями склада сырья, изделия, менеджера, поставщика, банка (банковских расчетов). Таким образом, *виртуальные компьютерные корпорации* поддерживают широкий спектр работ и услуг - начиная от соединения предприятий по ресурсам, производству, сбыту, снабжению, управлению, информационному обеспечению и т.д., обеспечивая базовую компетентность - совокупность знаний и умений по организации, координации и согласованию общих организационных, материальных, информационных ресурсов элементов корпорации, поддержки и развития ее инфраструктуры (архитектуры, инноваций, активов, рекламы и др.) и заканчивая обеспечением корпоративных функций системы с меньшими затратами и более качественно, уменьшая бюрократизм и уровни иерархии в системе, а также время реакции на изменения на рынке.

Основные характеристики *виртуальной корпорации*: наличие основного вида бизнеса (деятельности) для всех подсистем; концентрация пользователей вокруг этого вида деятельности и общие взаимосвязанные цели, планирование и ресурсообеспечение, общие (интегрированные) стратегии поведения и актуализации ресурсов, общая (интегрированная) технология актуализации ресурсов.

В *виртуальной корпорации* "новые рабочие или служащие" будут в основном иметь дело с инфологическими моделями рабочих мест, инструментов, сырья, поставщиков, рынков сбыта и услуг, "новые менеджеры" будут принимать оперативные и более интеллектуальные решения по гораздо более широкому кругу вопросов, а "новое руководство" будет занято многокритериальными стратегическими проблемами. Следовательно, от них требуется качественно новый уровень профессионализма, ответственности и стремление к самообразованию.

Пример. *Виртуальная маркетинговая корпорация "Да Винчи"* объединяет ряд горнорудных месторождений, производственные (машиностроительные и строительные), транспортные, инвестиционные, экологические системы. Все подсистемы "Да Винчи" поставляются без доработок под конкретный объект (как детские конструкторы сборно-разборного типа). Один из сценариев, предлагаемых в проекте (Venture Management Model), моделирует нижеследующую ситуацию. Горнодобывающая компания ведет разработки в Новой Гвинее. Построенный в этой местности отель может быть расширен для обслуживания растущего потока деловых клиентов этой компании, а также туристов. Консорциуму, имеющему бизнес в сфере коммуникаций и гостиничных услуг, предлагается долевое участие в развитии этой местности и эксплуатации отеля. Для снижения накладных расходов на расширение отеля и инфраструктуры туризма привлекаются крупные строительные компании (на условиях долевого участия в прибылях). Отметим при этом, что критерии эффективности бизнеса в таком составе - различны, а процесс принятия стратегических решений сопряжен с конфликтными интересами партнеров, динамически изменяющейся их картиной. Для

реализации этой корпорации имеются *электронная* (мультимедийная) *почта* для поддержки процессов принятия решений первыми лицами, средства *телеконференций* для функциональных подразделений и аналитиков, геоинформационная система, САПР, взаимодействующая с *СУБД* через структуру данных с пространственной привязкой, система компьютерного делопроизводства на всех этапах. Используются современные технологии типа "*клиент-сервер*" и объектно-ориентированные под Windows NT, Windows-95 (рабочие места), Unix (сервер), полные версии MS Office и компьютерный документооборот. В системе электронного документооборота используются: полнотекстовый поиск, доступ к проектной документации на всех этапах жизненного цикла проекта, подготовка интерактивной технической документации. Документ может содержать текст, например, HTML-документ, иллюстрации в одном или нескольких слоях, редакторские правки и комментарии участников различных *рабочих групп*, участвующих в проекте, трехмерные объекты из программ САПР, подключаемые к документу видео- и аудиофайлы.

Пример. Технологии виртуальной реальности широко используют различные тренажеры для обучения пилотов самолетов, водителей автомобилей, капитанов судов, которые позволяют помещать обучаемого в соответствующие воображаемые ситуации (включая и аварийные), в том числе и никогда не существовавшие в реальности и не "укладывающиеся" в рамки законов классической механики, физики; эффекты виртуальной реальности создаются часто за счет одновременного воздействия на различные органы чувств, включая подсознание, сенсомоторику. Интересны проекты создания хирургических тренажеров с использованием методов и средств виртуальной реальности. Важной формой виртуальной реальности (виртуального понятия) является рынок. Если раньше под рынком понималось реальное место встречи продавцов и покупателей, то теперь это понятие состоит из экономических, коммерческих, производственных и коммуникационных отношений и систем; они теперь могут встречаться и реализовывать свои функции в компьютерных системах.

13. **Когнитивные технологии** - методы, средства и приемы, обеспечивающие визуальное, гипермедийное представление условий задач и/или предметной области, которое помогает находить или стратегию решения (или само решение), либо позволяет оценивать и сравнивать пути решения, принять тот или иной адекватный выбор.

Пример. Когнитивная графика, позволяющая геометрически, образно представлять предметную среду и построить, исходя из этого, требуемый графический объект, в частности, пространственное представление этого объекта. Есть и средства, и методы визуального программирования (проектирования программ), в частности, среда Visual-C. Когнитивные методы выбора решений в области бизнеса позволяют принимать решения и определять стратегии поведения на основе качественных данных, личностных суждений (эффективно для ликвидации неопределенностей). Например, модель принятия решений Института США и Канады РАН, используя блоки, подмодели типа "Мир", "Ценность", "Средство", "Интерес", "Стереотип", "Цель", "Сценарий", "Проблема", позволяет изменять содержательное наполнение этих блоков, генерировать новые цели и сценарии (используя старые).

14. Технологии информационного **реинжиниринга** - методы и средства коренного пересмотра, перепроектирования информационных сетей и процессов с целью достижения резких, например, "порядковых" улучшений в ключевых показателях информационных сетей и систем, в частности, по показателям типа "производительность-стоимость", "время-объем информации", "функционирование-документация", "технология-удобство" и др. Реинжиниринговые мероприятия изменяют работу (из моноплановой она становится многоплановой), роль работника (от подконтрольного исполнителя - к принятию самостоятельных решений), оценку эффективности работы и оплаты труда (от оценки трудозатрат - к оценке результата), роль менеджера (от контролирующей функции - к тренинговой), и, самое главное, организационную структуру (от иерархической - к матричной и сетевой).

Пример. Сокращение времени принятия решения и цикла подготовки и подписания документов, например, средствами компьютерного делопроизводства, сетями связи и экспертными системами, которые обеспечивают доступ руководителя, принимающего решение, ко всем этапам, узлам и инструментариям подготовки решения; перенос акцента с проблемы "Как делать?" на проблему "Что делать?". Отметим, что популярную в информационных системах, сетях технологию "удаленный сервер данных и клиентский доступ" можно считать реинжиниринговой. *Реинжиниринг* системы подготовки докладов, отчетов, например, может свести цикл подготовки отчета с 20-30 операций до 5-10. Простое усовершенствование не может дать таких результатов.

15. Объектно-ориентированные технологии, технологии объектно-ориентированного анализа (технологии представления и актуализации информации, информационных процессов, систем как совокупностей объектов и классов с использованием следующих понятий: объект, экземпляр класса - все то, что может быть полно описано некоторыми атрибутами состояния; класс - совокупность объектов с одинаковыми атрибутами; инкапсуляция - скрытие внутренней информации, возможность отделения объектов и классов от внешнего мира; наследование - возможность создавать из классов-родителей новые классы-потомки, сохраняющие атрибуты и свойства родителей; полиморфизм - способность объектов выбирать метод представления на основе типов данных, актуализируемых сообщений).

Инструменты объектно-ориентированного анализа: атрибуты (описания объектов, классов); операции (процессы, применяемые к классам объектов); потоки данных (группы элементов данных, реализующие связи между объектами); наследование (агрегирование и обобщение).

Пример. Объектно-ориентированные среды программирования, например, C++, Smalltalk; объектно-ориентированный инжиниринг или набор приемов и методов проектирования бизнеса, наиболее эффективно обеспечивающих заданные цели и прибыль; объектно-ориентированный пользовательский интерфейс, использующий, например, понятия "класс описаний", "класс языков", "класс операционных сред" и др. При объектно-ориентированном программировании в среде языка APL, например, процедуры исполняются в соответствии с логикой и инструкциями некоторой программы, которая определяет последовательность и содержание действий; выполнение этой программы инициируется с помощью сообщения, посылаемого заданному объекту пользователем, другой программой или объектом. Получатель сообщения решает, какая программа будет выполнена.

Пример. HTML - статичное средство. Чтобы "оживить" содержимое Web, сделать интерактивные HTML-страницы, используется среда JavaScript. Одной из важных для информатики объектно-ориентированных систем является Java-система, сред - Java-интерпретирующая машина, технологий - Java-технология. Рассмотрим их вкратце. В основе всех их лежит язык программирования Java, ориентированный на сеть Internet и серверы WWW. Язык Java произошел от языка программирования Oak, с синтаксисом, близким к синтаксису языка C++. Средствами языка Java можно разрабатывать приложения для различных платформ: Intel Pentium, Macintosh, Sun и др. Java-программы бывают автономного использования (выполняемые в режиме интерпретации на конкретной компьютерной платформе) и апплеты, (applets), выполняемые в режиме интерпретации виртуальной Java-машиной, которая встроена практически во все современные браузеры. Апплеты Java встраиваются в документы HTML, хранящиеся на сервере WWW. С помощью апплетов можно сделать страницы сервера Web динамичными и интерактивными. Все данные для обработки апплеты могут получить только от сервера Web. Язык Java является объектно-ориентированным и имеет объемную библиотеку классов, значительно упрощающих разработку приложений, так как программист больше внимания может уделить функциональной части приложения, а не организации интерфейса, динамических массивов и т.п. В широком смысле, Java - это технология, изначально рассчитанная на интеграцию с сетевой Web-средой, полностью независимой от платформы. Виртуальная Java-машина - машина, на которой исходные Java-программы интерпретируются в коды этой машины. Это

делает Java-среду мощным и удобным средством разработки клиентских компонентов Web-систем. В Java-среде пользователь может осуществлять динамическую загрузку объектов из сети, т.е. ему не нужны дорогостоящие работы по наладке, администрированию клиентских Java-систем, так как для обеспечения работы клиента на новой версии достаточно загрузить ее на сервере. Имеются инструментальные среды, например, Java Studio, позволяющие проектировать приложения вообще без программирования, из готовых компонент, устанавливая между ними связи и отношения в соответствии с внутренней логикой приложения. Для повышения производительности Java-приложений в браузерах используется компиляция Just-In-Time compilation ("на лету"). При первой загрузке апплета его код транслируется в обычную исполняемую программу, которая сохраняется на диске и запускается. В результате общая скорость выполнения апплета увеличивается в несколько раз.

16. **Средо-ориентированные технологии** (интерактивные технологии проектирования, разработки, актуализации информационных систем, в которых сперва строится нужная среда, инструментарий, а затем происходит их автоматизированная настройка с помощью выполнения процедур типа: переместить, вставить, удалить, указать, активизировать и др.; готовые среды, "как кубики", объединяются в нужные структуры, а затем настраиваются на конкретные классы проблем или пользователей, причем изменения одних из них могут изменять и другие).

Пример. Средо-ориентированные системы программирования, в которых часто используется "оконный интерфейс", "оконная среда". Они основываются на понятиях "окно", "рамка", "фрейм", "рисунок на экране", "оконное меню" и др., каждый из которых ассоциируется с наиболее подходящей инструментальной средой: тексты - с текстовым процессором, таблицы - с электронной таблицей, графики - со средой деловой графики и т.д. К этим типовым средам могут быть добавлены также и разработанные самим программистом среды, а также библиотеки сред. Отметим, что интерактивное планирование позволяет находить оптимальные структуры и набор ресурсов для достижения поставленной цели.

В последние годы вырос интерес к распределенным системам - программным комплексам, составные части которых функционируют на разных компьютерах в сети, используя при взаимодействии технологии различного уровня, от непосредственного использования пакетов TCP/IP до технологий с высоким уровнем абстракции, таких, например, как CORBA. В этих системах обеспечены следующие возможности, невыполнимые при использовании традиционных технологий:

- масштабируемость, т.е. эффективное обслуживание различного числа клиентов одновременно;
- надежность создаваемых приложений, т.е. устойчивость не только к ошибкам пользователей, но и к сбоям в системе коммуникаций;
- непрерывная длительная работа (режим 24×7, т.е. 24 часа в течение 7 дней недели);
- высокий уровень безопасности системы, т.е. защиты и отслеживания, протоколирования информации на всех этапах функционирования;
- высокая скорость разработки приложений и простота их сопровождения и модификации (достаточен средний уровень программиста).

Технология CORBA создавалась некоммерческой организацией - содружеством разработчиков и пользователей программного обеспечения OMG как универсальная технология создания распределенных систем с использованием языков Java, C, Ada, Smalltalk, Delphi, Perl, Python и др. Клиентская часть может быть написана на любом языке программирования, поддерживающим CORBA.

Так как CORBA - стандартная инфраструктура разработки и использования различных платформ, ОС и приложений, то все спецификации CORBA являются полностью открытыми. CORBA реализует высокий уровень абстракции - все проблемы и описания взаимодействия с операционной системой или сетевыми средствами осуществляются на низком уровне и скрытно от прикладного программиста. Высокий уровень абстракции достигается за счет отображения инструкций на языке спецификаций - на конкретный язык программирования.

CORBA может передавать данные различных типов: структуры, объединения и др. Предусмотрена система описания и контроля типов. Для каждого языка используется свое отображение данных на языке спецификаций. CORBA поддерживает статический и динамический способ организации удаленных вызовов и имеет развитые средства получения информации о серверах. CORBA обладает высоким уровнем устойчивости к сбоям за счет большей изоляции клиентов и серверов, автоматического сохранения состояния объектов, более мощной и продуманной схемы управления транзакциями. Управление транзакциями берет на себя так называемый Сервис Управления Транзакциями CORBA (Object Transaction Service). CORBA обеспечивает высокий уровень безопасности. Предусмотрена идентификация пользователя, списки прав доступа к ресурсам, система аудита и многое другое. Интеграция CORBA и сети Интернет выполняется за счет использования протокола, построенного поверх TCP/IP, что позволяет использовать URL-имена в качестве имен для Службы Именования CORBA.

17. **CASE-технологии** (Computer-Aided System Engineering -автоматизированное проектирование информационных систем, или технологии, позволяющие автоматизировать основные этапы и процедуры жизненного цикла информационных систем: от анализа исходного состояния и целей - до проектирования интерфейсов, привычных проектировщику, пользователю и основных процедур функционирования системы; чем больше этапов и процедур автоматизируется, тем лучше и быстрее получается информационная система, тем шире ее приложения).

Пример. Технология STRADIS (STRategic Architecture for the Deployment of Information Systems - стратегическая архитектура для развертывания информационных систем) определяет и поддерживает основные этапы жизненного цикла системы: цели, их приоритеты, требования к ресурсам, распределению работ, составу и содержанию проектной документации, методика выполнения процедур проектирования и программирования, тестирования и управления. Включает в себя следующий инструментарий для этого: графический редактор (графическая среда), СУБД, средства описания сценариев диалога с системой, выходных документов и др.

18. **Технологии** и системы **компьютерной алгебры**, системы символьных преобразований, аналитических вычислений (системы, позволяющие производить автоматические преобразования формул и алгебраических выражений, в частности, приведение подобных членов в алгебраическом выражении, нахождение первообразной заданной аналитически функции, ее дифференцирование и т.д.).

Пример. Система Reduce для формульных преобразований, которая позволяет как находить эквивалентные алгебраические выражения, так и вычислять их численные значения (в том числе и комплекснозначные), суммировать конечные и бесконечные ряды (сумма - как функция!), производить алгебраические операции с полиномами, матрицами, интегрировать и дифференцировать.

19. **Нечеткие технологии** (технологии обработки данных и вывода знаний, принятия решений на основе описания систем аппаратом нечетких множеств и нечеткой логики).

Пример. Медицинский диагноз часто основан на нечетких, неопределенных четко связях симптомов и болезней, их нечеткой зависимости, поэтому для компьютерной постановки диагноза, построения экспертной системы постановки диагноза эта технология особенно эффективна, так как позволяет делать нечеткие выводы, которые затем могут быть проверены. Проблемы дактилоскопии также могут быть решены эффективно с помощью нечетких систем распознавания отпечатков.

Все новые *информационные технологии*, так или иначе, используют методы и проблематику *искусственного интеллекта, инженерии знаний*, часто переплетаются и интегрируются.

Проблематику **искусственного интеллекта** составляют знания, информация о данной области, которые пока объективно непонятны, неточны, не формализуемы, не структурируемы, не актуализируемы доступными средствами (и могут стать таковыми в процессе функционирования системы, приобретения знаний).

Инженерия знаний - наука, изучающая проблемы выявления, структурирования, формализации и актуализации знаний для разработки различного типа интеллектуальных систем, технологий.

Все новые *информационные технологий* должны обеспечивать целенаправленность, информативность, адекватность, точность, полноту, воспринимаемость и структурированность сообщений, а также гибкость, комфортность, своевременность и простоту их актуализации во времени, в пространстве и информационно.

Все *информационные технологии* - основа многих других технологий, а также способ актуализации информации, основа мышления.

Основные тенденции развития новых *информационных технологий*, независимо от сферы их использования:

- возрастание роли и активности (актуальности) информационного ресурса, т.е. качество и оперативность принимаемых интеллектуальных решений в обществе во все большей степени зависит от содержания, точности и своевременности получаемой информации, ее пространственно-временных характеристик;
- развитие способности к активному техническому, программному и технологическому взаимодействию (стандартизации и совместимости таких взаимодействий), т.е. появление более совершенных стандартов взаимодействия, все чаще - уже на уровне проектных работ, на уровне разработки спецификаций;
- изменение структуры инфологических и структурных взаимодействий, ликвидация промежуточных звеньев (непосредственность), т.е. устранение этапов и функций посредников информационного обмена и услуг, ликвидация промежуточных функций внутри компаний и между ними, более широкое распространение, упрощение доступа, снижение цен и т.д.;
- глобализация или использование пространственных, временных и организационных возможностей и емкости информационного рынка (практически беспредельного);
- конвергенция или формирование рынка новых *информационных технологий*, состоящего из основных сегментов - частное потребление (развлечения, бытовые услуги и т.п.), обеспечение бизнеса (производство, продажа, маркетинг и т.п.), интеллектуальная профессиональная работа (автоформализация профессиональных знаний и др.).

Пример. В 1990 г. около 40% интеллектуальных работников в США использовали на своих рабочих местах новые *информационные технологии*, в частности, концерн Microsoft инвестирует в новые медиапроекты до 20% своего научного бюджета, выпуская энциклопедии и справочники на CD, работая параллельно с нормальным телевидением в сети интернет, открывая в Интернете свои мультимедийные журналы, например, Slate (<http://www.slate.com>) и др. Выполнение японской программы создания компьютеров пятого поколения сдерживается тем, что новая архитектура программного обеспечения пока не сочетается с существующими центрами *искусственного интеллекта*, новые протоколы не могут быть использованы в старых системах связи, а новые машинные языки не подходят для старых систем и т.д. В банковской сфере уменьшается роль мелких банков, так как на внедрение *информационных технологий* требуются значительные ресурсы. Имеющие доступ к терминалам общего пользования ("электронным киоскам") заказывают товары и получают электронные купоны. Примером глобальной программы является программа ESPRIT (Европейская стратегическая программа исследований в области *информационных технологий*). Примерами аппаратно-программной конвергенции могут быть принтеры с функциями ксерокса и факса.

Задачи и упражнения

1. Выбрать одну-две новые технологии и построить для них примеры использования, указать достоинства и недостатки.
2. Построить несколько макетов (логических моделей) *БД* социально-экономического направления (например, пенсионного фонда). Описать структуру записей, атрибуты

- полей базы, сформулировать запросы. Осуществить операции (поиска, сортировки, модификации) с базой данных. Оценить объем информации в *БД*.
3. Построить несколько макетов (логических моделей) баз знаний по социально-экономической предметной области. Построить несколько макетов (логических моделей) экспертной системы по социально-экономической проблеме. Привести примеры проблем, которые можно решить эффективно с помощью экспертной системы. Осуществить какие-либо корректные операции с построенными базами знаний на логическом уровне. Построить компьютерные модели баз знаний по реальным социально-экономическим системам (процессам) и рассмотреть их эксплуатационные ситуации и области приложения. Оценить объем информации (качественно и количественно) в построенной (или другой) базе знаний. Осуществить постановку некоторых задач, которые можно решать с помощью некоторой базы знаний и (или) экспертной налоговой системы. Выполнить операции логического вывода из базы знаний, возможно, упростив для этого структуру базы знаний.
 4. Построить несколько сценариев проведения *телеконференций* по различным налоговым проблемам. Описать работу организатора (модератора) и пользователя *телеконференции*. Оценить объем информации в сеансе *телеконференции*. Осуществить постановку некоторых задач, которые можно решать с помощью *телеконференции*. Описать технологию решения этих задач. Привести примеры социально-экономических последствий проведения *телеконференций* и использования *электронной почты*. Оценить эти последствия. Привести примеры *телеконференции* по вашей специальности.
 5. Описать работу некоторой гипотетической *виртуальной корпорации* с участием специалистов по Вашей будущей специальности.
 6. Описать спецификации и процедуру *реинжиниринга* системы обучения студентов по Вашей будущей специальности.