

Курс лекций по дисциплине
Прикладной искусственный интеллект

Искусственный интеллект - основа новых информационных технологий

Введение в искусственный интеллект

Искусственный интеллект (ИИ) как наука существует около полувека. Первой интеллектуальной системой считается программа "Логик-Теоретик", предназначенная для доказательства теорем и исчисления высказываний. Ее работа впервые была продемонстрирована 9 августа 1956 г. В создании программы участвовали такие известные ученые, как А. Ньюэлл, А. Тьюринг, К. Шеннон, Дж. Лоу, Г. Саймон и др. За прошедшее с тех пор время в области ИИ разработано великое множество компьютерных систем, которые принято называть интеллектуальными. Области их применения охватывают практически все сферы человеческой деятельности, связанные с обработкой информации.

На сегодняшний день не существует единого определения, которое однозначно описывает эту научную область. Среди многих точек зрения на нее доминируют следующие три.

Согласно первой точке зрения, исследования в области ИИ относятся к фундаментальным, в процессе которых разрабатываются новые модели и методы решения задач, традиционно считавшихся интеллектуальными, не поддававшихся ранее формализации и автоматизации.

Согласно второй точке зрения это направление связано с новыми идеями решения задач на ЭВМ, с разработкой новых технологий программирования и с переходом к компьютерам не фон-неймановской архитектуры.

Третья точка зрения, наиболее прагматическая, основана на том, что в результате исследований, проводимых в области ИИ, появляется множество прикладных систем, способных решать задачи, для которых ранее создаваемые системы были непригодны. По последней трактовке ИИ является экспериментальной научной дисциплиной, в которой роль эксперимента заключается в проверке и уточнении интеллектуальных систем, представляющих собой аппаратно-программные информационные комплексы.

Основные направления исследований в области искусственного интеллекта

Интеллектуальные информационные системы проникают во все сферы нашей жизни, поэтому трудно провести строгую классификацию направлений, по которым ведутся активные и многочисленные исследования в области ИИ. Рассмотрим кратко некоторые из них.

Разработка интеллектуальных информационных систем или систем, основанных на знаниях. Это одно из главных направлений ИИ. Основной целью построения таких систем являются выявление, исследование и применение знаний высококвалифицированных экспертов для решения сложных задач, возникающих на практике. При построении систем, основанных на знаниях (СОЗ), используются знания, накопленные экспертами в виде конкретных правил решения тех или иных задач. Это направление преследует цель имитации человеческого искусства анализа неструктурированных и слабоструктурированных проблем. В данной области исследований осуществляется разработка моделей представления, извлечения и структурирования знаний, а также изучаются проблемы создания баз знаний (БЗ), образующих ядро СОЗ. Частным случаем СОЗ являются экспертные системы (ЭС).

Разработка естественно-языковых интерфейсов и машинный перевод. Проблемы компьютерной лингвистики и машинного перевода разрабатываются в ИИ с 1950-х гг. Системы машинного перевода с одного естественного языка на другой обеспечивают быстроту и систематичность доступа к информации, оперативность и единообразие перевода больших потоков, как правило, научно-технических текстов. Системы машинного перевода строятся как интеллектуальные системы, поскольку в их основе лежат БЗ в определенной предметной области и сложные модели, обеспечивающие дополнительную трансляцию "исходный язык оригинала - язык смысла - язык перевода". Они базируются на структурно-логическом подходе, включающем последовательный анализ и синтез естественно-языковых сообщений. Кроме того, в них осуществляется ассоциативный поиск аналогичных фрагментов текста и их переводов в специальных базах данных (БД). Данное направление охватывает также исследования методов и разработку систем, обеспечивающих реализацию процесса общения человека с компьютером на естественном языке (так называемые системы ЕЯ-общения).

Генерация и распознавание речи. Системы речевого общения создаются в целях повышения скорости ввода информации в ЭВМ, разгрузки зрения и рук, а также для реализации речевого общения на значительном расстоянии. В таких системах под текстом понимают фонемный текст (как слышится).

Обработка визуальной информации. В этом научном направлении решаются задачи обработки, анализа и синтеза изображений. Задача обработки изображений связана с трансформированием графических образов, результатом которого являются новые изображения. В задаче анализа исходные изображения преобразуются в данные другого типа, например в текстовые описания. При синтезе изображений на вход системы поступает алгоритм построения изображения, а выходными данными являются графические объекты (системы машинной графики).

Обучение и самообучение. Эта актуальная область ИИ включает модели, методы и алгоритмы, ориентированные на автоматическое накопление и формирование знаний с использованием процедур анализа и обобщения данных. К данному направлению относятся не так давно появившиеся системы добычи данных (Data-mining) и системы поиска закономерностей в компьютерных базах данных (Knowledge Discovery).

Распознавание образов. Это одно из самых ранних направлений ИИ, в котором распознавание объектов осуществляется на основании применения специального математического аппарата, обеспечивающего отнесение объектов к классам, а классы описываются совокупностями определенных значений признаков.

Игры и машинное творчество. Машинное творчество охватывает сочинение компьютерной музыки, стихов, интеллектуальные системы для изобретения новых объектов. Создание интеллектуальных компьютерных игр является одним из самых развитых коммерческих направлений в сфере разработки программного обеспечения. Кроме того, компьютерные игры предоставляют мощный арсенал разнообразных средств, используемых для обучения.

Программное обеспечение систем ИИ. Инструментальные средства для разработки интеллектуальных систем включают:

- специальные языки программирования, ориентированные на обработку символьной информации (LISP, SMALLTALK, РЕФАЛ),
- языки логического программирования (PROLOG), языки представления знаний (OPS 5, KRL, FRL),
- интегрированные программные среды, содержащие арсенал инструментальных средств для создания систем ИИ (KE, ARTS, GURU, G2),
- оболочки экспертных систем (BUILD, EMYCIN, EXSYS Professional, ЭКСПЕРТ), которые позволяют создавать прикладные ЭС, не прибегая к программированию.

Новые архитектуры компьютеров. Это направление связано с созданием компьютеров не фон-неймановской архитектуры, ориентированных на обработку символьной информации. Известны удачные промышленные решения параллельных и векторных компьютеров, однако в настоящее время они имеют весьма высокую стоимость, а также недостаточную совместимость с существующими вычислительными средствами.

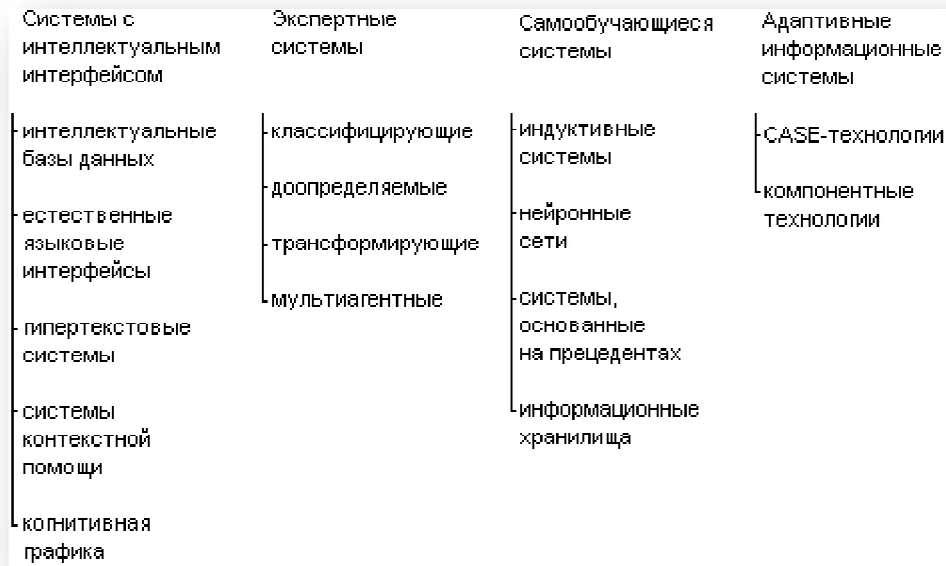
Интеллектуальные роботы. Создание интеллектуальных роботов составляет конечную цель робототехники. В настоящее время в основном используются программируемые манипуляторы с жесткой схемой управления, названные роботами первого поколения. Несмотря на очевидные успехи отдельных разработок, эра интеллектуальных автономных роботов пока не наступила. Основными сдерживающими факторами в разработке автономных роботов являются нерешенные проблемы в области интерпретации знаний, машинного зрения, адекватного хранения и обработки трехмерной визуальной информации

Классификация интеллектуальных информационных систем

Интеллектуальная информационная система (ИИС) основана на концепции использования базы знаний для генерации алгоритмов решения прикладных задач различных классов в зависимости от конкретных информационных потребностей пользователей.

Для ИИС характерны следующие признаки:

- развитые коммуникативные способности;
- умение решать сложные плохо формализуемые задачи;
- способность к самообучению;
- адаптивность.



Каждому из перечисленных признаков условно соответствует свой класс ИИС. Различные системы могут обладать одним или несколькими признаками интеллектуальности с различной степенью проявления.

Средства ИИ могут использоваться для реализации различных функций, выполняемых ИИС. На рисунке приведена классификация ИИС, признаками которой являются следующие интеллектуальные функции:

- коммуникативные способности - способ взаимодействия конечного пользователя с системой;
- решение сложных плохо формализуемых задач, которые требуют построения оригинального алгоритма решения в зависимости от конкретной ситуации, характеризующейся неопределенностью и динамичностью исходных данных и знаний;
- способность к самообучению - умение системы автоматически извлекать знания из накопленного опыта и применять их для решения задач;
- адаптивность - способность системы к развитию в соответствии с объективными изменениями области знаний.

Системы с интеллектуальным интерфейсом

Применение ИИ для усиления коммуникативных способностей информационных систем привело к появлению систем с интеллектуальным интерфейсом, среди которых можно выделить следующие типы.

1. **Интеллектуальные базы данных.** Позволяют в отличие от традиционных БД обеспечивать выборку необходимой информации, не

присутствующей в явном виде, а выводимой из совокупности хранимых данных.

2. **Естественно-языковой интерфейс.** Применяется для доступа к интеллектуальным базам данных, контекстного поиска документальной текстовой информации, голосового ввода команд в системах управления, машинного перевода с иностранных языков. Для реализации ЕЯ-интерфейса необходимо решить проблемы морфологического, синтаксического и семантического анализа, а также задачу синтеза высказываний на естественном языке. При морфологическом анализе осуществляются распознавание и проверка правильности написания слов в словаре. Синтаксический контроль предполагает разложение входных сообщений на отдельные компоненты, проверку соответствия грамматическим правилам внутреннего представления знаний и выявление недостающих частей. Семантический анализ обеспечивает установление смысловой правильности синтаксических конструкций. В отличие от анализа синтез высказываний заключается в преобразовании цифрового представления информации в представление на естественном языке.
3. **Гипертекстовые системы.** Используются для реализации поиска по ключевым словам в базах данных с текстовой информацией. Для более полного отражения различных смысловых отношений терминов требуется сложная семантическая организация ключевых слов. Решение этих задач осуществляется с помощью интеллектуальных гипертекстовых систем, в которых механизм поиска сначала работает с базой знаний ключевых слов, а затем - с самим текстом. Аналогичным образом проводится поиск мультимедийной информации, включающей кроме текста графическую информацию, аудио- и видеообразы.
4. **Системы контекстной помощи.** Относятся к классу систем распространения знаний. Такие системы являются, как правило, приложениями к документации. Системы контекстной помощи - частный случай гипертекстовых и ЕЯ-систем. В них пользователь описывает проблему, а система на основе дополнительного диалога конкретизирует ее и выполняет поиск относящихся к ситуации рекомендаций. В обычных гипертекстовых системах, наоборот, компьютерные приложения навязывают пользователю схему поиска требуемой информации.
5. **Системы когнитивной графики.** Ориентированы на общение с пользователем ИИС посредством графических образов, которые генерируются в соответствии с изменениями параметров моделируемых или наблюдаемых процессов. Когнитивная графика позволяет в наглядном и выразительном виде представить множество параметров, характеризующих изучаемое явление, освобождает пользователя от анализа тривиальных ситуаций, способствует быстрому освоению программных средств и повышению

конкурентоспособности разрабатываемых ИИС. Применение когнитивной графики особенно актуально в системах мониторинга и оперативного управления, в обучающих и тренажерных системах, в оперативных системах принятия решений, работающих в режиме реального времени.

Экспертные системы

Экспертные системы как самостоятельное направление в искусственном интеллекте сформировалось в конце 1970-х годов. История ЭС началась с сообщения японского комитета по разработке ЭВМ пятого поколения, в котором основное внимание уделялось развитию "интеллектуальных способностей" компьютеров с тем, чтобы они могли оперировать не только данными, но и знаниями, как это делают специалисты (эксперты) при выработке умозаключений.

Область исследования ЭС называют "инженерией знаний". Этот термин был введен Е. Фейгенбаумом и в его трактовке означает "привнесение принципов и инструментария из области искусственного интеллекта в решение трудных прикладных проблем, требующих знаний экспертов". Другими словами, ЭС применяются для решения неформализованных проблем, к которым относят задачи, обладающие одной (или несколькими) из следующих характеристик:

- задачи не могут быть представлены в числовой форме;
- исходные данные и знания о предметной области обладают неоднозначностью, неточностью, противоречивостью;
- цели нельзя выразить с помощью четко определенной целевой функции;
- не существует однозначного алгоритмического решения задачи;
- алгоритмическое решение существует, но его нельзя использовать по причине большой размерности пространства решений и ограничений на ресурсы (времени, памяти).

Главное отличие ЭС и систем искусственного интеллекта от систем обработки данных состоит в том, что в них используется символьный, а не числовой способ представления данных, а в качестве методов обработки информации применяются процедуры логического вывода и эвристического поиска решений.

ЭС охватывают самые разные предметные области, среди которых лидируют бизнес, производство, медицина, проектирование и системы управления.

Во многих случаях ЭС являются инструментом, усиливающим интеллектуальные способности эксперта. Кроме того, ЭС может выступать в роли:

- консультанта для неопытных или непрофессиональных пользователей;
- ассистента эксперта-человека в процессах анализа вариантов решений;
- партнера эксперта в процессе решения задач, требующих привлечения знаний из разных предметных областей.

Для классификации ЭС используются следующие признаки:

- способ формирования решения;
- способ учета временного признака;
- вид используемых данных и знаний;
- число используемых источников знаний.

По способу формирования решения ЭС можно разделить на анализирующие и синтезирующие. В системах первого типа осуществляется выбор решения из множества известных решений на основе анализа знаний, в системах второго типа решение синтезируется из отдельных фрагментов знаний.

В зависимости от способа учета временного признака ЭС делят на статические и динамические. Статические ЭС предназначены для решения задач с неизменяемыми в процессе решения данными и знаниями, а динамические ЭС допускают такие изменения.

По видам используемых данных и знаний различают ЭС с детерминированными и неопределенными знаниями. Под неопределенностью знаний и данных понимаются их неполнота, ненадежность, нечеткость.

ЭС могут создаваться с использованием одного или нескольких источников знаний.

В соответствии с перечисленными признаками можно выделить четыре основных класса ЭС (см. рис.):

- классифицирующие,
- доопределяющие,
- трансформирующие
- мультиагентные.

Самообучающиеся системы

Самообучающиеся интеллектуальные системы основаны на методах автоматической классификации ситуаций из реальной практики, или на методах обучения на примерах. Примеры реальных ситуаций составляют так называемую обучающую выборку, которая формируется в течение определенного исторического периода. Элементы обучающей выборки описываются множеством классификационных признаков.

Стратегия "обучения с учителем" предполагает задание специалистом для каждого примера значений признаков, показывающих его принадлежность к определенному классу ситуаций. При обучении "без учителя" система должна самостоятельно выделять классы ситуаций по степени близости значений классификационных признаков.

В процессе обучения проводится автоматическое построение обобщающих правил или функций, описывающих принадлежность ситуаций к классам, которыми система впоследствии будет пользоваться при интерпретации незнакомых ситуаций. Из обобщающих правил, в свою очередь, автоматически формируется база знаний, которая периодически корректируется по мере накопления информации об анализируемых ситуациях.

Построенные в соответствии с этими принципами самообучающиеся системы имеют следующие недостатки:

- относительно низкую адекватность баз знаний возникающим реальным проблемам из-за неполноты и/или зашумленности обучающей выборки;
- низкую степень объяснимости полученных результатов;
- поверхностное описание проблемной области и узкую направленность применения из-за ограничений в размерности признакового пространства.

Индуктивные системы позволяют обобщать примеры на основе принципа индукции "от частного к общему". Процедура обобщения сводится к классификации примеров по значимым признакам. Алгоритм классификации примеров включает следующие основные шаги.

1. Выбор классификационного признака из множества заданных.
2. Разбиение множества примеров на подмножества по значению выбранного признака.
3. Проверка принадлежности каждого подмножества примеров одному из классов.

4. Проверка окончания процесса классификации. Если какое-то подмножество примеров принадлежит одному подклассу, т.е. у всех примеров этого подмножества совпадает значение классификационного признака, то процесс классификации заканчивается.
5. Для подмножеств примеров с несовпадающими значениями классификационных признаков процесс распознавания продолжается, начиная с первого шага. При этом каждое подмножество примеров становится классифицируемым множеством.

Нейронные сети

Представляют собой классический пример технологии, основанной на примерах. Нейронные сети - обобщенное название группы математических алгоритмов, обладающих способностью обучаться на примерах, "узнавая" впоследствии черты встреченных образцов и ситуаций. Благодаря этой способности нейронные сети используются при решении задач обработки сигналов и изображений, распознавания образов, а также для прогнозирования.

Нейронная сеть - это кибернетическая модель нервной системы, которая представляет собой совокупность большого числа сравнительно простых элементов - нейронов, топология соединения которых зависит от типа сети. Чтобы создать нейронную сеть для решения какой-либо конкретной задачи, следует выбрать способ соединения нейронов друг с другом и подобрать значения параметров межнейронных соединений.

В системах, основанных на прецедентах, БЗ содержит описания конкретных ситуаций (прецеденты). Поиск решения осуществляется на основе аналогий и включает следующие этапы:

- получение информации о текущей проблеме;
- сопоставление полученной информации со значениями признаков прецедентов из базы знаний;
- выбор прецедента из базы знаний, наиболее близкого к рассматриваемой проблеме;
- адаптация выбранного прецедента к текущей проблеме;
- проверка корректности каждого полученного решения;
- занесение детальной информации о полученном решении в БЗ.

Прецеденты описываются множеством признаков, по которым строятся индексы быстрого поиска. Однако в системах, основанных на прецедентах, в отличие от индуктивных систем допускается нечеткий поиск с получением множества допустимых альтернатив, каждая из которых оценивается

некоторым коэффициентом уверенности. Наиболее эффективные решения адаптируются к реальным ситуациям с помощью специальных алгоритмов.

Системы, основанные на прецедентах, применяются для распространения знаний и в системах контекстной помощи.

Информационные хранилища

ИХ отличаются от интеллектуальных баз данных, тем, что представляют собой хранилища значимой информации, регулярно извлекаемой из оперативных баз данных. Хранилище данных - это предметно-ориентированное, интегрированное, привязанное ко времени, неизменяемое собрание данных, применяемых для поддержки процессов принятия управленческих решений. Предметная ориентация означает, что данные объединены в категории и хранятся в соответствии с теми областями, которые они описывают, а не с приложениями, которые их используют. В хранилище данные интегрируются в целях удовлетворения требований предприятия в целом, а не отдельной функции бизнеса. Привязанность данных ко времени выражает их "историчность", т.е. атрибут времени всегда явно присутствует в структурах хранилища данных. Неизменяемость означает, что, попав однажды в хранилище, данные уже не изменяются в отличие от оперативных систем, где данные присутствуют только в последней версии, поэтому постоянно меняются.

Технологии извлечения знаний из хранилищ данных основаны на методах статистического анализа и моделирования, ориентированных на поиск моделей и отношений, скрытых в совокупности данных. Эти модели могут в дальнейшем использоваться для оптимизации деятельности предприятия или фирмы.

Для извлечения значимой информации из хранилищ данных имеются специальные методы (OLAP-анализа, Data Mining или Knowledge Discovery), основанные на применении методов мате-матической статистики, нейронных сетей, индуктивных методов построения деревьев решений и др.

Технология OLAP (On-line Analytical Processing - оперативный анализ данных) предоставляет пользователю средства для формирования и проверки гипотез о свойствах данных или отношениях между ними на основе разнообразных запросов к базе данных. Они применяются на ранних стадиях процесса извлечения знаний, помогая аналитику сфокусировать внимание на важных переменных. Средства Data Mining отличаются от OLAP тем, что кроме проверки предполагаемых зависимостей они способны самостоятельно (без участия пользователя) генерировать гипотезы о закономерностях, существующих в данных, и строить модели, позволяющие количественно

оценить степень взаимного влияния исследуемых факторов на основе имеющейся информации.

Адаптивные информационные системы

Потребность в адаптивных информационных системах возникает в тех случаях, когда поддерживаемые ими проблемные области постоянно развиваются. В связи с этим адаптивные системы должны удовлетворять ряду специфических требований, а именно:

- адекватно отражать знания проблемной области в каждый момент времени;
- быть пригодными для легкой и быстрой реконструкции при изменении проблемной среды.

Адаптивные свойства информационных систем обеспечиваются за счет интеллектуализации их архитектуры. Ядром таких систем является постоянно развиваемая модель проблемной области, поддерживаемая в специальной базе знаний - *репозитории*. Ядро системы управляет процессами генерации или переконфигурирования программного обеспечения.

В процессе разработки адаптивных информационных систем применяется оригинальное или типовое проектирование. Оригинальное проектирование предполагает разработку информационной системы с "чистого листа" на основе сформулированных требований. Реализация этого подхода основана на использовании систем автоматизированного проектирования, или **CASE-технологий** (Designer2000, SilverRun, Natural Light Storm и др.).

При типовом проектировании осуществляется адаптация типовых разработок к особенностям проблемной области. Для реализации этого подхода применяются инструментальные средства **компонентного (сборочного) проектирования информационных систем** (R/3, BAAN IV, Prodis и др.). Главное отличие подходов состоит в том, что при использовании CASE-технологии на основе *репозитория* при изменении проблемной области каждый раз выполняется генерация программного обеспечения, а при использовании сборочной технологии - конфигурирование программ и только в редких случаях - их переработка.

ЭТАПЫ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ

Разработка программных комплексов экспертных систем как за рубежом, так и в нашей стране находится на уровне скорее искусства, чем науки. Это связано с тем, что долгое время системы искусственного интеллекта внедрялись в основном во время фазы проектирования, а чаще всего разрабатывалось несколько прототипных версий программ, прежде чем был получен конечный продукт. Такой подход действует хорошо в исследовательских условиях, однако в коммерческих условиях он является слишком дорогим, чтобы оправдать коммерчески жизненный продукт.

Процесс разработки промышленной экспертной системы, опираясь на традиционные технологии [4,8,10], можно разделить на шесть более или менее независимых этапов (рис 16.7), практически не зависящих от предметной области.

Последовательность этапов дана для общего представления о создании идеального проекта. Конечно, последовательность эта не вполне фиксированная. В действительности каждый последующий этап разработки ЭС приносит новые идеи, которые могут повлиять на предыдущие решения и даже привести к их переработке. Именно поэтому многие специалисты по информатике весьма критично относятся к методологии экспертных систем. Они считают, что расходы на разработку таких систем очень большие, время разработки слишком длительное, а полученные в результате программы ложатся тяжелым бременем на вычислительные ресурсы.

В целом за разработку экспертных систем целесообразно браться организации, где накоплен опыт по автоматизации рутинных процедур обработки информации, например:

- информационный поиск;
- сложные расчеты;
- графика;
- обработка текстов.

Решение таких задач, во-первых, подготавливает высококвалифицированных специалистов по информатике, необходимых для создания интеллектуальных систем, во-вторых, позволяет отделить от экспертных систем не экспертные задачи.

Этап 1: Выбор подходящей проблемы

Этот этап включает деятельность, предшествующую решению начать разрабатывать конкретную ЭС. Он включает:

- определение проблемной области и задачи;
- нахождение эксперта, желающего сотрудничать при решении проблемы, и назначение коллектива разработчиков;
- определение предварительного подхода к решению проблемы;
- анализ расходов и прибыли от разработки;

- подготовку подробного плана разработки.

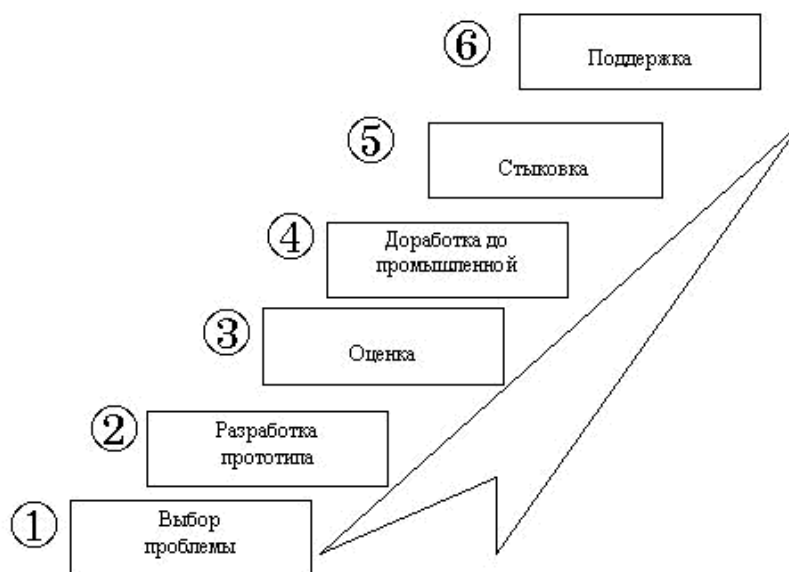


Рис. Этапы разработки ЭС

Правильный выбор проблемы представляет, наверное, самую критическую часть разработки в целом. Если выбрать неподходящую проблему, можно очень быстро увязнуть в "болоте" проектирования задач, которые никто не знает, как решать. Неподходящая проблема может также привести к созданию экспертной системы, которая стоит намного больше, чем экономит. Дело будет обстоять еще хуже, если разработать систему, которая работает, но не приемлема для пользователей. Даже если разработка выполняется самой организацией для собственных целей, эта фаза является подходящим моментом для получения рекомендаций извне, чтобы гарантировать удачно выбранный и осуществимый с технической точки зрения первоначальный проект.

При выборе области применения следует учитывать, что если знание, необходимое для решения задач, постоянное, четко формулируемое и связано с вычислительной обработкой, то обычные алгоритмические программы, по всей вероятности, будут самым целесообразным способом решения проблем в этой области,

Экспертная система ни в коем случае не устранил потребность в реляционных базах данных, статистическом программном обеспечении, электронных таблицах и системах текстовой обработки. Но если результативность задачи зависит от знания, которое является субъективным, изменяющимся, символьным или вытекающим частично из соображений здравого смысла, тогда область может обоснованно выступать претендентом на экспертную систему.

Приведем некоторые факты, свидетельствующие о необходимости разработки и внедрения экспертных систем:

- нехватка специалистов, расходующих значительное время для оказания помощи другим;

- потребность в многочисленном коллективе специалистов, поскольку ни один из них не обладает достаточным знанием;
- сниженная производительность, поскольку задача требует полного анализа сложное набора условий, а обычный специалист не в состоянии просмотреть (за отведенное время) все эти условия;
- большое расхождение между решениями самых хороших и самых плохих исполнителей;
- наличие конкурентов, имеющих преимущество в том, что они лучше справляются с поставленной задачей.

Подходящие задачи имеют следующие характеристики:

- являются узкоспециализированными;
- не зависят в значительной степени от общечеловеческих знаний или соображении здравого смысла;
- не являются для эксперта ни слишком легкими, ни слишком сложными (время, необходимое эксперту для решения проблемы, может составлять от трех часов до трех недель);
- условия исполнения задачи определяются самим пользователем системы;
- имеет результаты, которые можно оценить.

Обычно экспертные системы разрабатываются путем получения специфических знаний от эксперта и ввода их в систему. Некоторые системы могут содержать стратегии одного индивида. Следовательно, найти подходящего эксперта - это ключевой шаг в создании экспертных систем.

В процессе разработки и последующего расширения системы инженер по знаниям и эксперт обычно работают вместе. Инженер по знаниям помогает эксперту структурировать знания, определять и формализовать понятия и правила, необходимые для решения проблемы.

Во время первоначальных бесед они решают, будет ли их сотрудничество успешным. Это немаловажно, поскольку обе стороны будут работать вместе по меньшей мере в течение одного года. Кроме них в коллектив разработчиков целесообразно включить потенциальных пользователей и профессиональных программистов.

Предварительный подход к программной реализации задачи определяется исходя из характеристик задачи и ресурсов, выделенных на ее решение. Инженер по знаниям выдвигает обычно несколько вариантов, связанных с использованием имеющихся на рынке программных средств. Окончательный выбор возможен лишь на этапе разработки прототипа.

После того как задача определена, необходимо подсчитать расходы и прибыли от разработки экспертной системы. В расходы включаются затраты на оплату труда коллектива разработчиков. В дополнительные расходы приобретаемого программного инструментария, с помощью которого разрабатывается экспертная система.

Прибыль возможна за счет снижения цены продукции, повышения производительности труда, расширения номенклатуры продукции или услуг

или даже разработки новых видов продукции или услуг в этой области. Соответствующие расходы и прибыли от системы определяются относительно времени, в течение которого возвращаются средства, вложенные в разработку. На современном этапе большая часть фирм, развивающих крупные экспертные системы, предпочли разрабатывать дорогостоящие проекты, приносящие значительные прибыли.

Наметились тенденции разработки менее дорогостоящих систем, хотя и с более длительным сроком возвращаемости вложенных в них средств, так как программные средства разработки экспертных систем непрерывно совершенствуются.

После того как инженер по знаниям убедился, что:

- данная задача может быть решена с помощью экспертной системы;
 - экспертную систему можно создать предлагаемыми на рынке средствами;
 - имеется подходящий эксперт;
 - предложенные критерии производительности являются разумными;
 - затраты и срок их возвращаемости приемлемы для заказчика;
- он составляет план разработки. План определяет шаги процесса разработки и необходимые затраты, а также ожидаемые результаты.

Этап 2: Разработка прототипной системы

Понятие прототипной системы

Прототипная система является усеченной версией экспертной системы, спроектированной для проверки правильности кодирования фактов, связей и стратегий рассуждения эксперта. Она также дает возможность инженеру по знаниям привлечь эксперта к активному участию в разработке экспертной системы и, следовательно, к принятию им обязательства приложить все усилия для создания системы в полном объеме.

Объем прототипа - несколько десятков правил, фреймов или примеров. На рис. 16.8 изображены шесть стадий разработки прототипа и минимальный коллектив разработчиков, занятых на каждой из стадий (пять стадий заимствованы из [10]). Приведем краткую характеристику каждой из стадий, хотя эта схема представляет грубое приближение к сложному итеративному процессу.

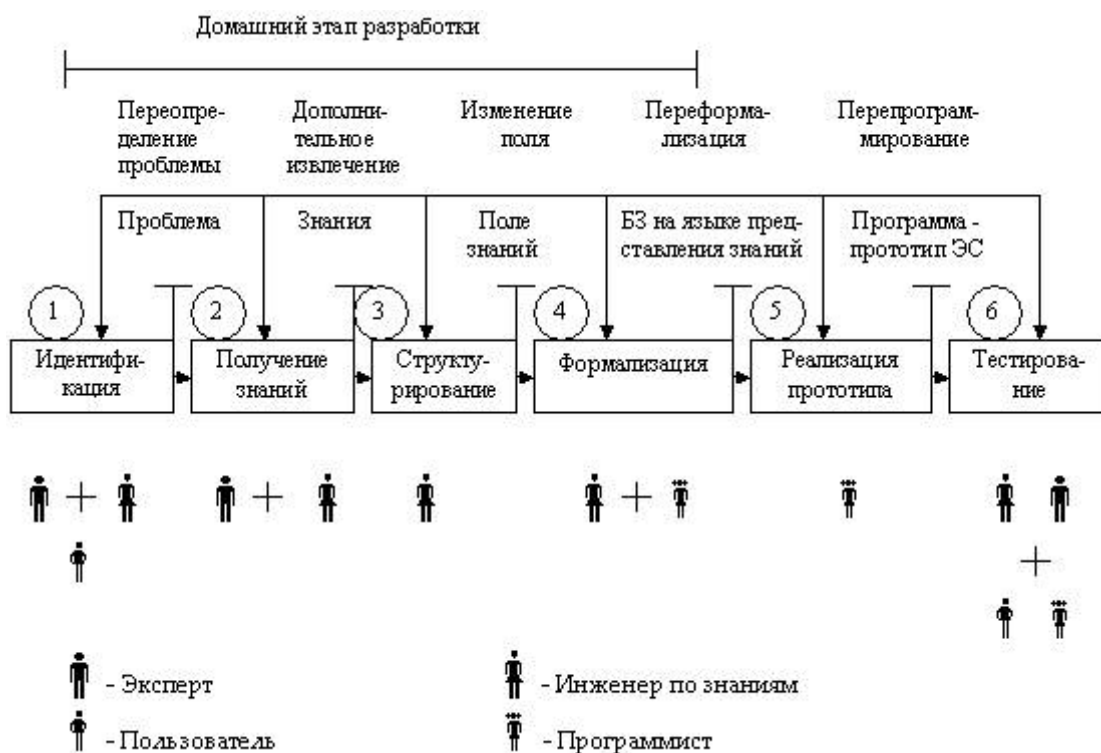


Рис. Стадии разработки прототипа ЭС

Хотя любое теоретическое разделение бывает часто условным, осознание коллективом разработчиков четких задач каждой стадии представляется целесообразным. Роли разработчиков (эксперт, программист, пользователь и аналитик) являются постоянными на протяжении всей разработки. Совмещение ролей нежелательно.

Сроки приведены условно, так как зависят от квалификации специалистов и особенностей задачи.

Идентификация проблемы

Уточняется задача, планируется ход разработки прототипа экспертной системы, определяются:

- необходимые ресурсы (время, люди, ЭВМ и т.д.);
- источники знаний (книги, дополнительные эксперты, методики);
- имеющиеся аналогичные экспертные системы;
- цели (распространение опыта, автоматизация рутинных действий и др.);
- классы решаемых задач и т.д.

Идентификация проблемы - знакомство и обучение коллектива разработчиков, а также создание неформальной формулировки проблемы.

Средняя продолжительность 1 - 2 недели.

Извлечение знаний

Происходит перенос компетентности экспертов на инженеров по знаниям с использованием различных методов:

- анализ текстов;
- диалоги;
- экспертные игры;
- лекции;

- дискуссии;
- интервью;
- наблюдение и другие.

Извлечение знаний - получение инженером по знаниям наиболее полного представления о предметной области и способах принятия решения в ней.

Средняя продолжительность 1 -3 месяца.

Структурирование или концептуализация знаний

Выявляется структура полученных знаний о предметной области, т.е. определяются:

- терминология;
- список основных понятий и их атрибутов;
- отношения между понятиями;
- структура входной и выходной информации;
- стратегия принятия решений;
- ограничения стратегий и т.д.

Концептуализация знаний - разработка неформального описания знаний о предметной области в виде графа, таблицы, диаграммы или текста, которое отражает основные концепции и взаимосвязи между понятиями предметной области.

Такое описание называется *полем знаний*. Средняя продолжительность этапа 2 - 4 недели.

Формализация.

Строится формализованное представление концепций предметной области на основе выбранного языка представления знаний (ЯПЗ). Традиционно на этом этапе используются:

- логические методы (исчисления предикатов I порядка и др.);
- продукционные модели (с прямым и обратным выводом);
- семантические сети;
- фреймы;
- объектно-ориентированные языки, основанные на иерархии классов, объектов и др.

Формализация знаний - разработка базы знаний на языке, который, с одной стороны, соответствует структуре поля знаний, а с другой - позволяет реализовать прототип системы на следующей стадии программной реализации.

Все чаще на этой стадии используется симбиоз языков представления знаний, например, в системе ОМЕГА - фреймы + семантические сети + полный набор возможностей языка исчисления предикатов. Средняя продолжительность 1 - 2 месяца.

Реализация

Создается прототип экспертной системы, включающий базу знаний и остальные блоки, при помощи одного из следующих способов:

- программирование на традиционных языках типа Паскаль, Си и др.;
- программирование на специализированных языках, применяемых в задачах искусственного интеллекта: LISP, FRL, SmallTalk и др.;
- использование инструментальных средств разработки ЭС типа СПЭИС, ПИЭС;
- использование "пустых" ЭС или "оболочек" типа ЭКСПЕРТ, ФИАКР и др.

Реализация - разработка программного комплекса, демонстрирующего жизнеспособность подхода в целом. Чаще всего первый прототип отбрасывается на этапе реализации действующей ЭС. Средняя продолжительность 1 - 2 месяца.

Тестирование

Оценивается и проверяется работа программ прототипа с целью приведения в соответствие с реальными запросами пользователей. Прототип проверяется на: удобство и адекватность интерфейсов ввода-вывода (характер вопросов в диалоге, связность выводимого текста результата и др.)

- эффективность стратегии управления (порядок перебора, использование нечеткого вывода и др.);
- качество проверочных примеров;
- корректность базы знаний (полнота и непротиворечивость правил).

Тестирование - выявление ошибок в подходе и реализации прототипа и выработка рекомендаций по доводке системы до промышленного варианта. Средняя продолжительность 1 - 2 недели.

Этап 3: Развитие прототипа до промышленной ЭС

При неудовлетворительном функционировании прототипа эксперт и инженер по знаниям имеют возможность оценить, что именно будет включено в разработку окончательного варианта системы.

Если первоначально выбранные объекты или свойства оказываются неподходящими, их необходимо изменить. Можно сделать оценку общего числа эвристических правил, необходимых для создания окончательного варианта экспертной системы. Иногда при разработке промышленной системы выделяют дополнительные этапы для перехода: *демонстрационный прототип - исследовательский прототип - действующий прототип - промышленная система.*

Однако чаще реализуется плавный переход от демонстрационного прототипа к промышленной системе, при этом, если программный инструментальный выбран удачно, необязательна перепись другими программными средствами.

Понятие же коммерческой системы в нашей стране входит в понятие промышленный программный продукт, или промышленной ЭС.

Таблица. Переход от прототипа к промышленной ЭС

Демонстрационный прототип ЭС	Система решает часть задач, демонстрируя ж из неспособность полхода (несколько десятков правил или понятий)
Исследовательский прототип ЭС	Система решает большинство задач, но не устойчива в работе и не полностью проверена [несколько сотен правил или понятий)
Действующий прототип ЭС	Система надежно решает все задачи на реальных примерах, но для сложной задачи требует много времени и памяти
Промышленная система	Система обеспечивает высокое качество решений при минимизации требуемого времени и памяти: переписывается с использованием более эффективных средств представления знаний
Коммерческая система	Промышленная система, пригодная к продаже, т.е. хорошо документирована и снабжена сервисом

Основное на третьем этапе заключается в добавлении большого числа дополнительных эвристик. Эти эвристики обычно увеличивают *глубину* системы, обеспечивая большее число правил для трудноуловимых аспектов отдельных случаев. В то же время эксперт и инженер по знаниям могут расширить охват системы, включая правила, управляющие дополнительными подзадачами или дополнительными аспектами экспертной задачи (метазнания).

После установления основной структуры ЭС инженер по знаниям приступает к разработке и адаптации интерфейсов, с помощью которых система будет общаться с пользователем и экспертом. Необходимо обратить особое внимание на языковые возможности интерфейсов, их простоту и удобство для управления работой ЭС. Система должна обеспечивать пользователю возможность легким и естественным образом спрашивать непонятное, приостанавливать работу и т.д. В частности, могут оказаться полезными графические представления.

На этом этапе разработки большинство экспертов узнают достаточно о вводе правил и могут сами вводить в систему новые правила. Таким образом, начинается процесс, во время которого инженер по знаниям передает право собственности и контроля за системой эксперту для уточнения, детальной разработки и обслуживания.

Этап 4: Оценка системы

После завершения этапа разработки промышленной экспертной системы необходимо провести ее тестирование в отношении критериев эффективности. К тестированию широко привлекаются другие эксперты с

целью апробирования работоспособности системы на различных примерах. Экспертные системы оцениваются главным образом для того, чтобы проверить точность работы программы и ее полезность. Оценку можно проводить, исходя из различных критериев, которые сгруппируем следующим образом:

- критерии пользователей (понятность и "прозрачность" работы системы, удобство интерфейсов и др.);
- критерии приглашенных экспертов (оценка советов-решений, предлагаемых системой, сравнение ее с собственными решениями, оценка подсистемы объяснений и др.);
- критерии коллектива разработчиков (эффективность реализации, производительность, время отклика, дизайн, широта охвата предметной области, непротиворечивость БЗ, количество тупиковых ситуаций, когда система не может принять решение, анализ чувствительности программы к незначительным изменениям в представлении знаний, весовых коэффициентах, применяемых в механизмах логического вывода, данных и т.п.).

Этап 5: Стыковка системы

На этом этапе осуществляется стыковка экспертной системы с другими программными средствами в среде, в которой она будет работать, и обучение людей, которых она будет обслуживать. Иногда это означает внесение существенных изменений. Такое изменение требует непрямого вмешательства инженера по знаниям или какого-либо другого специалиста, который сможет модифицировать систему. Под стыковкой подразумевается также разработка связей между экспертной системой и средой, в которой она действует.

Когда экспертная система уже готова, инженер по знаниям должен убедиться в том, что эксперты, пользователи и персонал знают, как эксплуатировать и обслуживать ее. После передачи им своего опыта в области информационной технологии инженер по знаниям может полностью предоставить ее в распоряжение пользователей.

Для подтверждения полезности системы важно предоставить каждому из пользователей возможность поставить перед ЭС реальные задачи, а затем проследить, как она выполняет эти задачи. Чтобы система была одобрена, необходимо представить ее как помощника, освобождающего пользователей от обременительных задач, а не как средство их замещения.

Стыковка включает обеспечение связи ЭС с существующими базами данных и другими системами на предприятии, а также улучшение системных факторов, зависящих от времени, чтобы можно было обеспечить ее более эффективную работу и улучшить характеристики ее технических средств, если система работает в необычной среде (например, связь с измерительными устройствами).

Пример. Успешно состыкована со своим окружением система PUFF - экспертная система для диагностики заболеваний легких. После того, как PUFF была закончена и все были удовлетворены ее работой, систему перекодировали с LISPа на Бейсик. Затем систему перенесли на ПК, которая уже работала и в больнице. В свою очередь, эта ПК была связана с измерительными приборами. Данные с измерительных приборов сразу поступают в ПК. PUFF обрабатывает эти данные и печатает рекомендации для врача. Врач в принципе не взаимодействует с PUFF. Система полностью интегрирована со своим окружением - она представляет собой интеллектуальное расширение аппарата исследования легких, который врачи давно используют.

Пример. Другая система, которая хорошо функционирует в своем окружении, - CAT-1 - экспертная система для диагностики неисправностей дизелей локомотивов.

Эта система была разработана также на LISPе, а затем переведена на FORTH, чтобы ее можно было более эффективно использовать в различных локомотивных цехах. Мастер по ремонту запрашивает систему: определить возможные причины неисправности дизеля. Система связана с видеодиском, с помощью которого мастеру дают визуальные объяснения и подсказки относительно более подробных проверок, которые ему нужно сделать.

Кроме того, если оператор не уверен в том, как устранить неисправность, система предоставляет ему обучающие материалы, которые фирма подготовила предварительно, и показывает ему на видеотерминале. Таким образом, мастер по ремонту может с помощью экспертной системы диагностировать проблему, найти тестовую процедуру, которую он должен использовать, получить на дисплее объяснение, как провести тест, или получит инструкции о том, как справиться с возникшей проблемой.

Этап 6: Поддержка системы

При перекодировании системы на язык, подобный Си, повышается ее быстродействие и увеличивается переносимость, однако гибкость при этом уменьшается. Это приемлемо лишь в том случае, если система сохраняет все знания проблемной области, и это знание не будет изменяться в ближайшем будущем. Однако, если экспертная система создана именно из-за того, что проблемная область изменяется, то необходимо поддерживать систему в инструментальной среде разработки.

Пример. Удачным примером ЭС, внедренной таким образом, является XCON (R1) - ЭС, которую фирма DEC использует для комплектации ЭВМ семейства VAX. Одна из ключевых проблем, с которой столкнулась фирма DEC, - необходимость постоянного внесения изменений для новых версий оборудования, новых спецификаций и т.д. Для этой цели

КЛАССЫ ЭКСПЕРТНЫХ СИСТЕМ

По степени сложности решаемых задач экспертные системы можно классифицировать следующим образом:

- По способу формирования решения экспертные системы разделяются на два класса: аналитические и синтетические. Аналитические системы предполагают выбор решений из множества известных альтернатив (определение характеристик объектов), а синтетические системы - генерацию неизвестных решений (формирование объектов).
- По способу учета временного признака экспертные системы могут быть статическими или динамическими. Статические системы решают задачи при неизменяемых в процессе решения данных и знаниях, динамические системы допускают такие изменения. Статические системы осуществляют монотонное непрерываемое решение задачи от ввода исходных данных до конечного результата, динамические системы предусматривают возможность пересмотра в процессе решения полученных ранее результатов и данных.
- По видам используемых данных и знаний экспертные системы классифицируются на системы с детерминированными (четко определенными) знаниями и неопределенными знаниями. Под неопределенностью знаний (данных) понимается их неполнота (отсутствие), недостоверность (неточность измерения), двусмысленность (многозначность понятий), нечеткость (качественная оценка вместо количественной).
- По числу используемых источников знаний экспертные системы могут быть построены с использованием одного или множества источников знаний. Источники знаний могут быть альтернативными (множество миров) или дополняющими друг друга (кооперирующими).

В соответствии с перечисленными признаками классификации, как правило, выделяются следующие четыре основные класса экспертных систем

	Анализ	Синтез	
Детерминированность знаний	Классифицирующие	Трансформирующие	Один источник знаний
	Доопределяющие	Мультиагентные	Несколько источников знаний
Неопределенность знаний	Статика	Динамика	

Классифицирующие ЭС решают задачи распознавания ситуаций, когда по набору заданных признаков (факторов) выявляется сущность некоторой ситуации, в зависимости от которой выбирается определенная

последовательность действий. Таким образом, в соответствии с исходными условиями среди альтернативных решений находится одно, наилучшим образом удовлетворяющее поставленной цели и ограничениям.

Основным методом формирования решений в таких системах является дедуктивный логический вывод.

Доопределяющие ЭС. Используются для решения задач с не полностью определенными данными и знаниями. В таких ЭС возникают задачи интерпретации нечетких знаний и выбора альтернативных направлений поиска в пространстве возможных решений. В качестве методов обработки неопределенных знаний могут использоваться байесовский вероятностный подход, коэффициенты уверенности, нечеткая логика, могут использоваться эвристические приемы выбора единиц знаний из их конфликтного набора, например, на основе использования приоритетов важности, или получаемой степени определенности результата, или значений функций предпочтений и т.д.

Для аналитических задач классифицирующего и доопределяющего типов характерны следующие проблемные области:

- Интерпретация данных - выбор решения из фиксированного множества альтернатив на базе введенной информации о текущей ситуации. Основное назначение - определение сущности рассматриваемой ситуации, выбор гипотез, исходя их фактов. Типичным примером является экспертная система анализа финансового состояния предприятия.
- Диагностика - выявление причин, приведших к возникновению ситуации. Требуется предварительная интерпретация ситуации с последующей проверкой дополнительных фактов, например, выявление факторов снижения эффективности производства.
- Коррекция - диагностика, дополненная возможностью оценки и рекомендации действий по исправлению отклонений от нормального состояния рассматриваемых ситуаций.

Трансформирующие ЭС относятся к синтезирующим динамическим экспертным системам, в которых предполагается повторяющееся преобразование знаний в процессе решения задач. В ЭС данного класса используются различные способы обработки знаний:

В качестве методов решения задач в трансформирующих экспертных системах используются разновидности гипотетического вывода:

- генерации и тестирования, когда по исходным данным осуществляется генерация гипотез, а затем проверка сформулированных гипотез на подтверждение поступающими фактами;
- предположений и умолчаний, когда по неполным данным подбираются знания об аналогичных классах объектов, которые в дальнейшем динамически адаптируются к конкретной ситуации в зависимости от ее развития;
- использование общих закономерностей (метауправления) в случае неизвестных ситуаций, позволяющих генерировать недостающее знание.

Мультиагентные системы - это динамические ЭС, основанные на интеграции нескольких разнородных источников знаний. Эти источники обмениваются между собой получаемыми результатами в ходе решения задач, например, через "доску объявлений"

Для таких динамических систем характерна интеграция в базе знаний нескольких разнородных источников знаний, обменивающихся между собой получаемыми результатами на динамической основе, Для мультиагентных систем характерны следующие особенности:

- Проведение альтернативных рассуждений на основе использования различных источников знаний с механизмом устранения противоречий;
- Распределенное решение проблем, которые разбиваются на параллельно решаемые подпроблемы, соответствующие самостоятельным источникам знаний;
- Применение множества стратегий работы механизма вывода заключений, в зависимости от типа решаемой проблемы;
- Обработка больших массивов данных, содержащихся в базе данных;
- Использование различных математических моделей и внешних процедур, хранимых в базе моделей;
- Способность прерывания решения задач в связи с необходимостью получения дополнительных данных и знаний от пользователей, моделей, параллельно решаемых подпроблем.

Для **синтезирующих динамических** экспертных систем наиболее применимы следующие проблемные области:

- Проектирование - определение конфигурации объектов с точки зрения достижения заданных критериев эффективности и ограничений, например, проектирование бюджета предприятия или портфеля инвестиций.
- Прогнозирование - предсказание последствий развития текущих ситуаций на основе математического и эвристического моделирования, например, прогнозирование трендов на биржевых торгах.

- Диспетчирование - распределение работ во времени, составление расписаний, например, планирование графика освоения капиталовложений.
- Планирование - выбор последовательности действий пользователей по достижению поставленной цели, например, планирование процессов поставки продукции.
- Мониторинг - слежение за текущей ситуацией с возможной последующей коррекцией. Для этого выполняется диагностика, прогнозирование, а в случае необходимости планирование и коррекция действий пользователей, например, мониторинг сбыта готовой продукции.
- Управление - мониторинг, дополненный реализацией действий в автоматических системах, например, принятие решений на биржевых торгах.

По данным публикации, в которой проводится анализ 12500 действующих экспертных систем, распределение экспертных систем по проблемным областям имеет следующий вид:

В основе самообучающихся систем лежат методы автоматической классификации примеров ситуаций реальной практики (обучения на примерах). Примеры реальных ситуаций накапливаются за некоторый исторический период и составляют обучающую выборку. Эти примеры описываются множеством признаков классификации. Причем обучающая выборка может быть:

- "с учителем", когда для каждого примера задается в явном виде значение признака его принадлежности некоторому классу ситуаций (классообразующего признака);
- "без учителя", когда по степени близости значений признаков классификации система сама выделяет классы ситуаций.

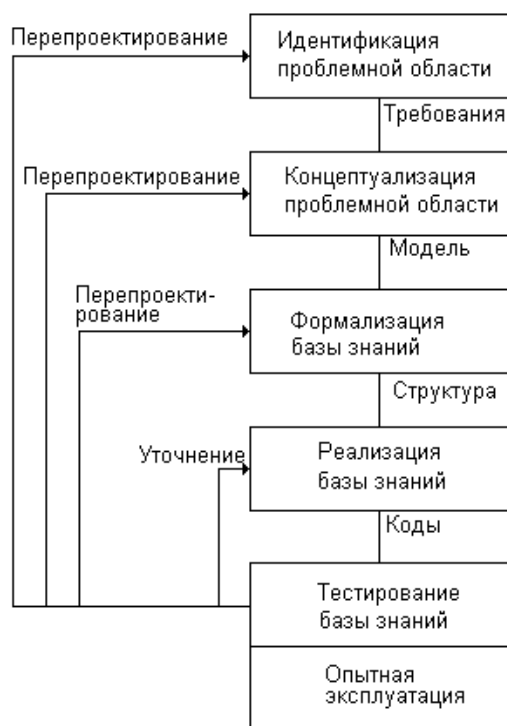
ТЕХНОЛОГИЯ СОЗДАНИЯ ЭКСПЕРТНЫХ СИСТЕМ -

Этапы создания экспертной системы

- Слабая формализуемость процесса принятия решений, его альтернативность и нечеткость, качественная и символьная природа используемых знаний, динамичность изменения проблемной области - все эти характерные особенности применения экспертных систем обуславливают сложность и большую трудоемкость их разработки по сравнению с другими подклассами ИИС. Поэтому в дальнейшем вопросы

проектирования и реализации интеллектуальных информационных систем будут рассматриваться для класса экспертных систем.

- Извлечение знаний при создании экспертной системы предполагает изучение множества источников знаний, к которым относятся специальная литература, базы фактуальных знаний, отчеты о решении аналогичных проблем, а самое главное, опыт работы специалистов в исследуемой проблемной области - экспертов. Успех проектирования экспертной системы во многом определяется тем, насколько компетентны привлекаемые к разработке эксперты и насколько они способны передать свой опыт инженерам по знаниям. Вместе с тем, эксперты не имеют представления о возможностях и ограничениях ЭС. Следовательно процесс разработки ЭС должен быть организован инженерами по знаниям таким образом, чтобы в процессе их итеративного взаимодействия с экспертами они получили весь необходимый объем знаний для решения четко очерченных проблем. Этапы проектирования экспертной системы представлены на рисунке.
- На начальных этапах идентификации и концептуализации, связанных с определением контуров будущей системы, инженер по знаниям выступает в роли ученика, а эксперт - в роли учителя, мастера. На заключительных этапах реализации и тестирования инженер по знаниям демонстрирует результаты разработки, адекватность которой проблемной области оценивает эксперт. На этапе тестирования это могут быть совершенно другие эксперты.
- На этапе тестирования созданные экспертные системы оцениваются с позиции двух основных групп критериев: точности и полезности.
- С точностью работы связаны такие характеристики, как правильность делаемых заключений, адекватность базы знаний проблемной области, соответствие применяемых методов решения проблемы экспертным. Поэтому конечные оценки системе ставят специалисты в проблемной области - эксперты. Полезность же экспертной системы характеризуется степенью удовлетворения требований пользователя в части получения необходимых рекомендаций, легкости и естественности взаимодействия с системой, надежности, производительности и стоимости эксплуатации, способности обоснования решений и обучения, настройки на изменение потребностей. Оценивание экспертной системы осуществляется по набору тестовых примеров как из предшествующей практики экспертов, так и специально подобранных ситуаций. Результаты тестирования подлежат статистической обработке, после чего делаются выводы о степени точности работы экспертной системы.



- Следующий этап жизненного цикла экспертной системы - внедрение и опытная эксплуатация в массовом порядке без непосредственного контроля со стороны разработчиков и переход от тестовых примеров к решению реальных задач. Важнейшим критерием оценки становятся соотношение стоимости системы и ее эффективности. На этом этапе осуществляется сбор критических замечаний и внесение необходимых изменений. В результате опытной эксплуатации может потребоваться разработка новых специализированных версий, учитывающих особенности проблемных областей.
- На всех этапах разработки инженер по знаниям играет активную роль, а эксперт - пассивную. По мере развития самообучающихся свойств экспертных систем роль инженера по знаниям уменьшается, а активное поведение заинтересованного в эффективной работе экспертной системы пользователя-эксперта возрастает. Описание приемов извлечения знаний инженерами знаний представлено в таблице.

Приемы	Описание
1. Наблюдение	Инженер наблюдает, не вмешиваясь, за тем, как эксперт решает реальную задачу
2. Обсуждение задачи	Инженер на представительном множестве задач неформально обсуждает с экспертом данные, знания и процедуры решения
3. Описание задачи	Эксперт описывает решение задач для типичных запросов

4. Анализ решения	Эксперт комментирует получаемые результаты решения задачи, детализируя ход рассуждений
5. Проверка системы	Эксперт предлагает инженеру перечень задач для решения (от простых до сложных), которые решаются разработанной системой
6. Исследование системы	Эксперт исследует и критикует структуру базы знаний и работу механизма вывода
7. Оценка системы	Инженер предлагает новым экспертам оценить решения разработанной системы

- Первые два этапа разработки экспертной системы составляют логическую стадию, не связанную с применением четко определенного инструментального средства. Последующие этапы реализуются в рамках физического создания проекта на базе выбранного инструментального средства. Вместе с тем, процесс создания экспертной системы, как сложного программного продукта, имеет смысл выполнять методом прототипного проектирования, сущность которого сводится к постоянному наращиванию базы знаний, начиная с логической стадии. Технология разработки прототипов представлена в таблице.

Этап разработки	Характер прототипа	Количество правил	Срок разработки	Стоимость
Идентификация	Демонстрационный	50 - 100	1 - 2 мес.	
Концептуализация	Исследовательский			
Формализация	Исследовательский	200 - 500 3 - 6 мес.		25 - 50т.\$
Реализация	Действующий	500 - 1000	6 - 12 мес.	
Тестирование	Промышленный	1000 - 1500	1 - 1,5 года	300тыс.\$
Опытная эксплуатация	Коммерческий	1500 - 3000	1,5 - 3 года	2 - 5 млн.\$

- Прототипная технология создания экспертной системы означает, что простейший прототип будущей системы реализуется с помощью любого подручного инструментального средства еще на этапах идентификации и концептуализации, в дальнейшем этот прототип детализируется, концептуальная модель уточняется, реализация выполняется в среде окончательно выбранного инструментального средства. После каждого этапа возможны итеративные возвраты на уже выполненные этапы проектирования, что способствует постепенному проникновению инженера по знаниям в глубину решаемых проблем, эффективности использования выделенных ресурсов, сокращению времени разработки, постоянному улучшению компетентности и производительности системы.
- Пример разработки экспертной системы гарантирования (страхования) коммерческих займов CLUES (loan-uderwriting expert systems) представлен в таблице. Эта система создавалась в интегрированной среде ART группой разработчиков в составе одного менеджера проекта, двух инженеров по знаниям, двух программистов, ответственных за сопряжение ЭС с существующей информационной системой и аналитическим инструментом, одного контролера качества. Сложность созданной системы: 1000 правил, 180 функций, 120 объектов. Эффективность: при оценке 8500 кредитов в месяц годовая экономия на обработке информации составляет 0,91 млн. долл., при 30000 кредитов - 2,7 млн. долл. При этом в 50% случаев система принимает самостоятельные решения, в остальных случаях дает экспертам диагностику возникающих проблем. Время оценки кредита сократилось с 50 минут до 10-15 минут. Перечисленные показатели эффективности позволили компании Contrywide расширить сферу своей деятельности во всех штатах США и увеличить оборот с 1 млрд. долл. в месяц в 1991 году до 5 млрд. долл. в 1993 году.

Идентификация проблемной области

Этап идентификации проблемной области включает определение назначения и сферы применения экспертной системы, подбор экспертов и группы инженеров по знаниям, выделение ресурсов, постановку и параметризацию решаемых задач.

Начало работ по созданию экспертной системы инициируют руководители компаний (предприятий, учреждений). Обычно необходимость разработки экспертной системы в той или иной сфере деятельности связана с затруднениями лиц, принимающих решение, что сказывается на эффективности функционирования проблемной области. Эти затруднения могут быть обусловлены недостаточным опытом работы в данной области, сложностью постоянного привлечения экспертов, нехваткой трудовых ресурсов для решения простых интеллектуальных задач, необходимостью интеграции разнообразных источников знаний. Как правило, назначение экспертной системы связано с одной из следующих областей:

- обучение и консультация неопытных пользователей;
- распространение и использование уникального опыта экспертов;
- автоматизация работы экспертов по принятию решений;
- оптимизация решения проблем, выдвижение и проверка гипотез.

Период времени	Этап
Ноябрь 1991г.	Постановка проблемы
Январь 1992г.	Создание отдела ЭС
Февраль - апрель 1992г.	Интервьюирование экспертов
Апрель - май 1992г.	Моделирование и создание первого прототипа
Май - июнь 1992г.	Кодирование (реализация)
Июнь - сентябрь 1992г.	Внутреннее тестирование. Системная интеграция
Сентябрь - декабрь 1992г.	Альфа-тестирование на известных примерах
Декабрь - январь 1993г.	Бета-тестирование на реальных примерах
Февраль 1993г.	Внедрение в отрасли розничной торговли (20% кредитов)
Май 1993г.	Внедрение в потребительский сектор

	(10% кредитов)
Август 1993г.	Внедрение в отрасли оптовой торговли (35% кредитов)
Февраль 1994г.	Внедрение в корреспондентскую сеть (35% кредитов)

Сфера применения экспертной системы характеризует тот круг задач, который подлежит формализации, например, "оценка финансового состояния предприятия", "выбор поставщика продукции", "формирование маркетинговой стратегии" и т.д. Обычно сложность решаемых в экспертной системе проблем должна соответствовать трудоемкости работы эксперта в течение нескольких часов. Более сложные задачи имеет смысл разбивать на совокупности взаимосвязанных задач, которые подлежат разработке в рамках нескольких экспертных систем.

Ограничивающими факторами на разработку экспертной системы выступают отводимые сроки, финансовые ресурсы и программно-техническая среда. От этих ограничений зависит количественный и качественный состав групп инженеров по знаниям и экспертов, глубина прорабатываемых вопросов, адекватность и эффективность решения проблем. Обычно различают три стратегии разработки экспертных систем (таблица):

- широкий набор задач, каждая из которых ориентирована на узкую проблемную область;
- концентрированный набор задач, определяющий основные направления повышения эффективности функционирования экономического объекта;
- комплексный набор задач, определяющий организацию всей деятельности экономического объекта.

После предварительного определения контуров разрабатываемой экспертной системы инженеры по знаниям совместно с экспертами осуществляют более детальную постановку проблем и параметризацию системы. К основным параметрам проблемной области относятся следующие:

- класс решаемых задач (интерпретация, диагностика, коррекция, прогнозирование, планирование, проектирование, мониторинг, управление);
- критерии эффективности результатов решения задач (минимизация использования ресурсов, повышение качества продукции и обслуживания, ускорение оборачиваемости капитала и т.д.);
- критерии эффективности процесса решения задач (повышение точности принимаемых решений, учет большего числа факторов,

просчет большего числа альтернативных вариантов, адаптивность к изменениям проблемной области и информационных потребностей пользователей, сокращение сроков принятия решений);

- цели решаемых задач (выбор из альтернатив, например, выбор поставщика или синтез значения, например, распределение бюджета по статьям);
- подцели (разбиение задачи на подзадачи, для каждой из которых определяется своя цель);
- исходные данные (совокупность используемых факторов);
- особенности используемых знаний (детерминированность/неопределенность, статичность/динамичность, одноцелевая/многоцелевая направленность, единственность/множественность источников знаний).

Стратегии разработки экспертных систем			
	Широкий набор задач	Концентрированный набор задач	Комплексный набор задач
Назначение	Автоматизация	Стандартизация, повышение качества	Реорганизация бизнес-процессов
Требования к разработчикам	Эксперты-пользователи	Профессиональные команды	Междисциплинарные команды
Стоимость	Низкая на проект	Высокая на проект	Высокая на проект
Риск	Диверсифицированный	Концентрированный	Концентрированный
Примеры	DuPont du Nemours Оболочка Insight Plus Сотни экспертных систем. Сотни правил в каждой ЭС	DEC, ЭС конфигурирования компьютеров XCON, продажи XSEL 17000 правил, эффект 27 млн. долл.	Херох Среда разработки информационной системы ART-Enterprise (Inference) Интеллект. моделирование ReThink (Gensym)

МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Построение концептуальной модели

На этапе построения концептуальной модели создается целостное и системное описание используемых знаний, отражающее сущность

функционирования проблемной области. От качества построения концептуальной модели проблемной области во многом зависит насколько часто в дальнейшем по мере развития проекта будет выполняться перепроектирование базы знаний. Хорошая концептуальная модель может только уточняться (детализироваться или упрощаться), но не перестраиваться.

Результат концептуализации проблемной области обычно фиксируется в виде наглядных графических схем на объектном, функциональном и поведенческом уровнях моделирования:

- объектная модель описывает структуру предметной области как совокупности взаимосвязанных объектов;
- функциональная модель отражает действия и преобразования над объектами;
- поведенческая модель рассматривает взаимодействия объектов во временном аспекте.

Первые две модели описывают статические аспекты функционирования проблемной области, а третья модель - динамику изменения ее состояний. Естественно, что для различных классов задач могут требоваться разные виды моделей, а следовательно, и ориентированные на них методы представления знаний. Рассмотрим каждую из представленных видов моделей.

Объектная модель отражает фактуальное знание о составе объектов, их свойств и связей. Элементарной единицей структурного знания является факт, описывающий одно свойство или одну связь объекта, который представляется в виде триплета:

предикат (Объект, Значение).

Если предикат определяет название свойства объекта, то в качестве значения выступает конкретное значение этого свойства, например:

профессия ("Иванов", "Инженер").

Если предикат определяет название связи объекта, то значению соответствует объект, с которым связан первый объект, например:

Работает ("Иванов", "Механический цех").

В качестве важнейших типизированных видов отношений рассматриваются следующие отношения:

- "род" - "вид" (обобщение);
- "целое" - "часть" (агрегация);
- "причина" - "следствие";
- "цель" - "средство";
- "функция" - "аргумент";
- "ассоциация";

- "хронология";
- "пространственное положение" и др.

Так, отношения обобщения ("род" - "вид") фиксируется на уровне названий классов объектов, например:

есть-подкласс (Инженеры, Личности).

Под классом объектов понимается совокупность объектов с одинаковым набором предикатов (свойств и связей). Класс объектов часто описывается в виде n-арного реляционного отношения, например:

личности (ФИО, Профессия, Подразделение, ...).

Если объекты обладают частично пересекающимся набором предикатов, то осуществляется более сложная классификация объектов: класс объектов по значениям какого-либо свойства (признака) разбивается на подклассы таким образом, что класс объектов содержит общие для подклассов свойства и связи, а каждый из подклассов отражает специфические свойства и связи, например:

личности (ФИО, Год рождения, Профессия, Подразделение, ...)

инженеры (ФИО, ВУЗ, Оклад, ...)

рабочие (ФИО, Разряд, Тарифная ставка, ...)

При этом подклассы объектов автоматически наследуют общие свойства и связи вышестоящих классов, а совокупность взаимосвязанных по отношению обобщения классов объектов образует иерархию наследования свойств.

Отношение агрегации классов объектов ("целое" - "часть") отражает составные части объектов, которое можно представить в бинарном виде на именах двух классов объектов:

- есть-часть (Оборудование, Цех);
- есть-часть (Рабочие, Цех).

Аналогично представляются другие семантические отношения:

- причина-следствие (Задолженность, Банкротство);
- аргумент-функция (Спрос, Цена);
- средство-цель (Покупка акций, Прибыль);
- ассоциация (Производство, Обслуживание);
- хронология (Отгрузка, Поставка);
- пространственное положение (Сборка, Технический контроль).

Обычно объектное знание представляется графически средствами ER-моделей (модель "Сущность - Связь")

Функциональная модель описывает преобразования фактов, зависимости между ними, показывающие, как одни факты образуются из других. В качестве единицы функционального знания определим функциональную зависимость фактов в виде импликации:

$$A1 \wedge A2 \wedge A3 \wedge \dots \wedge An \rightarrow B,$$

означающей, что факт B имеет место только в том случае, если имеет место конъюнкция фактов или их отрицаний A1, A2, ..., An, например:

*сбыт (Товар, "Слабый") и
прибыль (Товар, "Ничтожная") и
потребители (Товар, "Любители нового") и
число_конкурентов (Товар, "Небольшое") ->
жизненный_цикл (Товар, "Выведение на рынок").*

Функциональную зависимость фактов можно трактовать как отражение следующих отношений фактов:

- "Причина" - "Следствие";
- "Средство" - "Цель";
- "Аргумент" - "Функция";
- "Ситуация" - "Действие".

В качестве термов конъюнкции фактов могут выступать более сложные логические условия.

Функциональная модель строится путем последовательной декомпозиции целей, а именно: для цели определяются подцели, для которых в свою очередь устанавливаются подцели и так дальше, пока в качестве подцелей не окажутся исходные факты (процесс декомпозиции "сверху" - "вниз"). Каждой цели (подцели) соответствует некоторая задача (подзадача), которая не может быть решена, пока не будут достигнуты ее нижестоящие подцели (решены подзадачи). Таким образом, функциональная модель отражает в обобщенной форме процесс решения характерных для нее задач.

Обычно функциональные зависимости фактов представляются графически в виде деревьев целей или графов "И" - "ИЛИ", в которых каждый зависимый факт представляет собой целевую переменную - корневую вершину, а определяющие его факты-аргументы - связанные с корнем подчиненные вершины, условие конъюнкции (совместности анализа

факторов) обозначается пересекающей дугой, а условие дизъюнкции (независимости влияния на цель факторов) никак не обозначается, причем если какой-либо факт-аргумент, в свою очередь, определяется другими фактами-аргументами, то он становится подцелью.

Поведенческая модель отражает изменение состояний объектов в результате возникновения некоторых событий, влекущих за собой выполнение определенных действий (процедур). Состояние объекта - это изменяющиеся во времени значения некоторого свойства. Набор действий, связанный с некоторым событием, составляет поведение объекта, которое выражается в виде правил или процедур. Задача определения поведенческой модели заключается в определении связей событий с поведением объектов и изменением их состояний. Как правило, событие отражается в форме сообщения, посылаемого объекту. Пример поведенческой модели в виде диаграммы потоков событий представлен в таблице.

Заказ		
СОБЫТИЕ (Сообщение)	ПОВЕДЕНИЕ (Действие)	СОСТОЯНИЕ (Жизненный цикл)
Оформляется	Создание заказа Проверка выполнимости	Оформлен
Откладывается	Заказ у производителя	Отложен
Выполняется	Отправляется Выписываются платежные документы	Выполнен
Оплачивается	Уничтожение заказа	Оплачен

ФОРМАЛИЗАЦИЯ БАЗЫ ЗНАНИЙ

На этапе формализации базы знаний осуществляется выбор метода представления знаний. В рамках выбранного формализма осуществляется проектирование логической структуры базы знаний.

Рассмотрим классификацию методов представления знаний с точки зрения особенностей отображения различных видов концептуальных моделей, а именно: соотношения структурированности и операционности,

детерминированности и неопределенности, статичности и динамичности знаний.

Так, объектные методы представления знаний в большей степени ориентированы на представление структуры фактуального знания, а правила - операционного.

- Логическая модель реализует и объекты, и правила с помощью предикатов первого порядка, является строго формализованной моделью с универсальным дедуктивным и монотонным методом логического вывода "от цели к данным";
- Продукционная модель позволяет осуществлять эвристические методы вывода на правилах и может обрабатывать неопределенности в виде условных вероятностей или коэффициентов уверенности, а также выполнять монотонный или немонотонный вывод;
- Семантическая сеть отображает разнообразные отношения объектов;
- Фреймовая модель, как частный случай семантической сети, использует для реализации операционного знания присоединенные процедуры;
- Объектно-ориентированная модель, как развитие фреймовой модели, реализуя обмен сообщениями между объектами, в большей степени ориентирована на решение динамических задач и отражение поведенческой модели.

Логическая модель представления знаний

Логическая модель предполагает унифицированное описание объектов и действий в виде предикатов первого порядка. Под предикатом понимается логическая функция на N - аргументах (признаках), которая принимает истинное или ложное значение в зависимости от значений аргументов. Отличие заключается в том, что для объектов соответствующие реляционные отношения задаются явно в виде фактов, а действия описываются как правила, определяющие логическую формулу вывода фактов из других фактов. Пример фрагмента базы знаний подбора претендентов на вакансии в языке логического программирования ПРОЛОГ представлен на рис. 2.4. (Обозначения: ":" - "если", "," - "и", "." - "конец утверждения").

vibor(Fio,Dolgnost):-

pretendent(Fio, Obrazov, Stag),

vacancy(Dolgnost, Obrazov, Opyt),

Stag >= Opyt.

pretendent("Иванов", "среднее" 10).

pretendent("Петров", "высшее", 12).

vacancy("менеджер", "высшее", 10).

vacancy("директор", "высшее", 15).

Механизм вывода осуществляет дедуктивный перебор фактов, относящихся к правилу по принципу "сверху - вниз", "слева - направо" или обратный вывод методом поиска в глубину. Так, в ответ на запрос *vibor(X, Y)* получим: *X="Петров", Y="менеджер"*.

Правила могут связываться в цепочки в результате использования одинакового предиката в посылке одного и в заключении другого правила.

Для логической модели характерна строгость формального аппарата получения решения. Однако, полный последовательный перебор всех возможных решений может приводить к комбинаторным взрывам, в результате чего поставленные задачи могут решаться недопустимо большое время. Кроме того, работа с неопределенностями знаний должна быть запрограммирована в виде самостоятельных метаправил, что на практике затрудняет разработку баз знаний с помощью логического формализма.

Продукционные модели представления знаний

Продукционные модели используются для решения более сложных задач, которые основаны на применении эвристических методов представления знаний, позволяющих настраивать механизм вывода на особенности проблемной области и учитывать неопределенность знаний.

В продукционной модели основной единицей знаний служит правило в виде: - "**если <посылка>, то <заключение>**", с его помощью выражены:

- пространственно-временные,
- причинно-следственные,
- функционально-поведенческие (ситуация - действие) отношения объектов.

Правилами могут быть описаны и сами объекты: "*объект - свойство*" или "*набор свойств - объект*", хотя чаще объекты описывают только в качестве переменных ("*атрибут - значение*") внутри правил. Продукционная модель предназначена для описания последовательности различных ситуаций или действий и в меньшей степени для структурированного описания объектов.

Продукционная модель предполагает более гибкую организацию работы механизма вывода по сравнению с логической моделью. В зависимости от

направления вывода возможна как прямая аргументация, управляемая данными (от данных к цели), так и обратная, управляемая целями (от целей к данным).

Прямой вывод используется в продукционных моделях при решении, например, задач интерпретации, когда по исходным данным нужно определить сущность некоторой ситуации или в задачах прогнозирования, когда из описания некоторой ситуации требуется вывести все следствия.

Обратный вывод применяется, когда нужно проверить определенную гипотезу или небольшое множество гипотез на соответствие фактам, например, в задачах диагностики.

Отличительной особенностью продукционной модели является способность осуществлять выбор правил из множества возможных на данный момент времени (из конфликтного набора) в зависимости от определенных критериев, например, важности, трудоемкости, достоверности получаемого результата и других характеристик проблемной области. Такая стратегия поиска решений называется поиском в ширину. Для ее реализации в описание продукции вводятся предусловия и постусловия в виде:

$\langle A, B, C \rightarrow D, E \rangle$, где

- импликация $C \rightarrow D$ представляет собственно правило;
- A - предусловие выбора класса правил;
- B - предусловие выбора правила в классе;
- E - постусловие правила, определяющее переход на следующее правило.

В предусловиях и постусловиях могут быть заданы дополнительные процедуры, например, по вводу и контролю данных, математической обработке и т.д. Введение предусловий и постусловий позволяет выбирать наиболее рациональную стратегию работы механизма вывода, существенно сокращая перебор относящихся к решению правил.

Сами правила могут иметь как простой, так и обобщенный характер. Простые правила описывают продукции над единичными объектами, обобщенные правила определяются на классах объектов (аналогично правилам языка ПРОЛОГ).

Для обработки неопределенностей знаний продукционная модель использует, как правило, либо методы обработки условных вероятностей Байеса, либо методы нечеткой логики Заде.

Байесовский подход предполагает начальное априорное задание предполагаемых гипотез (значений достигаемых целей), которые последовательно уточняются с учетом вероятностей свидетельств в пользу

или против гипотез, в результате чего формируются апостериорные вероятности:

$$P(H/E) = P(E/H) * P(H) / P(E) \text{ и } P(\bar{H}/E) = P(E/\bar{H}) * P(\bar{H}) / P(E),$$

где $P(H)$ - априорная вероятность гипотезы H ;

$P(\bar{H}) = 1 - P(H)$ - априорная вероятность отрицания гипотезы H ;

$P(E)$ - априорная вероятность свидетельства E ;

$P(H/E)$ - апостериорная (условная) вероятность гипотезы H при условии, что имеет место свидетельство E ;

$P(\bar{H}/E)$ - апостериорная (условная) вероятность отрицания гипотезы H при условии, что имеет место свидетельство E ;

$P(E/H)$ - вероятность свидетельства гипотезы E при подтверждении гипотезы H ;

$P(E/\bar{H})$ - вероятность свидетельства гипотезы E при отрицании гипотезы H .

Найдем отношения левых и правых частей представленных уравнений:

$$P(H/E) / P(\bar{H}/E) = (P(E/H) / P(E/\bar{H})) * (P(H) / P(\bar{H})) \text{ или } O(H/E) = L_s * O(H)$$

,

где

- $O(H)$ - априорные шансы гипотезы H , отражающие отношение числа позитивных проявлений гипотезы к числу негативных;

- $O(H/E)$ - апостериорные шансы гипотезы H при условии наличия свидетельства E ;

- L_s - фактор достаточности, отражающий степень воздействия на шансы гипотезы при наличии свидетельства E .

Аналогично выводится зависимость:

$O(H/\bar{E}) = L_n * O(H)$, где - $O(H/\bar{E})$ - апостериорные шансы гипотезы H при условии отсутствия свидетельства E ;

- L_n - фактор необходимости, отражающий степень воздействия на шансы гипотезы при отсутствии свидетельства E .

Шансы и вероятности связаны уравнениями:

$$O = P / (1 - P) \text{ и } P = O / (O + 1)$$

Отсюда апостериорная вероятность гипотезы рассчитывается через апостериорные шансы, которые в свою очередь получают перемножением априорных шансов на факторы достаточности или необходимости всех относящихся к гипотезе свидетельств в зависимости от их подтверждения или отрицания со стороны пользователя. Свидетельства рассматриваются как независимые аргументы на дереве целей.

Рассмотрим использование байесовского подхода на примере оценки надежности поставщика. Фрагмент подмножества правил представляется следующим образом:

Если Задолженность = "есть" ,

То Финансовое_состояние = "удовл." $L_s = 0.01$, $L_n = 10$

Если Рентабельность = "есть" ,
То Финансовое_состояние = "удовл." $L_s = 100$, $L_n = 0.001$

Пусть оцениваемое предприятие является рентабельным и без задолженностей. Априорная вероятность удовлетворительного финансового состояния любого поставщика составляет 0.5. Тогда расчет апостериорных шансов и вероятности удовлетворительного финансового состояния осуществляется по формулам:

$$O(H/E1,E2) = 1 * 10 * 100 = 1000;$$
$$P(H/E1,E2) = 1000 / 1001 = 0.99.$$

Для байесовского подхода к построению продукционной базы знаний характерна большая трудоемкость статистического оценивания априорных шансов и факторов достаточности и необходимости.

Подход на основе нечеткой логики. Более простым, но менее точным методом оценки достоверности используемых знаний является применение нечеткой логики, в которой вероятности заменяются на экспертные оценки определенности фактов и применения правил (факторы уверенности).

Факторы уверенности могут рассматриваться и как весовые коэффициенты, отражающие степень важности аргументов в процессе вывода заключений. *Итоговые факторы уверенности* получаемых решений главным образом отражают порядок достоверности результата, а не его точность, что вполне приемлемо во многих задачах.

Факторы уверенности измеряются по некоторой относительной шкале, например, от 0 до 100. В отличие от теории вероятностей сумма факторов уверенностей некоторых альтернативных значений необязательно составляет 100. Множество возможных значений некоторой переменной с различными факторами уверенностей для каждого значения составляет нечеткое множество вида: $\{ x_1 \text{ cf}_1, x_2 \text{ cf}_2, \dots, x_N \text{ cf}_N \}$, причем фактор уверенности в общем виде задается функцией принадлежности значений нечеткому множеству, например, как представлено на рис. 2.5. (Например, для рентабельности 6,7% получаем оценку в виде нечеткого множества {"неудовлетворительно" cf 66, "удовлетворительно" cf 33}).

Предполагается, что оценка факторов уверенностей исходных данных задается пользователем при описании конкретной ситуации, а факторы уверенности применения правил определяются инженерами знаний совместно с экспертами при наполнении базы знаний.

При объединении факторов уверенности конъюнктивно или дизъюнктивно связанных аргументов используются следующие формулы:

Конъюнкция (A и B) : $\min(cfA, cfB)$ или $cfA*cfB/100$ Дизъюнкция (A или B):
 $\max(cfA, cfB)$ или $cfA + cfB - cfA*cfB/100$

Объединение факторов уверенности в посылках правил осуществляется чаще всего, как например, в программном средстве GURU, по формулам "min/max", а левых и правых частей правил и одинаковых результатов нескольких правил соответственно по формулам "произведение" и "сумма". Для объединения одинаковых результатов нескольких правил используется оператор "+=", который означает не присваивание значения, а добавление значения. Аналогично используется оператор "- =" для удаления значения. Факторы уверенности в последнем случае объединяются по формуле:

$$cfA*(100 - cfB)/100$$

Рассмотрим применение аппарата нечеткой логики на примере оценки надежности поставщика, в котором кроме фактора финансового состояния учитывается и фактор формы собственности. Пусть государственное предприятие не имеет задолженность с уверенностью 60 и предполагается, что его рентабельность удовлетворительна с уверенностью 80. Фрагмент множества правил имеет следующий вид:

Правило 1: Если Задолженность = "нет" и Рентабельность = "удовл."

То Финансовое_состояние = "удовл." cf 100

Правило 2: Если Финансовое_состояние = "удовл."

То Надежность += "есть" cf 90

Правило 3: Если Предприятие = "государств."

То Надежность+ = "есть" cf 50

Результат выполнения первого правила:

$$cf(\text{посылки}) = \min(60, 80) = 60,$$

$$cf(\text{Фин_сост.} = \text{"удовл."}) = 60*100/100 = 60.$$

Результат выполнения второго правила: $cf(\text{Надежность} = \text{"есть"}) =$

$$60*90/100 = 54$$

Результат выполнения третьего правила: $cf(\text{Надежность} = \text{"есть"}) = 54 +$

$$50 - 54*50/100 = 67$$

Динамические модели представления знаний

Моделирование рассуждений человека, как правило, не сводится только к прямой или обратной аргументации. Сложные проблемы решаются путем выдвижения во времени нескольких гипотез с анализом подтверждающих фактов и непротиворечивости следствий. Причем для многоцелевых проблемных областей происходит увязка гипотез по общим ограничениям. При этом возможны задержки в принятии решений, связанные со сбором подтверждающих фактов, доказательством подцелей, входящих в ограничения.

Следовательно, для подобных динамических проблем важна рациональная организация памяти системы для запоминания и обновления получаемых промежуточных результатов, обмен данными между различными источниками знаний для достижения нескольких целей, изменение стратегий вывода с выдвижения гипотез (прямая аргументация) к их проверке (обратная аргументация). Целям построения таких гибких механизмов вывода служит применение технологии "доски объявлений", через которую в результате осуществления событий источники знаний обмениваются сообщениями.

В целях динамического реагирования на события некоторые продукционные модели используют специальные правила-демоны, которые формулируются следующим образом:

"Всякий раз, как происходит некоторое событие, выполнить некоторое действие". Например:

Всякий раз, как становится известным значение переменной "Поставщик",

Выполнить набор правил "Финансовый анализ предприятия"

В программном средстве GURU подобное правило будет записано следующим образом:

IF: KNOWN("Поставщик") = true

THEN: CONSULT FIN_AN

Для динамических экспертных систем характерна также обработка времени как самостоятельного атрибута аргументации логического вывода:

Если в течение дня уровень запаса понизился больше, чем на 50 %

То выполнить набор правил "Выбор поставщика для поставки"

Общим недостатком всех формализмов представления знаний, основанных на правилах, является недостаточно глубокое отражение семантики проблемной области, что может сказываться на гибкости формулирования запросов пользователей к экспертным системам. Этот недостаток снимается в объектно-ориентированных методах представления знаний.

Семантические сети - представление знаний

Объектно-ориентированные методы представления знаний берут начало от семантических сетей, в которых типизируются отношения между объектами. Элементарной единицей знаний в семантической сети служит триплет (см. объектную концептуальную модель), в котором имя предиката представляет помеченную дугу между двумя узлами графа, соответствующими двум связанным объектам (рис. 2.6):

Важнейшими типизированными отношениями объектов являются:

"Род" - "Вид",
"Целое" - "Часть",
"Причина" - "Следствие",
"Средство" - "Цель",
"Аргумент" - "Функция",
"Ситуация" - "Действие".

Типизация отношений позволяет однозначно интерпретировать смысл отображаемых в базе знаний ситуаций и настраивать механизм вывода на особенности этих отношений. Так, отображение отношений "Род" - "Вид" дает возможность осуществлять наследование атрибутов классов объектов и, таким образом, автоматизировать процесс вывода заключений от общего к частному.

Фреймы - представление знаний

Развитием семантических сетей являются фреймовые методы представления знаний, в которых все атрибуты (поименованные отношения) объектов собираются в одну структуру данных, называемую фреймом. Причем в качестве значений слотов (атрибутов) могут выступать как обычные значения данных, так и действия, направленные на получение этих значений. Таким образом, действия реализуются в виде присоединенных процедур или процедур-демонов, вызываемых по определенным условиям. В этом плане фреймовый метод представления знаний в большей степени операционно-ориентирован по сравнению с семантической сетью.

Неопределенность описания знаний реализуется в результате неполного заполнения всех слотов. Фреймовая модель способна делать предположения о значениях данных на основе механизма наследования свойств в иерархии обобщения. В качестве способов наследования атрибутов применяются следующие возможности: S - идентичность значений одноименных слотов; U - различные значения одноименных слотов; R - значение слота фрейма должно находиться в пределах, заданных в одноименном слоте фрейма верхнего уровня; O - в случае неизвестности значения слота фрейма нижнего уровня принимается значение слота фрейма верхнего уровня.

Способность изменения значений слотов с течением времени позволяет решать динамические задачи. Во фреймовых моделях могут выполняться как прямая, так и обратная аргументация, когда в прямом направлении в зависимости от состояния слотов фреймов запускаются процедуры-демоны (неизвестно значение - "if-needed", известно значение - "if-added", удаляется значение - "if-removed"), а обратная аргументация срабатывает путем запуска присоединенных процедур при обращении к неизвестным значениям атрибутов. Фреймовые модели позволяют более гибко комбинировать

прямой и обратный вывод. Пример описания фреймовой модели представления знаний для задачи заключения контрактов с поставщиками:

КОНТРАКТ						
Имя слота	Указатель	Тип	Значение	IF-NEEDED	IF-ADDED	IF-REMOVED
Super-class	U	FRAME	ROOT			
Sub-class	U	FRAME	Проект			
			Отвергнутый			
			Заклученный			
Код изделия	U	String	101	Ввести	Проектировать	
Статус	U	Boolean	Y		Заклучить	Отвергнутый
Код поставщика	U	String	123			
Срок поставки	U	Date	01.06.96			
Объем поставки	U	Real	2000			
Стоимость	U	Real	100000			

ПРОЕКТИРУЕМЫЙ КОНТРАКТ						
Имя слота	Указатель	Тип	Значение	IF-NEEDED	IF-ADDED	IF-REMOVED
Super-class	S	FRAME	Контракт			

Sub-class	-		
Код изделия	S	String 101	Выбор поставщика
Статус	S	Boolean Y	
Код поставщика	S	String 123	Надежность
Срок поставки	S Date	01.06.08	
Объем поставки	S Real	2000	
Стоимость	S	Process	Расчет

ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ РЕАЛИЗАЦИИ ЭКСПЕРТНОЙ СИСТЕМЫ

Этап реализации экспертной системы

На этапе реализации экспертной системы происходит физическое наполнение базы знаний и настройка всех программных механизмов в рамках выбранного инструментального средства, а при необходимости и допрограммирование специализированных модулей программного инструмента.

Особенности реализации экспертной системы во многом определяются характером инструментального средства, в качестве которого могут выступать программные оболочки (shells), генераторы (интегрированные среды), языки представления знаний (языки программирования). Так, оболочки имеют реализованные механизмы вывода, накопления, объяснения знаний, диалоговый компонент, что, с одной стороны, упрощает разработку программной части экспертной системы, поскольку не требуется программирование, а с другой стороны, усложняет разработку базы знаний вследствие возможного несоответствия формализма системы требованиям структуры. Использование языков представления знаний таких как: язык логического программирования PROLOG, язык функционального программирования LISP, язык объектно-ориентированного

программирования SmallTalk, язык продукционных правил OPS5 и др. повышает гибкость разрабатываемой системы и одновременно увеличивает трудоемкость разработки.

Наиболее приемлемыми инструментальными средствами для создания экспертных систем являются генераторы или интегрированные среды разработки, например, G2 (фирма Gensym, дистрибьютор фирма ArgusSoft), ART-Enterprise (фирма Inference, дистрибьютор фирма Метатехнология), GURU (фирма MDBS, дистрибьютор фирма ЦПС Тверь). Они позволяют настраивать программные средства на особенности проблемных областей, при необходимости предоставляют возможность программировать на встроенных языках четвертого поколения и осуществлять эффективный экспорт/импорт данных с другими инструментальными средствами.

Среди отечественных разработок следует отметить экспертную оболочку ЭКО (ArgusSoft) и программный комплекс SIMER - MIRAGE (Исследовательский центр искусственного интеллекта ИПС РАН), который предоставляет инструментальные средства, как автоматизации разработки, так и поддержки экспертных систем.

Инструментальные средства создания и поддержки экспертных систем являются дорогостоящими продуктами и стоят от тысяч до десятков тысяч долларов. Однако, для готовых баз знаний инструментальные средства могут поставляться в исполнительной версии (RUN-TIME) на порядок дешевле.

Цена предметно-ориентированных систем может быть на порядок выше универсальных, преимущество же их заключается в более простой адаптации к конкретной предметной области, а следовательно, и в сокращении затрат на разработку. Например, интеллектуальная система для разработки финансовых приложений Cogensys Judgment Software (Cogensys Corp) стоит 200 тыс. долл.

Алгоритм выбора инструментального средства

Среди специализированных инструментальных средств интеллектуальных систем основной удельный вес занимают экспертные системы реального времени, позволяющие динамически управлять непрерывными процессами (70% рынка).

Бесспорным лидером в разработке экспертных систем реального времени является фирма Gensym с инструментальным средством G2 (дистрибьютор в России - фирма ArgusSoft), имеющая внедрения в таких компаниях как IBM, NASA, General Electric, Nissan и др. На базе G2, в свою очередь, созданы такие проблемно-ориентированные комплексы, как GDA для решения задач диагностики, ReThink для моделирования бизнес-процессов (бизнес-реинжиниринга), NeurOnline для поддержки нейронной

сети, IPS для решения задач динамического планирования, FaultExpert для управления телекоммуникациями и др.

В процессе жизненного цикла разработки экспертной системы инструментальные средства могут сменять друг друга по мере расширения базы знаний. Так, на этапе проектирования прототипа требуется его быстрая разработка в ущерб производительности, в то время как на этапе разработки промышленной версии на первый план выходит обеспечение эффективности функционирования.

На выбор инструментальных средств экспертной системы, в основе которых лежит определенный метод представления знаний, основное влияние оказывает класс решаемых задач (проблемных областей) и соответственно характер полученной концептуальной модели, определяющий множество требований в части отображения объектов, действий над объектами, методов обработки неопределенностей, механизмов вывода (таблица 1).

Инструментальные средства, в свою очередь, характеризуются определенными возможностями по реализации этих требований. В таблице 2. представлено выполнение требований для наиболее популярных инструментальных средств (цифрой показывается номер места в упорядоченной последовательности).

Тогда сущность алгоритма выбора инструментальных средств сводится к наложению требований проблемной области к формализмам знаний на возможности инструментальных средств и определению наилучших по заданным ограничениям (таблица 3).

Таблица 1					
Описание метода представления знаний	Классы решаемых задач				
	Интер-претация	Диагностика	Прогнозирование	Проектирование	Планирование
Описание объектов:					
Предикаты	4 2	4		4	4
Правила	1 1	1	3	2	
Семант. сеть	2	4	3	2	3
Фреймы (Объекты)	3	3	2	1	1
Действия:					

Предикаты	3 2	4		4	4
Правила	1 1	1	2	2	
Сообщения	2 3	3		1	1
Процедуры	4 4	2		3	3
Неопределенность:					
Неполнота	2 4	3		1	1
Нечеткость	3 2		4	3	3
Многозначность	4	3	2	2	2
Недостоверность	1	1	1	4	4
Вывод:					
Прямая цепочка	1	3	1	3	1
Обратн. цепочка	3	1	6	6	6
Объект.ориент.	2	2	2	1	2
Гипотет. вывод	4	4	4	2	5
Обраб. времени	6	6	3	5	3
Доска объявлений	5	5	5	4	4

Таблица 2						
Описание метода представления знаний	Программные инструментальные средства					
	ЭКО ArgusSoft	GURU MDBS	Nexpert Object Neuron Data	LEVEL- 5 Level Co	ART Enterprise Inference	G2 Gensym
Описание объектов:						
Семант. сеть					*	
Объекты (Фреймы)		*		*	**	
Предикаты					*	
Правила	*	*		*	**	
Действия:						

Правила	*	*	*	**
Сообщения		*	*	*
Предикаты				*
Процедуры	*	*	*	**
Неопределенность:				
Наслед. свойств		*	*	**
Кэф. уверен.		*	*	* *
Неч. множества		*	*	*
Усл. вероятн.	*			
Вывод:				
Прямая цепочка	*	*	*	**
Обратн. цепочка	**	*	*	**
Объект. ориент.		*	*	**
Гипотет. вывод				* *
Обраб. времени			*	*
Доска объявл.				*

Таблица 3						
Классы решаемых задач	Программные инструментальные средства					
	ЭКО	GURU	Nexpert Object	LEVEL-5	ART Enterprise	G2
Интерпретация	3	1	1	1	2	3
Диагностика	1	2	2	2	3	2
Прогнозирование	2	3	4	3	4	4
Проектирование	-	-	3	5	1	5
Планирование	-	-	5	4	5	1

Этап тестирования оценивает экспертную систему с позиции двух основных групп критериев: точности и полезности. Точность работы: правильность заключений, адекватность базы знаний проблемной области, соответствие методов решения проблемы экспертным. Полезность: ответы на

запросы пользователя; удобство интерфейса; объяснение получаемых результатов; надежность, адаптируемость, производительность и стоимость эксплуатации.

Этап внедрения и опытной эксплуатации - это переход от тестовых примеров к решению реальных задач.

Реализация экспертных систем экономического анализа деятельности предприятия

Особенности экспертных систем экономического анализа

Архитектура экспертной системы экономического анализа (особенности формирования базы знаний, выбора методов логического вывода, пользовательского интерфейса) во многом зависит от целей и глубины анализа: внешнего (для сторонних организаций) или внутреннего (для самого предприятия).

Целью внешнего анализа предприятия является определение общего состояния предприятия, т.е. интерпретация его экономического положения с точки зрения выявления возможностей эффективного взаимодействия с ним внешних организаций. Таким анализом занимаются банки при выдаче кредитов, инвесторы при размещении своего капитала, фирмы-партнеры при осуществлении закупочно-сбытовой или подрядной деятельности.

Наиболее зарекомендовавшим себя методом внешнего анализа, интегрирующим множество различных экономических показателей предприятия, служит рейтинговый метод, который формирует "снизу-вверх" интегральную оценку финансового состояния предприятия.

Примером экспертной системы внешнего анализа является система оценки кредитоспособности предприятия EvEnt, в которой общая оценка кредитоспособности суммируется из оценок отдельных факторов с учетом их весовой значимости на общую оценку по формуле:

$$(OW_{ijj} = \dots) O_{ij}, \text{ где}$$

O_{ij} - оценка влияния j - го фактора на i - й вышестоящий фактор по некоторой числовой шкале, а W_{ij} - вес (коэффициент) влияния j - го фактора на i - й фактор.

Правила базы знаний оценивают отдельные факторы, реализуя так называемый дизъюнктивный (независимый) подход к построению правил. Примеры правил имеют следующий вид:

IF: Управление="удовлетворительно"
THEN: Фин.состояние+="удовлетворительно" cf 40
IF: Финансовая структура="удовлетворительно"
THEN: Фин.состояние+="удовлетворительно" cf 60
IF: Ресурсы="удовлетворительно"
THEN: Фин.состояние+="удовлетворительно" cf 50
.....
IF: Качество управления="удовлетворительно"
THEN: Управление+="удовлетворительно" cf 80
IF: Структура управления="удовлетворительно"
THEN: Управление+="удовлетворительно" cf 90
и т.д.

В качестве весов в данном примере используются факторы уверенности, поэтому вместо формулы при разработке аналогичной системы может применяться формула объединения факторов уверенности для дизъюнкции.

В результате внедрения системы EvEnt для 80 % ситуаций решения формируются без экспертов. Если раньше на оценку предприятия экспертом банка требовалось в среднем 2-3 недели, то после внедрения экспертной системы основные затраты стали связываться со сбором и вводом исходных данных в течение 2-3 дней, а собственно оценка предприятия занимает порядка 20 минут. При этом стоимость экспертизы в среднем сократилась с 10000 долларов до 1000 долларов.

Метод классификации ситуаций

В качестве метода внешнего анализа может применяться также *метод классификации ситуаций*, когда по множеству признаков классификации, в качестве которых в данном случае выступает множество показателей деятельности предприятия, последовательно строится дерево решений, отражающее эту классификацию. В случае индуктивного вывода дерево решений строится по обучающей выборке автоматически. Пример классифицирующего дерева решений для оценки кредитоспособности предприятий, построенного в системе индуктивного вывода ИЛИС по обучающей выборке из 100 реально оцененных в одном из банков предприятий. В обучающей выборке в качестве классифицирующих признаков использовались коэффициенты автономии, мобильности, отношения собственных и заемных средств, покрытия, абсолютной ликвидности, ликвидности, а также качественные признаки репутации и величины. *Классообразующим признаком* является признак "Класс кредитоспособности" (1 - высший класс, 5 - низший класс). В результате

обобщения примеров обучающей выборки часть признаков была формально отброшена: коэффициенты мобильности, ликвидности и величина предприятия, причем по различным ветвям дерева решений наблюдалась различная последовательность классификации.

Для каждой отдельной ветки дерева решения строится правило, в котором все признаки классификации последовательно связываются в конъюнкцию (&) факторов левой части правила (так называемый конъюнктивный подход), например:

*IF: Кпокрыт. ≥ 1.55 & Кпокрыт. < 2 & Репутац=3 & Ксоб.заем. ≥ 0.625 & Ксоб.заем. < 0.75 & Кавтоном. ≥ 0.375 & Кавтоном. < 0.6
THEN: Кред.сн = 1*

Ограничения метода классификации ситуаций (конъюнктивного подхода) по сравнению с рейтинговым методом (дизъюнктивным подходом) при использовании правил принятия решений связаны с необходимостью жесткого задания всех признаков классификации по соответствующему пути дерева решения. Отсутствие хотя бы одного из признаков может привести к неудаче логического вывода.

Для внутреннего экономического анализа свойственен поиск направлений повышения эффективности деятельности предприятия, т.е. диагностика узких мест и определение рекомендаций по их устранению. В основе диагностики лежит метод последовательной декомпозиции "сверху-вниз" или дезагрегации "целое - часть", когда проблема последовательно разбивается на подпроблемы, пока на каком-либо уровне не станет ясным, какая подпроблема в действительности имеет место. Примером применения декомпозиционного метода к построению экспертных систем служит система внутреннего финансового анализа FINEX.

В случае применения экспертной системы внутреннего финансового анализа FINEX экспертиза осуществляется автоматически на основе введенных данных финансовой отчетности. При этом анализ финансовых показателей выполняется последовательно по принципу "сверху-вниз" и "слева-направо" в соответствии с деревом взаимосвязи показателей. В случае обнаружения некоторого "узкого места" (неудовлетворительного значения показателя) может быть включен диалоговый режим работы экспертной системы, в котором система последовательно опрашивает пользователя на предмет качественной оценки тех или иных процессов, причем вопросы задаются в порядке, зависящем от предыдущих ответов.

Для проведения комплексного экономического анализа предприятия целесообразно комбинировать применение описанных выше методов к построению наборов правил. В МЭСИ разработан исследовательский прототип экспертной системы "Финансовый анализ предприятий" в среде интегрированного ППП Интерэксперт(GURU), реализующий и рейтинговый, и классификационный, и декомпозиционный методы анализа.

Функциями экспертной системы финансового анализа предприятия являются:

- Ввод и проверка правильности составления бухгалтерской отчетности;
- Анализ финансового состояния предприятия;
- Анализ результатов финансово-хозяйственной деятельности предприятия и диагностика эффективности использования ресурсов.

В ходе ввода и проверки бухгалтерской отчетности осуществляется логический контроль зависимостей различных статей баланса предприятия, отчета о финансовых результатах и их использовании, справки к этому отчету и приложений к балансу. При этом правила логического контроля выполняются последовательно по декомпозиционному методу.

Анализ финансового состояния предприятия предполагает комплексную рейтинговую и классификационную оценку платежеспособности и финансовой устойчивости предприятия.

Анализ результатов финансово-хозяйственной деятельности предусматривает оценку важнейших показателей рентабельности и оборачиваемости капитала. Диагностика эффективности использования ресурсов сводится к поиску отклонений в использовании основных и оборотных средств от нормативных значений с последующей декомпозицией анализа.

Статические и динамические экспертные системы

Статические и динамические экспертные системы

В начале восьмидесятых годов в исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название "экспертные системы" (ЭС). Цель исследований по ЭС состоит в разработке программ, которые при решении задач, трудных для эксперта-человека, получают результаты, не уступающие по качеству и эффективности решениям, получаемым экспертом. Решения экспертных систем обладают "прозрачностью", т.е. могут быть объяснены пользователю на качественном уровне. Это качество экспертных систем обеспечивается их способностью рассуждать о своих знаниях и умозаключениях. ЭС способны пополнять свои знания в ходе взаимодействия с экспертом.

ЭС подразделяются на статические и динамические. Статические ЭС используются в тех приложениях, где можно не учитывать изменения окружающего мира, происходящие за время решения задачи. Статическая

система состоит из базы данных, предназначенной для хранения исходных и промежуточных данных решаемой в текущий момент задачи, базы знаний, которая хранит долгосрочные данные, описывающих рассматриваемую область (а не текущих данных), и правил, описывающих целесообразные преобразования данных этой области, решателя, формирующего такую последовательность правил, которые, будучи примененными к исходным данным, приводят к решению задачи, компонента приобретения знаний, автоматизирующего процесс наполнения ЭС знаниями, осуществляемый пользователем-экспертом, Объяснительного компонента, который объясняет, как система получила решение и Диалогового компонента, ориентированного на организацию дружественного общения с пользователем как в ходе решения задач.

В динамической ЭС учитываются изменения окружающей среды, происходящие за время решения задачи. В архитектуру этой системы по сравнению со статической ЭС вводятся два компонента: подсистема моделирования внешнего мира и подсистема связи с внешним окружением. Последняя осуществляет связи с внешним миром через систему датчиков и контроллеров. Кроме того, традиционные компоненты статической ЭС (база знаний и машина вывода) претерпевают существенные изменения, чтобы отразить временную логику происходящих в реальном мире событий.

В настоящее время технология экспертных систем используется для решения различных типов задач (интерпретация, предсказание, диагностика, планирование, конструирование, контроль, отладка, инструктаж, управление) в самых разнообразных проблемных областях, таких, как финансы, нефтяная и газовая промышленность, энергетика, транспорт, фармацевтическое производство, космос, металлургия, горное дело, химия, образование, целлюлозно-бумажная промышленность, телекоммуникации и связь и др.

Интерфейс с конечным пользователем

Система G2 предоставляет разработчику богатые возможности для формирования простого, ясного и выразительного графического интерфейса с пользователем с элементами мультипликации. Предлагаемый инструментарий позволяет наглядно отображать технологические процессы практически неограниченной сложности на разных уровнях абстракции и детализации. Кроме того, графическое отображение взаимосвязей между объектами приложения может напрямую использоваться в декларативных конструкциях языка описания знаний.

RTworks не обладает собственными средствами для отображения текущего состояния управляемого процесса. Разработчик приложения вынужден использовать систему Dataview фирмы VI Corporation, что в значительной степени ограничивает его возможности.

Интерфейс с пользователем TDC Expert ограничен возможностями системы TDC 3000, т.е. взаимодействие с конечным пользователем ограничивается текстовым режимом работы.

Представление знаний в экспертных системах

Первый и основной вопрос, который надо решить при представлении знаний, - это вопрос определения состава знаний, т.е. определение того, "ЧТО ПРЕДСТАВЛЯТЬ" в экспертной системе. Второй вопрос касается того, "КАК ПРЕДСТАВЛЯТЬ" знания. Необходимо отметить, что эти две проблемы не являются независимыми. Действительно, выбранный способ представления может оказаться непригодным в принципе либо неэффективным для выражения некоторых знаний.

По нашему мнению, вопрос "КАК ПРЕДСТАВЛЯТЬ" можно разделить на две в значительной степени независимые задачи: как организовать (структурировать) знания и как представить знания в выбранном формализме.

Стремление выделить организацию знаний в самостоятельную задачу вызвано, в частности, тем, что эта задача возникает для любого языка представления и способы решения этой задачи являются одинаковыми (либо сходными) вне зависимости от используемого формализма.

Итак, в круг вопросов, решаемых при представлении знаний, будем включать следующие:

- определение состава представляемых знаний;
- организацию знаний;
- представление знаний, т.е. определение модели представления.

Состав знаний ЭС определяется следующими факторами:

- проблемной средой;
- архитектурой экспертной системы;
- потребностями и целями пользователей;
- языком общения.

В соответствии с общей схемой статической экспертной системы для ее функционирования требуются следующие знания:

- знания о процессе решения задачи (т.е. управляющие знания), используемые интерпретатором (решателем);
- знания о языке общения и способах организации диалога, используемые лингвистическим процессором (диалоговым компонентом);

- знания о способах представления и модификации знаний, используемые компонентом приобретения знаний;
- поддерживающие структурные и управляющие знания, используемые объяснительным компонентом.

Для динамической ЭС, кроме того, необходимы следующие знания:

1. знания о методах взаимодействия с внешним окружением;
2. знания о модели внешнего мира.

Зависимость состава знаний от требований пользователя проявляется в следующем:

- какие задачи (из общего набора задач) и с какими данными хочет решать пользователь;
- каковы предпочтительные способы и методы решения;
- при каких ограничениях на количество результатов и способы их получения должна быть решена задача;
- каковы требования к языку общения и организации диалога;
- какова степень общности (конкретности) знаний о проблемной области, доступная пользователю;
- каковы цели пользователей.

Состав знаний о языке общения зависит как от языка общения, так и от требуемого уровня понимания.

С учетом архитектуры экспертной системы знания целесообразно делить на интерпретируемые и не интерпретируемые. К первому типу относятся те знания, которые способен интерпретировать решатель (интерпретатор). Все остальные знания относятся ко второму типу. Решатель не знает их структуры и содержания. Если эти знания используются каким-либо компонентом системы, то он не "осознает" этих знаний. Не интерпретируемые знания подразделяются на вспомогательные знания, хранящие информацию о лексике и грамматике языка общения, информацию о структуре диалога, и поддерживающие знания. Вспомогательные знания обрабатываются естественно-языковой компонентой, но ход этой обработки решатель не осознает, так как этот этап обработки входных сообщений является вспомогательным для проведения экспертизы. Поддерживающие знания используются при создании системы и при выполнении объяснений. Поддерживающие знания выполняют роль описаний (обоснований) как интерпретируемых знаний, так и действий системы. Поддерживающие знания подразделяются на технологические и семантические. Технологические поддерживающие знания содержат сведения о времени создания описываемых ими знаний, об авторе знаний и т.п. Семантические поддерживающие знания содержат смысловое описание этих знаний. Они

содержат информацию о причинах ввода знаний, о назначении знаний, описывают способ использования знаний и получаемый эффект. Поддерживающие знания имеют описательный характер.

Интерпретируемые знания можно разделить на предметные знания, управляющие знания и знания о представлении. Знания о представлении содержат информацию о том, каким образом (в каких структурах) в системе представлены интерпретируемые знания.

Предметные знания содержат данные о предметной области и способах преобразования этих данных при решении поставленных задач. Отметим, что по отношению к предметным знаниям знания о представлении и знания об управлении являются метазнаниями. В предметных знаниях можно выделить описатели и собственно предметные знания. Описатели содержат определенную информацию о предметных знаниях, такую, как коэффициент определенности правил и данных, меры важности и сложности. Собственно предметные знания разбиваются на факты и исполняемые утверждения. Факты определяют возможные значения сущностей и характеристик предметной области. Исполняемые утверждения содержат информацию о том, как можно изменять описание предметной области в ходе решения задач. Говоря другими словами, исполняемые утверждения - это знания, задающие процедуры обработки. Однако мы избегаем использовать термин "процедурные знания", так как хотим подчеркнуть, что эти знания могут быть заданы не только в процедурной, но и в декларативной форме.

Управляющие знания можно разделить на фокусирующие и решающие. Фокусирующие знания описывают, какие знания следует использовать в той или иной ситуации. Обычно фокусирующие знания содержат сведения о наиболее перспективных объектах или правилах, которые целесообразно использовать при проверке соответствующих гипотез. В первом случае внимание фокусируется на элементах рабочей памяти, во втором - на правилах базы знаний. Решающие знания содержат информацию, используемую для выбора способа интерпретации знаний, подходящего к текущей ситуации. Эти знания применяются для выбора стратегий или эвристик, наиболее эффективных для решения данной задачи.

Качественные и количественные показатели экспертной системы могут быть значительно улучшены за счет использования метазнаний, т.е. знаний о знаниях. Метазнания не представляют некоторую единую сущность, они могут применяться для достижения различных целей. Перечислим возможные назначения метазнаний :

1. метазнания в виде стратегических метаправил используются для выбора релевантных правил;

2. метазнания используются для обоснования целесообразности применения правил из области экспертизы;
3. метаправила используются для обнаружения синтаксических и семантических ошибок в предметных правилах;
4. метаправила позволяют системе адаптироваться к окружению путем перестройки предметных правил и функций;
5. метаправила позволяют явно указать возможности и ограничения системы, т.е. определить, что система знает, а что не знает.

Вопросы организации знаний необходимо рассматривать в любом представлении, и их решение в значительной степени не зависит от выбранного способа (модели) представления. Выделим следующие аспекты проблемы организации знаний :

- организация знаний по уровням представления и по уровням детальности;
- организация знаний в рабочей памяти;
- организация знаний в базе знаний.

Уровни представления и уровни детальности

Для того чтобы экспертная система могла управлять процессом поиска решения, была способна приобретать новые знания и объяснять свои действия, она должна уметь не только использовать свои знания, но и обладать способностью понимать и исследовать их, т.е. экспертная система должна иметь знания о том, как представлены ее знания о проблемной среде. Если знания о проблемной среде назвать знаниями нулевого уровня представления, то первый уровень представления содержит метазнания, т.е. знания о том, как представлены во внутреннем мире системы знания нулевого уровня. Первый уровень содержит знания о том, какие средства используются для представления знаний нулевого уровня. Знания первого уровня играют существенную роль при управлении процессом решения, при приобретении и объяснении действий системы. В связи с тем, что знания первого уровня не содержат ссылок на знания нулевого уровня, знания первого уровня независимы от проблемной среды.

Число уровней представления может быть больше двух. Второй Уровень представления содержит сведения о знаниях первого уровня, т.е. знания о представлении базовых понятий первого уровня. Разделение знаний по уровням представления обеспечивает расширение области применимости системы.

Выделение уровней детальности позволяет рассматривать знания с различной степенью подробности. Количество уровней детальности во многом определяется спецификой решаемых задач, объемом знаний и

способом их представления. Как правило, выделяется не менее трех уровней детальности, отражающих соответственно общую, логическую и физическую организацию знаний. Введение нескольких уровней детальности обеспечивает дополнительную степень гибкости системы, так как позволяет производить изменения на одном уровне, не затрагивая другие. Изменения на одном уровне детальности могут приводить к дополнительным изменениям на этом же уровне, что оказывается необходимым для обеспечения согласованности структур данных и программ. Однако наличие различных уровней препятствует распространению изменений с одного уровня на другие.

Организация знаний в рабочей системе

Рабочая память (РП) экспертных систем предназначена для хранения данных. Данные в рабочей памяти могут быть однородны или разделяются на уровни по типам данных. В последнем случае на каждом уровне рабочей памяти хранятся данные соответствующего типа. Выделение уровней усложняет структуру экспертной системы, но делает систему более эффективной. Например, можно выделить уровень планов, уровень агенды (упорядоченного списка правил, готовых к выполнению) и уровень данных предметной области (уровень решений).

В современных экспертных системах данные в рабочей памяти рассматриваются как изолированные или как связанные. В первом случае рабочая память состоит из множества простых элементов, а во втором - из одного или нескольких (при нескольких уровнях в РП) сложных элементов (например, объектов). При этом сложный элемент соответствует множеству простых, объединенных в единую сущность. Теоретически оба подхода обеспечивают полноту, но использование изолированных элементов в сложных предметных областях приводит к потере эффективности.

Данные в РП в простейшем случае являются константами и (или) переменными. При этом переменные могут трактоваться как характеристики некоторого объекта, а константы - как значения соответствующих характеристик. Если в РП требуется анализировать одновременно несколько различных объектов, описывающих текущую проблемную ситуацию, то необходимо указывать, к каким объектам относятся рассматриваемые характеристики. Одним из способов решения этой задачи является явное указание того, к какому объекту относится характеристика.

Если РП состоит из сложных элементов, то связь между отдельными объектами указывается явно, например заданием семантических отношений. При этом каждый объект может иметь свою внутреннюю структуру. Необходимо отметить, что для ускорения поиска и сопоставления данные в РП могут быть связаны не только логически, но и ассоциативно.

Организация знаний в базе данных

Показателем интеллектуальности системы с точки зрения представления знаний считается способность системы использовать в нужный момент необходимые (релевантные) знания. Системы, не имеющие средств для определения релевантных знаний, неизбежно сталкиваются с проблемой "комбинаторного взрыва". Можно утверждать, что эта проблема является одной из основных причин, ограничивающих сферу применения экспертных систем. В проблеме доступа к знаниям можно выделить три аспекта: связность знаний и данных, механизм доступа к знаниям и способ сопоставления.

Связность (агрегация) знаний является основным способом, обеспечивающим ускорение поиска релевантных знаний. Большинство специалистов пришли к убеждению, что знания следует организовывать вокруг наиболее важных объектов (сущностей) предметной области. Все знания, характеризующие некоторую сущность, связываются и представляются в виде отдельного объекта. При подобной организации знаний, если системе потребовалась информация о некоторой сущности, то она ищет объект, описывающий эту сущность, а затем уже внутри объекта отыскивает информацию о данной сущности. В объектах целесообразно выделять два типа связей между элементами: внешние и внутренние. Внутренние связи объединяют элементы в единый объект и предназначены для выражения структуры объекта. Внешние связи отражают взаимозависимости, существующие между объектами в области экспертизы. Многие исследователи классифицируют внешние связи на логические и ассоциативные. Логические связи выражают семантические отношения между элементами знаний. Ассоциативные связи предназначены для обеспечения взаимосвязей, способствующих ускорению процесса поиска релевантных знаний.

Основной проблемой при работе с большой базой знаний является проблема поиска знаний, релевантных решаемой задаче. В связи с тем, что в обрабатываемых данных может не содержаться явных указаний на значения, требуемые для их обработки, необходим более общий механизм доступа, чем метод прямого доступа (метод явных ссылок). Задача этого механизма состоит в том, чтобы по некоторому описанию сущности, имеющемуся в рабочей памяти, найти в базе знаний объекты, удовлетворяющие этому описанию. Очевидно, что упорядочение и структурирование знаний могут значительно ускорить процесс поиска.

Нахождение желаемых объектов в общем случае уместно рассматривать как двухэтапный процесс. На первом этапе, соответствующем процессу выбора по ассоциативным связкам, совершается предварительный выбор в базе знаний потенциальных кандидатов на роль желаемых объектов. На

втором этапе путем выполнения операции сопоставления потенциальных кандидатов с описаниями кандидатов осуществляется окончательный выбор искомым объектам. При организации подобного механизма доступа возникают определенные трудности: Как выбрать критерий пригодности кандидата? Как организовать работу в конфликтных ситуациях? и т.п.

Операция сопоставления может использоваться не только как средство выбора нужного объекта из множества кандидатов; она может быть использована для классификации, подтверждения, декомпозиции и коррекции. Для идентификации неизвестного объекта он может быть сопоставлен с некоторыми известными образцами. Это позволит классифицировать неизвестный объект как такой известный образец, при сопоставлении с которым были получены лучшие результаты. При поиске сопоставление используется для подтверждения некоторых кандидатов из множества возможных. Если осуществлять сопоставление некоторого известного объекта с неизвестным описанием, то в случае успешного сопоставления будет осуществлена частичная декомпозиция описания.

Операции сопоставления весьма разнообразны. Обычно выделяют следующие их формы: синтаксическое, параметрическое, семантическое и принуждаемое сопоставления. В случае синтаксического сопоставления соотносят формы (образцы), а не содержание объектов. Успешным является сопоставление, в результате которого образцы оказываются идентичными. Обычно считается, что переменная одного образца может быть идентична любой константе (или выражению) другого образца. Иногда на переменные, входящие в образец, накладывают требования, определяющие тип констант, с которыми они могут сопоставляться. Результат синтаксического сопоставления является бинарным: образцы сопоставляются или не сопоставляются. В параметрическом сопоставлении вводится параметр, определяющий степень сопоставления. В случае семантического сопоставления соотносятся не образцы объектов, а их функции. В случае принуждаемого сопоставления один сопоставляемый образец рассматривается с точки зрения другого. В отличие от других типов сопоставления здесь всегда может быть получен положительный результат. Вопрос состоит в силе принуждения. Принуждение могут выполнять специальные процедуры, связываемые с объектами. Если эти процедуры не в состоянии осуществить сопоставление, то система сообщает, что успех может быть достигнут только в том случае, если определенные части рассматриваемых сущностей можно считать сопоставляющимися.

Методы поиска решений в экспертных системах

Методы решения задач, основанные на сведениях их к поиску, зависят от психодиагностики в психосоматике, а также другие системы. особенностей предметной области, в которой решается задача, и от требований,

предъявляемых пользователем к решению. Особенности предметной области с точки зрения методов решения можно характеризовать следующими параметрами:

- размер, определяющий объем пространства, в котором предстоит искать решение;
- изменяемость области, характеризует степень изменяемости области во времени и пространстве (здесь будем выделять статические и динамические области);
- полнота модели, описывающей область, характеризует адекватность модели, используемой для описания данной области. Обычно если модель не полна, то для описания области используют несколько моделей, дополняющих друг друга за счет отражения различных свойств предметной области;
- определенность данных о решаемой задаче, характеризует степень точности (ошибочности) и полноты (неполноты) данных. Точность (ошибочность) является показателем того, что предметная область с точки зрения решаемых задач описана точными или неточными данными; под полнотой (неполнотой) данных понимается достаточность (недостаточность) входных данных для однозначного решения задачи.

Требования пользователя к результату задачи, решаемой с помощью поиска, можно характеризовать количеством решений и свойствами результата и (или) способом его получения. Параметр "количество решений" может принимать следующие основные значения: одно решение, несколько решений, все решения. Параметр "свойства" задает ограничения, которым должен удовлетворять полученный результат или способ его получения. Так, например, для системы, выдающей рекомендации по лечению больных, пользователь может указать требование не использовать некоторое лекарство (в связи с его отсутствием или в связи с тем, что оно противопоказано данному пациенту). Параметр "свойства" может определять и такие особенности, как время решения ("не более чем", "диапазон времени" и т.п.), объем памяти, используемой для получения результата, указание об обязательности (невозможности) использования каких-либо знаний (данных) и т.п.

Итак, сложность задачи, определяемая вышеприведенным набором параметров, варьируется от простых задач малой размерности с неизменяемыми определенными данными и отсутствием ограничений на результат и способ его получения до сложных задач большой размерности с изменяемыми, ошибочными и неполными данными и произвольными ограничениями на результат и способ его получения. Из общих соображений ясно, что каким-либо одним методом нельзя решить все задачи. Обычно одни

методы превосходят другие только по некоторым из перечисленных параметров.

Рассмотренные ниже методы могут работать в статических и динамических проблемных средах. Для того чтобы они работали в условиях динамики, необходимо учитывать время жизни значений переменных, источник данных для переменных, а также обеспечивать возможность хранения истории значений переменных, моделирования внешнего окружения и оперирования временными категориями в правилах.

Существующие методы решения задач, используемые в экспертных системах, можно классифицировать следующим образом:

- методы поиска в одном пространстве - методы, предназначенные для использования в следующих условиях: области небольшой размерности, полнота модели, точные и полные данные;
- методы поиска в иерархических пространствах - методы, предназначенные для работы в областях большой размерности;
- методы поиска при неточных и неполных данных ;
- методы поиска, использующие несколько моделей, предназначенные для работы с областями, для адекватного описания которых одной модели недостаточно.

Предполагается, что перечисленные методы при необходимости должны объединяться для того, чтобы позволить решать задачи сложность которых возрастает одновременно по нескольким параметрам.

Инструментальный комплекс для создания статических экспертных систем

Рассмотрим особенности инструментальных средств для создания статических ЭС на примере комплекса ЭКО, разработанного в РосНИИ ИТ и АП. Наиболее успешно комплекс применяется для создания ЭС, решающих задачи диагностики (технической и медицинской), эвристического оценивания (риска, надежности и т.д.), качественного прогнозирования, а также обучения.

Комплекс ЭКО используется: для создания коммерческих и промышленных экспертных систем на персональных ЭВМ, а также для быстрого создания прототипов экспертных систем с целью определения применимости методов инженерии знаний в некоторой конкретной проблемной области.

На основе комплекса ЭКО было разработано более 100 прикладных экспертных систем. Среди них отметим следующие:

- поиск одиночных неисправностей в персональном компьютере;
- оценка состояния гидротехнического сооружения (Чарвакская ГЭС);
- подготовка деловых писем при ведении переписки с зарубежными партнерами;
- проведение скрининговой оценки иммунологического статуса;
- оценка показаний микробиологического обследования пациента, страдающего неспецифическими хроническими заболеваниями легких.

Средства представления знаний и стратегии управления

Комплекс ЭКО включает три компонента.

Ядром комплекса является интегрированная оболочка экспертных систем ЭКО, которая обеспечивает быстрое создание эффективных приложений для решения задач анализа в статических проблемных средах типа 1 и 2.

При разработке средств представления знаний оболочки преследовались две основные цели: эффективное решение достаточно широкого и практически значимого класса задач средствами персональных компьютеров; гибкие возможности по описанию пользовательского интерфейса и проведению консультации в конкретных приложениях. При представлении знаний в оболочке используются специализированные (частные) - утверждения типа "атрибут - значение" и частные правила, что позволяет исключить ресурсоемкую операцию сопоставления по образцу и добиться эффективности разрабатываемых приложений. Выразительные возможности оболочки удалось существенно расширить за счет интегрированности, обеспечиваемой путем вызова внешних программ через сценарий консультации и стыковки с базами данных (ПИРС и dBase IV) и внешними программами. В оболочке ЭКО обеспечивается слабая структуризация БЗ за счет ее разделения на отдельные компоненты - для решения отдельных подзадач в проблемной среде - модели (понятию "модель" ЭКО соответствует понятие "модуль" базы знаний системы G2).

С точки зрения технологии разработки ЭС оболочка поддерживает подходы, основанные на поверхностных знаниях и структурировании процесса решения.

Оболочка функционирует в двух режимах: в режиме приобретения знаний и в режиме консультации (решения задач). В первом режиме разработчик ЭС средствами диалогового редактора вводит в БЗ описание конкретного приложения в терминах языка представления знаний оболочки. Это описание компилируется в сеть вывода с прямыми адресными ссылками на конкретные утверждения и правила. Во втором режиме оболочка решает

конкретные задачи пользователя в диалоговом или пакетном режиме. При этом решения выводятся от целей к данным (обратное рассуждение).

Для расширения возможностей оболочки по работе с глубинными знаниями комплекс ЭКО может быть дополнен компонентом К-ЭКО (конкретизатором знаний), который позволяет описывать закономерности в проблемных средах в терминах общих (абстрактных) объектов и правил. К-ЭКО используется на этапе приобретения знаний вместо диалогового редактора оболочки для преобразования общих описаний в конкретные сети вывода, допускающие эффективный вывод решений средствами оболочки ЭКО. Таким образом, использование конкретизатора обеспечивает возможность работы с проблемными средами типа 2.

Третий компонент комплекса - система ИЛИС, позволяющая создавать ЭС в статических проблемных средах за счет индуктивного обобщения данных (примеров) и предназначенная для использования в тех приложениях, где отсутствие правил, отражающих закономерности в проблемной среде, возмещается обширным экспериментальным материалом. Система ИЛИС обеспечивает автоматическое формирование простейших конкретных правил и автономное решение задач на их основе; при этом используется жесткая схема диалога с пользователем. Поскольку при создании реальных приложений эксперты представляют, как правило, и знания о закономерностях в проблемной среде, и экспериментальный материал (для решения частных подзадач), возникает необходимость в использовании правил, сформированных системой ИЛИС, в рамках более сложных средств представления знаний. Комплекс ЭКО обеспечивает автоматический перевод таких правил в формат оболочки ЭКО. В результате удается получить полное (адекватное) представление реальной проблемной среды, кроме того, задать гибкое описание организации взаимодействия ЭС с конечным пользователем.

Инструментальный комплекс для создания экспертных систем реального времени

История развития ИС для создания ЭС реального времени началась в 1985 г., когда фирма Lisp Machine Inc. выпустила систему Pison для символьных ЭВМ Symbolics. Успех этого ИС привел к тому, что группа ведущих разработчиков Pison в 1986 г. образовала частную фирму Gensym, которая, значительно развив идеи, заложенные в Pison, в 1988 г. вышла на рынок с ИС под названием G2, версия 1.0. В настоящее время функционирует версия 4.2 и готовится к выпуску версия 5.0.

Основное предназначение программных продуктов фирмы Gensym (США) - помочь предприятиям сохранять и использовать знания и опыт их наиболее талантливых и квалифицированных сотрудников в интеллектуальных системах реального времени, повышающих качество

продукции, надежность и безопасность производства и снижающих производственные издержки. О том, как фирме Gensym удается справиться с этой задачей, говорит хотя бы то, что сегодня ей принадлежат 50% мирового рынка экспертных систем, используемых в системах управления.

С отставанием от Gensym на 2 - 3 года другие фирмы начали создавать свои ИС для ЭС РВ. С точки зрения независимых экспертов NASA, проводивших комплексное исследование характеристик и возможностей некоторых из перечисленных систем, в настоящее время наиболее продвинутым ИС, безусловно, остается G2 (Gensym, США); следующие места со значительным отставанием (реализовано менее 50% возможностей G2) занимают RTWorks - фирма Talarian (США), COMDALE/C (Comdale Techn. - Канада), COGSYS (SC - США), ILOG Rules (ILOG - Франция).

Классы задач, для которых предназначена G2 и подобные ей системы:

- мониторинг в реальном масштабе времени;
- системы управления верхнего уровня;
- системы обнаружения неисправностей;
- диагностика;
- составление расписаний;
- планирование;
- оптимизация;
- системы - советчики оператора;
- системы проектирования.

Инструментальные средства фирмы Gensym являются эволюционным шагом в развитии традиционных экспертных систем от статических предметных областей к динамическим. Немалую долю успеха фирме Gensym обеспечивают основные принципы, которых она придерживается в своих новых разработках:

- проблемно/предметная ориентация;
- следование стандартам;
- независимость от вычислительной платформы;
- совместимость снизу-вверх с предыдущими версиями;
- универсальные возможности, не зависящие от решаемой задачи;
- обеспечение технологической основы для прикладных систем;
- комфортная среда разработки;
- поиск новых путей развития технологии;
- распределенная архитектура клиент-сервер;
- высокая производительность.

Основным достоинством оболочки экспертных систем G2 для российских пользователей является возможность применять ее как интегрирующий компонент, позволяющий за счет открытости интерфейсов и

поддержки широкого спектра вычислительных платформ легко объединить уже существующие, разрозненные средства автоматизации в единую комплексную систему управления, охватывающую все аспекты производственной деятельности - от формирования портфеля заказов до управления технологическим процессом и отгрузки готовой продукции. Это особенно важно для отечественных предприятий, парк технических и программных средств которых формировался по большей части бессистемно, под влиянием резких колебаний в экономике.

Кроме системы G2, как базового средства разработки, фирма Gensym предлагает комплекс проблемно/предметно-ориентированных расширений для быстрой реализации сложных динамических систем на основе специализированных графических языков, включающих параметризуемые операторные блоки для представления элементов технологического процесса и типовых задач обработки информации. Набор инструментальных средств фирмы Gensym, сгруппированный по проблемной ориентации, охватывает все стадии производственного процесса и выглядит следующим образом:

- интеллектуальное управление производством - G2, G2 Diagnostic Assistant (GDA), NeurOn-Line (NOL), Statistical Process Control (SPC), BatchDesign_Kit;
- оперативное планирование - G2, G2 Scheduling Toolkit (GST), Dynamic Scheduling Packadge (DSP);
- разработка и моделирование производственных процессов - G2, ReThink, BatchDesign_Kit;
- управление операциями и корпоративными сетями - G2, Fault Expert.

Несмотря на то, что первая версия системы G2 появилась не так давно - в 1988 г., ее даже в богатой Америке никто не назовет дешевой. G2 можно назвать бестселлером на рынке программных продуктов - на начало 1996 г. в мире было установлено более 5000 ее копий. Фирма Gensym обслуживает более 30 отраслей - от аэрокосмических исследований до производства пищевых продуктов. Список пользователей G2 выглядит как справочник Who-Is-Who в мировой промышленности. 25 самых крупных индустриальных мировых корпораций используют G2. На базе G2 написано более 500 действующих приложений.

Чем же объясняется успех инструментального комплекса G2? Прежде всего G2 - динамическая система в полном смысле этого слова. G2 - это объектно-ориентированная интегрированная среда для разработки и сопровождения приложений реального времени, использующих базы знаний. G2 функционирует на большинстве существующих платформ. База знаний G2 сохраняется в обычном ASCII-файле, который однозначно интерпретируется на любой из поддерживаемых платформ. Перенос приложения не требует его перекомпиляции и заключается в простом переписывании файлов. Функциональные возможности и внешний вид приложения не претерпевают при этом никаких изменений.

Оглавление

Искусственный интеллект - основа новых информационных технологий	1
Введение в искусственный интеллект	2
Основные направления исследований в области искусственного интеллекта	3
Классификация интеллектуальных информационных систем	5
Системы с интеллектуальным интерфейсом	6
Экспертные системы	8
Самообучающиеся системы	10
Нейронные сети	11
Информационные хранилища	12
Адаптивные информационные системы	13
ЭТАПЫ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ	13
Этап 1: Выбор подходящей проблемы	14
Этап 2: Разработка прототипной системы	17
Этап 3: Развитие прототипа до промышленной ЭС	20
Этап 4: Оценка системы	21
Этап 5: Стыковка системы	22
Этап 6: Поддержка системы	23
КЛАССЫ ЭКСПЕРТНЫХ СИСТЕМ	23
ТЕХНОЛОГИЯ СОЗДАНИЯ ЭКСПЕРТНЫХ СИСТЕМ -	27
Этапы создания экспертной системы	27
Идентификация проблемной области	32
МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ	34
Построение концептуальной модели	34
ФОРМАЛИЗАЦИЯ БАЗЫ ЗНАНИЙ	38
Логическая модель представления знаний	39
Продукционные модели представления знаний	40
Динамические модели представления знаний	44
Семантические сети - представление знаний	45
Фреймы - представление знаний	46

ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ РЕАЛИЗАЦИИ ЭКСПЕРТНОЙ СИСТЕМЫ.....	48
Этап реализации экспертной системы.....	48
Алгоритм выбора инструментального средства	49
Реализация экспертных систем экономического анализа деятельности предприятия.....	53
Особенности экспертных систем экономического анализа.....	53
Метод классификации ситуаций	54
Статические и динамические экспертные системы	56
Статические и динамические экспертные системы	56
Интерфейс с конечным пользователем	57
Представление знаний в экспертных системах.....	58
Уровни представления и уровни детальности	61
Организация знаний в рабочей системе	62
Организация знаний в базе данных	63
Методы поиска решений в экспертных системах	64
Инструментальный комплекс для создания статических экспертных систем	66
Средства представления знаний и стратегии управления	67
Инструментальный комплекс для создания экспертных систем реального времени	68