

"Базы знаний и экспертные системы"

Конспект лекции по курсу: "Базы знаний и экспертные системы"

Экспертные системы играют важную роль в прогнозировании и управлении экономикой. Овладение методологией экспертных систем помогает принять решение в самых сложных и уникальных ситуациях. Чтобы уметь использовать ЭС на практике, важно знать об основных классах систем, типах данных и способах их хранения, основных стадиях разработки и реализации экспертных систем.

Мы публикуем полный курс лекций по дисциплине "Базы знаний и экспертные системы". Курс написан кандидатом технических наук Р.С.Кушниренко. Материал содержит подробную и доступную информацию о экспертных системах. Он будет полезен в самостоятельном изучении дисциплины, при выполнении лабораторных работ и подготовке к зачету.

№1. Введение

Цели и задачи дисциплины

По мере развития рыночных отношений в экономике России все большее значение приобретают проблемы принятия эффективных управленческих решений. Особенности экономики в наши дни являются ее крупномасштабность, многогранность, изменчивость и трудность прогнозирования по накопленным данным. Эти факторы в немалой степени обусловили проникновение искусственного интеллекта, и в частности экспертных систем (ЭС), в решение экономических задач. Интерес к ЭС вызван, по крайней мере, тремя причинами.

- 1) Они ориентированы на решение широкого круга неформальных задач, решение которых до недавнего времени считалось малодоступным для ЭВМ.
- 2) С помощью ЭС специалисты, не знающие программирования, могут самостоятельно разрабатывать интересующие их приложения.
- 3) С ЭС иногда удается достичь результатов, не уступающих возможностям экспертов.

Поэтому, главной задачей курса является знакомство с возможностями ЭС и овладение методологией их применения в решении практических задач

В результате изучения дисциплины студенты должны: знать основные классы экспертных систем и их возможности в решении практических экономических задач; типы знаний, способы их получения и методы их хранения и обработки; иметь представление об основных стадиях разработки ЭС и примерах их реализации.

ЭС как ИИС

ИИ – отрасль информатики. Спец-ты в области ИИ стремились разработать программы для ЭВМ, которые могли бы в некотором смысле «думать», т.е. решать задачи таким способом, который мы бы сочли разумным, если бы его применил человек. ИИ – программная система, имитирующая мышление человека.

В рамках ИИ разраб. ИИС.

Для интеллектуальных информационных систем, ориентированных на генерацию алгоритмов решения задач, характерны следующие признаки:

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

Коммуникативные способности ИИС характеризуют способ взаимодействия (интерфейса) конечного пользователя с системой, в частности, возможность формулирования произвольного запроса в диалоге с ИИС на языке, максимально приближенном к естественному.

Сложные плохо формализуемые задачи - это задачи, которые требуют построения оригинального алгоритма решения в зависимости от конкретной ситуации, для которой могут быть характерны неопределенность и динамичность исходных данных и знаний.

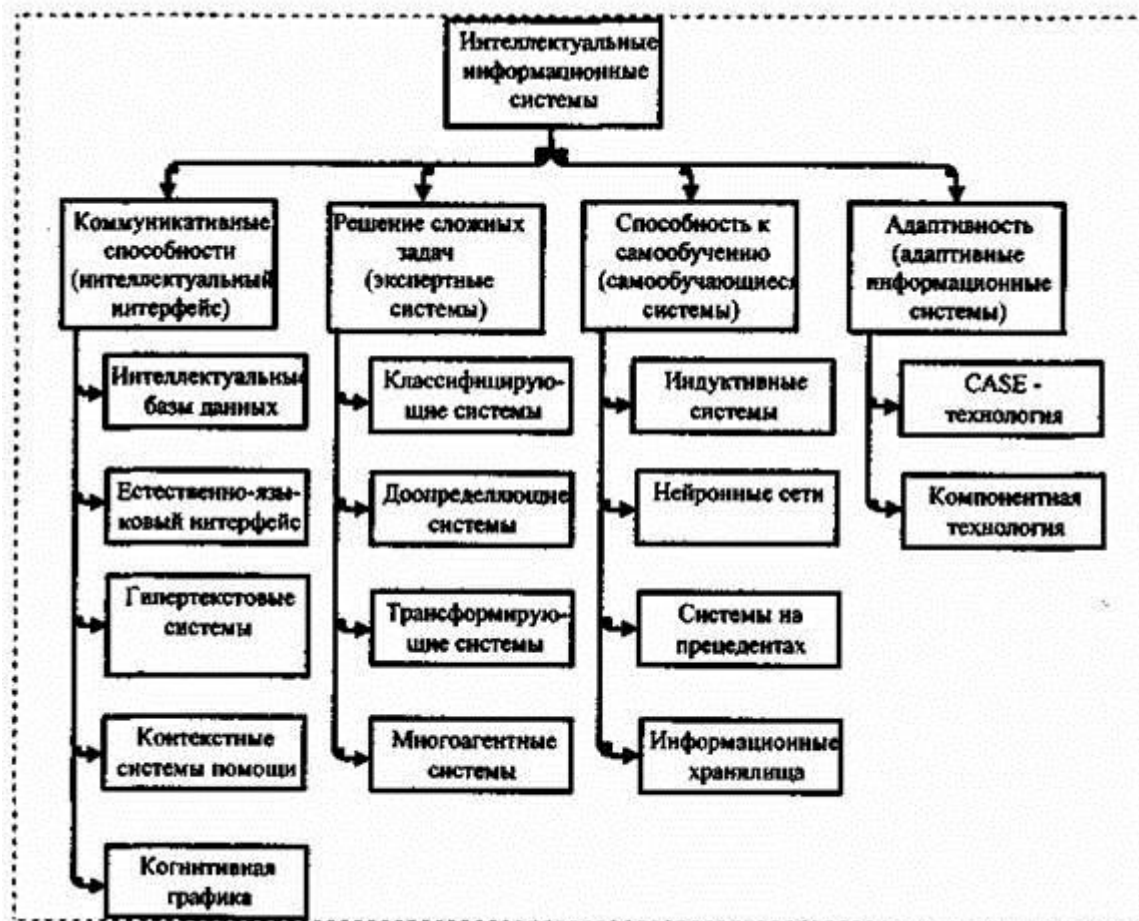
Способность к самообучению - это возможность автоматического извлечения знаний для решения задач из накопленного опыта конкретных ситуаций.

Адаптивность - способность к развитию системы в соответствии с объективными изменениями модели проблемной области.

В различных ИИС перечисленные признаки интеллектуальности развиты в неодинаковой степени и редко, когда все четыре признака реализуются одновременно. Условно каждому из признаков интеллектуальности соответствует свой класс ИИС:

- Системы с интеллектуальным интерфейсом;
- Экспертные системы;
- Самообучающиеся системы;
- Адаптивные системы.

Эксперт – это человек, который благодаря обучению и опыту может делать то, что остальные делать не могут. Они компетентны в некоторой ПО, т.е. обладают практически применимыми знаниями. Экспертные системы моделируют экспертизу, т.е. процесс принятия решения экспертом.



Назначение экспертных систем

Заключается в решении достаточно трудных для экспертов задач на основе накапливаемой базы знаний, отражающей опыт работы экспертов в рассматриваемой проблемной области. Достоинство применения экспертных систем заключается в возможности принятия решений в уникальных ситуациях, для которых алгоритм заранее не известен и формируется по исходным данным в виде цепочки рассуждений (правил принятия решений) из базы знаний. Причем решение задач предполагается осуществлять в условиях неполноты, недостоверности, многозначности исходной информации и качественных оценок процессов.

Характеристики экспертной системы

Сердцевину экспертной системы составляет база знаний, которая накапливается в процессе ее построения. Знания выражены в явном виде и организованы так, чтобы упростить принятие решений.

- Накопление и организация знаний - одна из самых важных характеристик экспертной системы. Последствия этого факта выходят за пределы построения программы, предназначенной для решения некоторого класса задач. Причина в том, что знания - основа экспертных систем - являются явными и доступными, что и отличает эти системы от большинства традиционных программ.

- Наиболее полезной характеристикой экспертной системы является то, что она применяет для решения проблем высококачественный опыт. Этот опыт может представлять уровень мышления наиболее квалифицированных экспертов в данной области, что ведет к решениям

творческим, точным и эффективным. Именно высококачественный опыт в сочетании с умением его применять, делает систему рентабельной, способной заслужить признание на рынке. Этому способствует также гибкость системы. Система может наращиваться постепенно в соответствии с нуждами бизнеса или заказчика. Это означает, что можно вначале вложить сравнительно скромные средства, а потом наращивать ее возможности по мере необходимости.

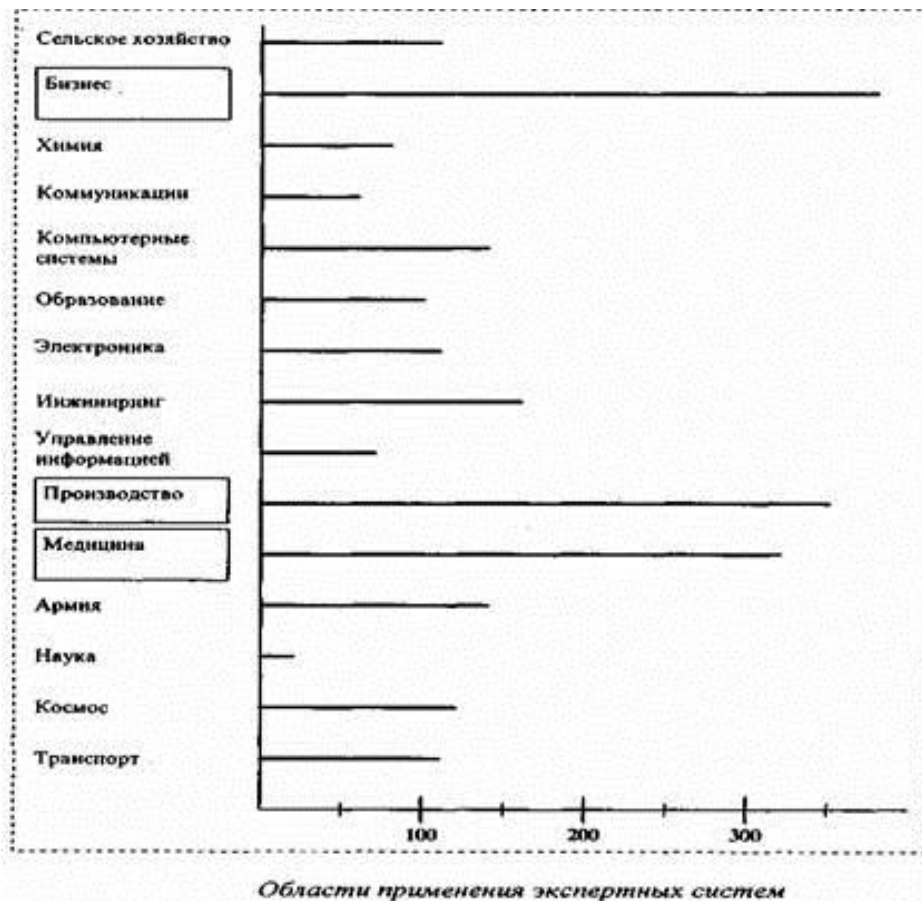
- Другой полезной чертой экспертных систем является наличие у них прогностических возможностей. Экспертная система может функционировать в качестве системы обработки информации или модели решения задачи в заданной области, давая ожидаемые ответы в конкретной ситуации и показывая, как изменятся эти ответы в новых ситуациях. Экспертная система может объяснить подробно, каким образом новая ситуация привела к изменениям. Это позволяет пользователю оценить возможное влияние новых фактов или информации и понять, как они связаны с решением. Аналогично, пользователь может оценить влияние новых стратегий или процедур на решение, добавляя новые правила или изменяя уже существующие.

- База знаний, определяющая компетентность экспертной системы, может также обеспечить новое качество: институциональную память. Если база знаний разработана в ходе взаимодействия с ведущими специалистами учреждения, отдела или штаба, то она представляет текущую политику или способы действия этой группы людей. Этот набор знаний становится сводом очень квалифицированных мнений и постоянно обновляющимся справочником наилучших стратегий и методов, используемых персоналом. Ведущие специалисты уходят, но их опыт остается. Это важно для деловой сферы и особенно ценно для вооруженных сил и правительственных органов с их частыми преобразованиями и персональными перемещениями.

-И последним важным свойством экспертных систем является то, что их можно использовать для обучения и тренировки руководящих работников и ведущих специалистов. Экспертные системы могут быть разработаны с расчетом на подобный процесс обучения, так как они уже содержат необходимые знания и способны объяснить процесс своего рассуждения. Необходимо только добавить программное обеспечение, поддерживающее соответствующий требованиям эргономики интерфейс между обучаемым и экспертной системой. Знания о методах обучения и возможном поведении пользователя также должны быть включены. В качестве инструмента обучения экспертная система обеспечивает новых служащих обширным багажом опыта и стратегий, по которым можно изучать рекомендуемую политику и методы. Систему можно также адаптировать для обучения новичков конкретным заданиям.

Применение ЭС

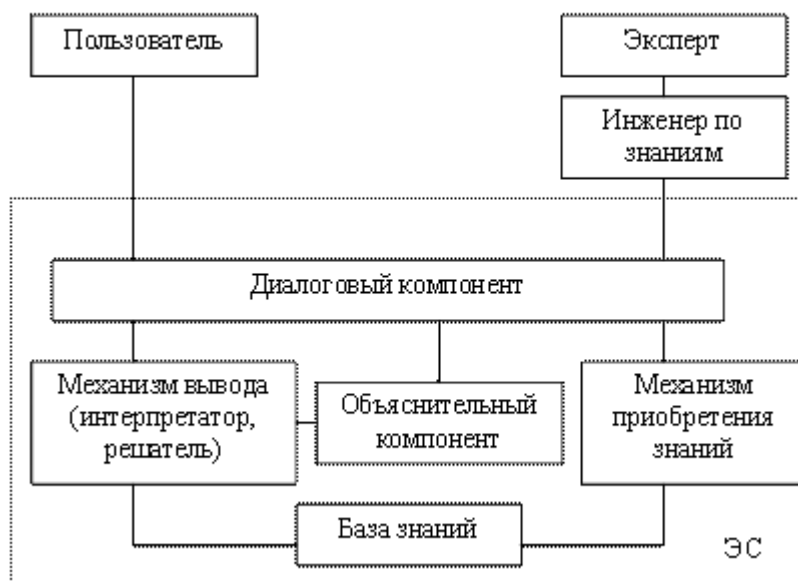
Экспертные системы используются во многих областях, среди которых лидирует сегмент приложений в бизнесе.



Архитектура ЭС

Архитектура экспертной системы включает в себя два основных компонента: базу знаний (хранилище единиц знаний) и программный инструмент доступа и обработки знаний, состоящий из механизмов вывода заключений (решения), приобретения знаний, объяснения получаемых результатов и интеллектуального интерфейса. Причем центральным компонентом экспертной системы является база знаний, которая выступает по отношению к другим компонентам как содержательная подсистема, составляющая основную ценность. "Know-how" базы знаний хорошей экспертной системы оценивается в сотни тысяч долларов, в то время как программный инструментарий - в тысячи или десятки тысяч долларов.

Архитектура ЭС



База знаний - это совокупность единиц знаний, которые представляют собой формализованное с помощью некоторого метода представления знаний отражение объектов проблемной области и их взаимосвязей, действий над объектами и, возможно, неопределенностей, с которыми эти действия осуществляются.

В качестве методов представления знаний чаще всего используются правила. Так, правила представляют собой конструкции:

Если <условие >

То <заключение> CF (Фактор определенности) <значение>

В качестве факторов определенности (CF), как правило, выступают либо условные вероятности байесовского подхода (от 0 до 1), либо коэффициенты уверенности нечеткой логики (от 0 до 100). Примеры правил имеют следующий вид:

Правило 1: Если Коэффициент рентабельности > 0.2

То Рентабельность = "удовл." CF 100

Правило 2: Если Задолженность = "нет" и Рентабельность = "удовл."

То Финансовое_сост. = "удовл." CF 80

Правило 3: Если Финансовое_сост. = "удовл." и Репутация="удовл."

То Надежность предприятия = "удовл." CF 90

Интеллектуальный интерфейс. Обмен данными между конечным пользователем и ЭС выполняет программа интеллектуального интерфейса, которая воспринимает сообщения пользователя и преобразует их в форму представления базы знаний и, наоборот, переводит внутреннее представление результата обработки в формат пользователя и выдает сообщение на требуемый носитель. Важнейшим требованием к организации диалога пользователя с ЭС

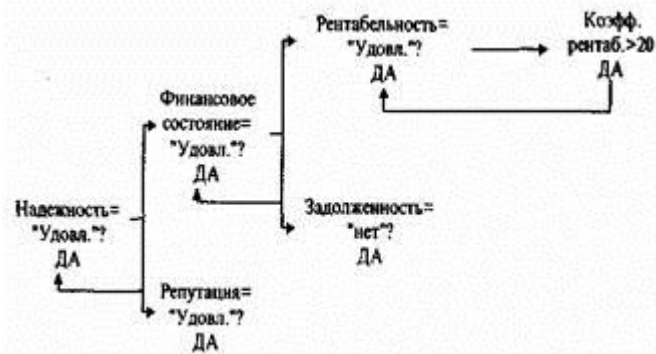
является естественность, которая не означает буквально формулирование потребностей пользователя предложениями естественного языка, хотя это и не исключается в ряде случаев. Важно, чтобы последовательность решения задачи была гибкой, соответствовала представлениям пользователя и велась в профессиональных терминах.

Механизм вывода. Этот программный инструмент получает от интеллектуального интерфейса преобразованный во внутреннее представление запрос, формирует из базы знаний конкретный алгоритм решения задачи, выполняет алгоритм, а полученный результат предоставляется интеллектуальному интерфейсу для выдачи ответа на запрос пользователя.

В основе использования любого механизма вывода лежит процесс нахождения в соответствии с поставленной целью и описанием конкретной ситуации (исходных данных), относящихся к решению единиц знаний (правил, объектов, прецедентов и т.д.) и связыванию их при необходимости в цепочку рассуждений, приводящую к определенному результату. Для представления знаний в форме правил это может быть прямая или обратная цепочка рассуждений.



Прямая цепочка рассуждений



Обратная цепочка рассуждений

Механизм объяснения. В процессе или по результатам решения задачи пользователь может запросить объяснение или обоснование хода решения. С этой целью ЭС должна предоставить соответствующий механизм объяснения. Объяснительные способности ЭС определяются возможностью механизма вывода запоминать путь решения задачи. Тогда на вопросы пользователя "Как?" и "Почему?" получено решение или запрошены те или иные данные, система всегда может выдать цепочку рассуждений до требуемой контрольной точки, сопровождая выдачу объяснения заранее подготовленными комментариями. В случае отсутствия решения задач объяснение должно выдаваться пользователю автоматически. Полезно иметь возможность и гипотетического объяснения решения задачи, когда система отвечает на вопросы, что будет в том или ином случае.

Однако не всегда пользователя может интересовать полный вывод решения, содержащий множество ненужных деталей. В этом случае система должна уметь выбирать из цепочки только ключевые моменты с учетом их важности и уровня знаний пользователя. Для этого в базе знаний необходимо поддерживать модель знаний и намерений пользователя. Если же пользователь продолжает не понимать полученный ответ, то система должна быть способна в диалоге на основе поддерживаемой модели проблемных знаний обучать пользователя тем или иным фрагментам знаний, т.е. раскрывать более подробно отдельные понятия и зависимости, если даже эти детали непосредственно в выводе не использовались.

Механизм приобретения знаний. База знаний отражает знания экспертов (специалистов) в данной проблемной области о действиях в различных ситуациях или процессах решения характерных задач. Выявлением подобных знаний и последующим их представлением в базе знаний занимаются специалисты, называемые инженерами знаний. Для ввода знаний в базу и их последующего обновления ЭС должна обладать механизмом приобретения знаний. В простейшем случае это интеллектуальный редактор, который позволяет вводить единицы знаний в базу и проводить их синтаксический и семантический контроль, например, на непротиворечивость; в более сложных случаях извлекать, знания путем специальных сценариев интервьюирования экспертов, или из вводимых примеров реальных ситуаций, как в случае индуктивного вывода, или из текстов, или из опыта работы самой интеллектуальной системы.

База данных (рабочая память) предназначена для хранения исходных и промежуточных данных решаемой в текущий момент задачи. Этот термин совпадает по названию, но не по смыслу с термином, используемым в информационно-поисковых системах (ИПС) и системах управления базами данных (СУБД) для обозначения всех данных (и в первую очередь не текущих, а долгосрочных), хранимых в системе.

Режимы работы ЭС

Экспертная система работает в двух режимах: приобретения знаний и решения задач (режим консультации или режим использования ЭС).

В режиме приобретения знания общение с ЭС осуществляет эксперт через посредничество инженера по знаниям. Эксперт описывает проблемную область в виде совокупности данных и правил. Данные определяют объекты, их характеристики и значения, существующие в области экспертизы. Правила определяют способы манипулирования данными, характерные для рассматриваемой проблемной области. Эксперт, используя компонент приобретения знаний, наполняет систему знаниями, которые позволяют ЭС в режиме решения самостоятельно (без эксперта) решать задачи из проблемной области.

Отметим, что режиму приобретения знаний при традиционном подходе к разработке программ соответствуют этапы алгоритмизации, программирования и отладки, выполняемые программистом. Таким образом, в отличие от традиционного подхода разработку программ осуществляет эксперт (с помощью ЭС), не владеющий программированием, а не программист.

В режиме консультации общение с ЭС осуществляет конечный пользователь, которого интересует результат и (или) способ получения решения.

После обработки данные поступают в РП. На основе входных данных из РП, общих данных о проблемной области и правил из БЗ решатель (интерпретатор) формирует решение задачи.

В отличие от традиционных программ ЭС в режиме решения задачи не только исполняет предписанную последовательность операций, но и предварительно формирует ее. Если ответ ЭС не понятен пользователю, то он может потребовать объяснения, как ответ получен. Для этого предназначена объяснительная компонента.

Состав участников разработки ЭС

В разработке ЭС участвуют представители следующих специальностей: эксперт в той проблемной области, задачи которой будет решать ЭС; инженер по знаниям - специалист по разработке ЭС; программист - специалист по разработке инструментальных средств (ИС). Необходимо отметить, что отсутствие среди участников разработки инженера по знаниям (т. е. его замена программистом) либо приводит к неудаче процесс создания ЭС, либо значительно удлиняет его.

Эксперт определяет знания, характеризующие проблемную область, обеспечивает полноту и правильность введенных в ЭС знаний.

Инженер по знаниям помогает эксперту выявить и структурировать знания, необходимые для работы ЭС, осуществляет выбор того ИС, которое наиболее подходит для данной проблемной области, и определяет способ представления знаний в этом ИС, выделяет и программирует (традиционными средствами) стандартные функции (типичные для данной проблемной области) которые будут использоваться в правилах, вводимых экспертом.

Программист разрабатывает ИС, содержащее все основные компоненты ЭС, осуществляет сопряжение ИС с той средой, в которой оно будет использовано.

№2. Классификация ЭС

Экспертные системы как любой сложный объект можно определить только совокупностью характеристик.

1. Назначение определяется следующей совокупностью параметров: цель создания ЭС - для обучения специалистов, для решения задач, для автоматизации рутинных работ, для тиражирования знаний экспертов и т. п.; основной пользователь - не специалист в области экспертизы, специалист, учащийся.

Сложность ЭС:

2. 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).

3. 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.

4. 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).

5. 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 (кооперирующимися).

6. По типу используемых методов и знаний ЭС делят на традиционные и гибридные. Традиционные ЭС используют в основном неформализованные методы инженерии знаний и неформализованные знания, полученные от экспертов. Гибридные ЭС используют и методы инженерии знаний, и формализованные методы, а также данные традиционного программирования и математики.

7. По степени сложности структуры ЭС делят на поверхностные и глубинные. Поверхностные ЭС представляют знания об области экспертизы в виде правил (условие ? действие). Условие каждого правила определяет образец некоторой ситуации, при соблюдении которой правило может быть выполнено. Поиск решения состоит в выполнении тех правил, образцы которых сопоставляются с текущими данными (текущей ситуацией в РП). При этом предполагается, что в процессе поиска решения последовательность формируемых таким образом ситуаций не оборвется до получения решения, т. е. не возникнет неизвестной ситуации, которая не сопоставится ни с одним правилом. Глубинные ЭС, кроме возможностей поверхностных систем, обладают способностью при возникновении неизвестной ситуации определять с помощью некоторых общих принципов, справедливых для области экспертизы, какие действия следует выполнить.

Совокупность рассматриваемых выше характеристик позволяет определить особенности конкретной ЭС. Однако пользователи зачастую стремятся охарактеризовать ЭС каким-либо одним обобщенным параметром. В этой связи говорят о поколениях ЭС. К первому поколению следует относить статические поверхностные ЭС, ко второму - статические глубинные ЭС (иногда ко второму поколению относят гибридные ЭС), а к третьему - динамические ЭС (вероятно, они, как правило, будут глубинными и гибридными).

Решаемые ЭС задачи можно характеризовать следующими аспектами: числом и сложностью правил, используемых в задаче; связностью правил; пространством поиска; количеством активных агентов, изменяющих предметную область; классом решаемых задач.

8. Пространство поиска ЭС может быть определено по крайней мере тремя подасpekтами: размером, глубиной и шириной. Размер пространства поиска дает обобщенную характеристику сложности задачи. Выделяют малые (до 101 состояний) и большие (свыше 101 состояний) пространства поиска. Глубина пространства поиска характеризуется средним числом последовательно применяемых правил, преобразующих исходные данные в конечный результат, ширина пространства - средним числом правил, пригодных к выполнению в текущем состоянии.

9. Класс решаемых задач характеризует методы, используемые ЭС для решения задачи. Данный аспект в существующих ЭС принимает следующие значения: задачи расширения, доопределения, преобразования. Задачи расширения и доопределения являются статическими, а задачи преобразования - динамическими.

К задачам расширения относятся задачи, в процессе решения которых осуществляется только увеличение информации о предметной области, не приводящее ни к изменению ранее выведенных данных, ни к выбору другого состояния области. Типичной задачей этого класса являются задачи классификации.

К задачам доопределения относятся задачи с неполной или неточной информацией о реальной предметной области, цель решения которых - выбор из множества альтернативных

текущих состояний предметной области того, которое адекватно исходным данным. В случае неточных данных альтернативные текущие состояния возникают как результат ненадежности данных и правил, что приводит к многообразию различных доступных выводов из одних и тех же исходных данных. В случае неполных данных альтернативные состояния являются результатом доопределения области, т. е. результатом предположений о возможных значениях недостающих данных.

К задачам преобразования относятся задачи, которые осуществляют изменения исходной или выведенной ранее информации о предметной области, являющиеся следствием изменений либо реального мира, либо его модели.

Большинство существующих ЭС решают задачи расширения, в которых нет ни изменений предметной области, ни активных агентов, преобразующих предметную область. Подобное ограничение неприемлемо при работе в динамических областях.

10. Выделяют следующие типы задач:

@ интерпретация данных - выбор решения из фиксированного множества альтернатив на базе введенной информации о текущей ситуации. Основное назначение - определение сущности рассматриваемой ситуации, выбор гипотез, исходя из 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.

@ конструирование, проектирование - разработка объекта с заданными свойствами при соблюдении установленных ограничений (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.

@ планирование - определение последовательности действий, приводящих к желаемому состоянию объекта; например, планирование процессов поставки продукции.

@ диспетчирование - распределение работ во времени, составление расписаний, например, планирование графика освоения капиталовложений.

@ слежение (мониторинг) - наблюдение за изменяющимся состоянием объекта и сравнение его показателей с установленными или желаемыми; для этого выполняется диагностика, прогнозирование, а в случае необходимости планирование и коррекция действий пользователей, например, мониторинг сбыта готовой продукции.

@ управление - воздействие на объект для достижения желаемого поведения (мониторинг, дополненный реализацией действий в автоматических системах), например, принятие решений на биржевых торгах.

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

	Анализ	Синтез	
Детерминированность знаний	Классифицирующие	Трансформирующие	Один источник знаний
Неопределенность знаний	Доопределяющие	Многоагентные	Множество источн. знаний
	Статика	Динамика	

Классы экспертных систем

Классифицирующие экспертные системы. К аналитическим задачам прежде всего относятся задачи распознавания различных 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.

Экспертные системы, решающие задачи распознавания ситуаций, называются классифицирующими, поскольку определяют принадлежность анализируемой ситуации к некоторому классу. В качестве основного метода формирования решений используется метод логического дедуктивного вывода от общего к частному, когда путем подстановки исходных данных в некоторую совокупность взаимосвязанных общих утверждений получается частное заключение.

Доопределяющие экспертные системы. Более сложный тип аналитических задач представляют задачи, которые решаются на основе неопределенных исходных данных и применяемых знаний. В этом случае экспертная система должна как бы доопределять недостающие знания, а в пространстве решений может получаться несколько возможных решений с различной вероятностью или уверенностью в необходимости их выполнения. В качестве методов работы с неопределенностями могут использоваться байесовский вероятностный подход, коэффициенты уверенности, нечеткая логика.

Доопределяющие экспертные системы могут использовать для формирования решения несколько источников знаний. В этом случае 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:

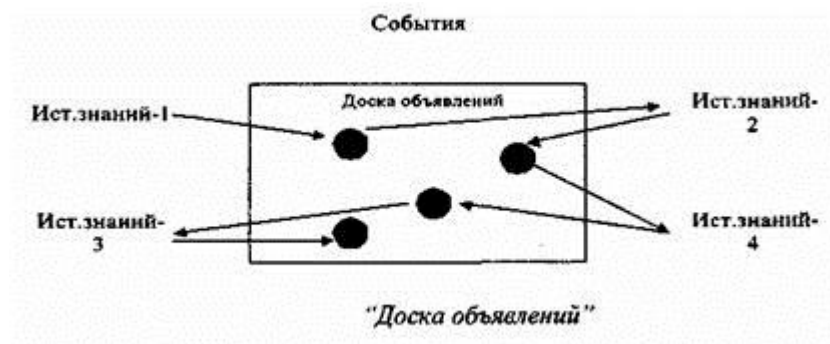
- Интерпретация данных • Диагностика • Коррекция

Трансформирующие экспертные системы. В отличие от аналитических статических экспертных систем синтезирующие динамические экспертные системы предполагают повторяющееся преобразование знаний в процессе решения задач, что связано с характером результата, который нельзя заранее предопределить, а также с динамичностью самой проблемной области.

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.

Многоагентные системы. Для таких динамических систем характерна интеграция в базе знаний нескольких разнородных источников знаний, обменивающихся между собой получаемыми результатами на динамической основе, например, через "доску объявлений".



Для многоагентных 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:

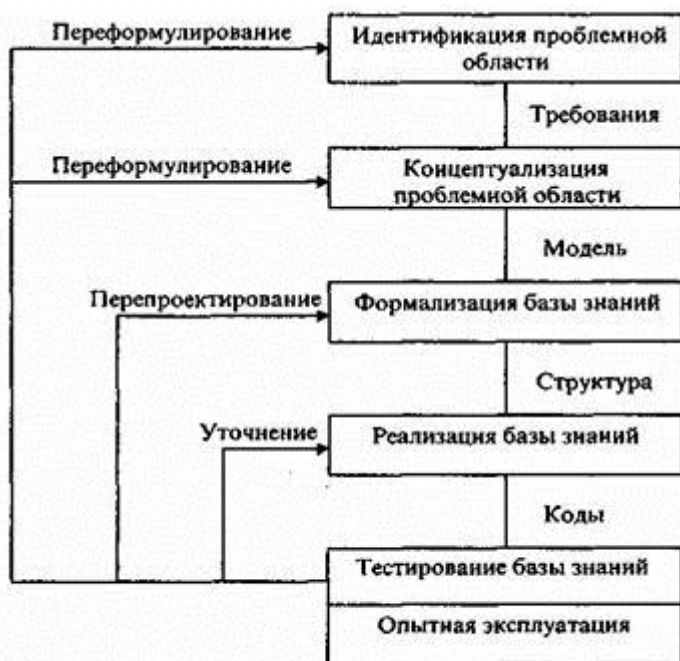
- Проектирование
- Диспетчирование
- Мониторинг
- Управление
- Прогнозирование
- Планирование

По данным публикации, в которой проводится анализ 12500 действующих экспертных систем, ЭС чаще исп-ся для диагностики, интерпретации.

№3. Этапы создания экспертной системы

Слабая формализуемость процесса принятия решений, его альтернативность и нечеткость, качественная и символическая 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.

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 рис.



Этапы создания экспертной системы

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 которых проблемной области оценивает эксперт. На этапе тестирования это могут быть совершенно другие эксперты.

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 табл.

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	200 - 500	3-6 мес.	
Формализация				
Реализация	Действующий	500 - 1000	6-12 мес.	25 - 50 т. \$
Тестирование	Промышленный	1000-1500	1-1,5 года	300 т. \$
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.

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.

Сфера применения экспертной системы характеризует тот круг задач, который подлежит формализации, например, "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.

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).

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:

"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-парного 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:

$$A_1 \wedge A_2 \wedge \dots \wedge A_m \rightarrow B$$

означающей, что факт В имеет место только в том случае, если имеет место конъюнкция фактов или их отрицаний A_1, A_2, \dots, A , 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. (СОБЫТИЕ-ПОВЕДЕНИЕ-СОСТОЯНИЕ)

Формализация базы знаний

На этапе формализации базы знаний осуществляется выбор метода представления знаний (совокупность средств структурирования и обработки единиц знаний). В рамках выбранного формализма осуществляется проектирование логической структуры базы знаний.

Классификация методов представления знаний с точки зрения особенностей отображения различных видов концептуальных моделей: соотношения структурированности и операционности, детерминированности и неопределенности, статичности и динамичности знаний.



Классификация методов представления 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;
- 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.

Реализация ЭС

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

Особенности реализации экспертной системы во многом определяются характером инструментального средства, в качестве которого могут выступать

- * программные оболочки (пустые ЭС),
- * средства автоматизации проектирования ЭС (генераторы, интегрированные среды),
- * языки представления знаний (языки инженерии знаний, программирования).

Оболочки ориентированы на работу с пользователем-непрофессионалом в области программирования. Использование оболочки сводится лишь к вводу БЗ. Каждая оболочка характеризуется фиксированным способом представления знаний и организации логического вывода. 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 Мета-технология), GURU (фирма MDBS, дистрибьютор фирма ЦПС Тверь), которые позволяют настраивать программные средства на особенности проблемных областей, при необходимости предоставляют возможность программировать на встроенных языках и осуществлять эффективный экспорт/импорт данных с другими инструментальными средствами. Они объединяют в себе различные методы решения задач, представления и интерпретации знаний. В их состав могут входить компоненты, модифицирующие или позволяющие строить свои оболочки. Эти инструментальные средства позволяют разработчику не программировать все или часть компонентов ЭС, а выбирать их из заранее составленного набора. 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.

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.

Тестирование и опытная эксплуатация ЭС

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.

№4. Стратегии поиска решения в ЭС

Методы поиска решения задач в ЭС зависят от особенностей проблемной области и от требований (ограничений), предъявляемых пользователем к решению. Особенности проблемной области с точки зрения методов решения можно охарактеризовать следующими параметрами:

- 1) размер пространства, в котором предстоит искать решение;
- 2) изменяемость области (статические и динамические области);
- 3) полнота модели, описывающей область (адекватность модели);
- 4) корректность данных решаемой задачи (степень точности и полноты данных).

Требования пользователя к решению задачи можно характеризовать:

- 1) количеством решений (одно, несколько, все решения);
- 2) свойствами результата (ограничениями). Поиск решения в одном пространстве

Поиск в пространстве состояний

Часто решение задачи формируется как поиск в пространстве состояний. Пусть задана тройка (S_0, F, S_k) , где S_0 - множество начальных состояний (условия задачи), F - множество операторов задачи, отображающих одни состояния в другие, S_k - множество конечных (целевых) состояний (решений задачи). Необходимо найти такую последовательность операторов, которая преобразует начальные состояния в конечные.

Процесс решения можно представить в виде графа $G=(X,Y)$, где $X=(x_0,x_1,x_2...)$ - множество вершин графа, каждая из которых отождествляется с одним из состояний, а Y - множество, содержащее пары вершин (x_i,x_j) , x_i и $x_j \in X$. Граф G задает пространство состояний, т.е. пространство, в котором осуществляется поиск решения. Решение задачи методом поиска в

пространстве состояний сводится к задаче поиска пути L на графе G . Часто дугам приписывают веса, отражающие стоимость применения соответствующих операторов. Стоимость пути между двумя вершинами определяется как сумма стоимостей всех дуг, образующих этот путь. В ряде приложений возникает задача нахождения пути с минимальной стоимостью.

Граф G может быть задан явно и неявно. Неявное задание графа состоит в определении множества X и множества операторов, которые будучи применены к некоторой вершине дают все ее дочерние вершины.

Построение пространства состояний осуществляется так. Берется некоторая вершина, к ней применяются все возможные операторы, порождающие все дочерние вершины (раскрытие вершин). Если получена целевая вершина, то она не раскрывается. Процесс построения пространства состояний заканчивается, если остались только целевые и терминальные вершины. На практике процесс порождения пространства состояний ограничивают либо временем, либо объемом памяти. Часто требуется организовать поиск так, чтобы все целевые вершины были найдены, если они существуют. Надежным способом обеспечения полноты является полный перебор. Необходимо определить порядок, в котором будут перебираться вершины графа. Обычно выделяют следующие стратегии поиска: поиск в глубину, поиск в ширину, обратный и прямой выводы.

Суть поиска в глубину состоит в том, что при выборе очередной подцели в пространстве состояний предпочтение всегда, когда возможно, отдается той, которая соответствует следующему, более детальному уровню описания задачи. Например, диагностирующая система, сделав на основе известных симптомов предположение о наличии определенного заболевания, будет продолжать запрашивать уточняющие признаки и симптомы этой болезни до тех пор, пока полностью не подтвердит или отвергнет выдвинутую гипотезу.

При поиске в ширину, напротив, система вначале проанализирует все симптомы, находящиеся на одном уровне пространства состояний, даже если они относятся к разным заболеваниям, и лишь затем перейдет к симптомам следующего уровня детализации. Вершины раскрываются в том порядке, в котором они порождались. Метод поиска в ширину требует больше памяти.

Поиск может быть от данных (прямой), от цели (обратный) и бинаправленный (двунаправленным).

В ЭС с обратным (индуктивным) выводом выдвигается некоторая гипотеза, а затем механизм вывода в процессе работы как бы возвращается назад, переходя от нее к фактам и пытается найти среди них те, которые подтверждают эту гипотезу. Если она оказалась правильной, то выбирается следующая гипотеза, детализирующая первую и являющаяся по отношению к ней подцелью. Далее отыскиваются факты, подтверждающие истинность подчиненной гипотезы. Обратный поиск применяется в тех случаях, когда цели известны и их сравнительно немного.

В системах с прямым (дедуктивным) выводом по известным фактам отыскивается заключение, которое из этих фактов следует.

Поиск методом редукции

При поиске методом редукции решение задачи сводится к решению совокупности образующих ее подзадач. Этот процесс повторяется для каждой подзадачи до тех пор, пока

каждая из подзадач, образующих решение, не будет очевидной. Подзадача считается очевидной, если ее решение известно или получено ранее. Процесс решения задачи разбиением ее на подзадачи можно представить в виде специального направленного графа, называемого И/ИЛИ графом. Каждой вершине этого графа ставится в соответствие описание некоторой задачи (подзадачи). В графе выделяют два типа вершин: конъюнктивные (решение задачи сводится к решению всех ее подзадач) и дизъюнктивные (необходимо решение одной из подзадач). Решение задачи сводится к нахождению решающего графа – подграфа из разрешимых вершин. Определение разрешимости вершины можно сформулировать рекурсивно:

1. конечные вершины разрешимы, так как их решение известно по исходному предположению
2. вершина ИЛИ разрешима, если разрешима по крайней мере одна из ее дочерних вершин
3. вершина И разрешима, если разрешимы все ее дочерние вершины

Перебор в пространстве состояний можно рассматривать как частный случай графа И/ИЛИ.

Для графа И/ИЛИ так же как для поиска в пространстве состояний, можно определить поиск и глубину и ширину как в прямом, так и в обратном направлении.

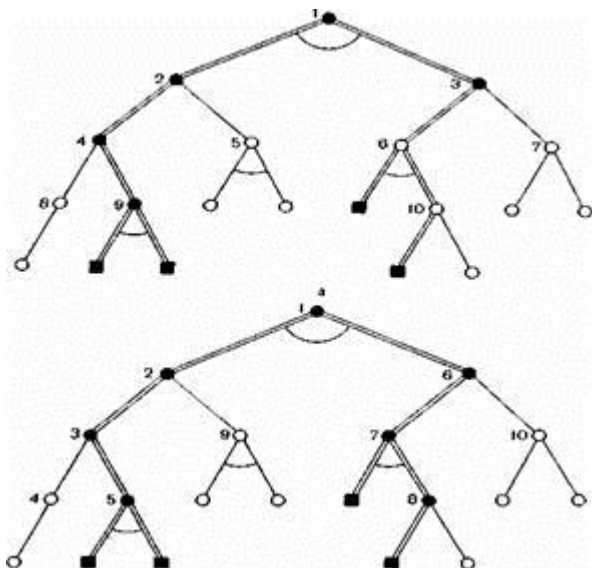


Рис. Пример разбиения задачи на подзадачи при поиске в ширину (а) и при поиске в глубину (б)

Эвристический поиск

Методы поиска в глубину и ширину называют слепым поиском, поскольку в этих методах порядок раскрытия вершин predetermined и ни как не зависит от расположения цели. При увеличении пространства поиска методы слепого поиска требуют чрезмерных затрат времени и (или) памяти. Стремление сократить время поиска привело к созданию эвристических методов поиска, т.е. методов, использующих некоторую информацию о проблемной области для рассмотрения не всего пространства поиска, а таких путей в нем, которые с наибольшей вероятностью приводят к цели.

Один способ сокращения перебора состоит в выборе более "информированного" оператора, который не строит так много вершин, не относящихся к делу. Другой способ состоит в использовании эвристической информации для определения на каждом шаге дальнейшего

направления перебора. Для этого необходимо ввести меру "перспективности" вершины в виде некоторой оценочной функции. В некоторых случаях удается ввести такую оценочную функцию, что она, сокращая перебор, не теряет свойства полноты. Чаще же используемые эвристики, сильно сокращая перебор, влекут за собой потерю свойства полноты. Как правило, оценочные функции пытаются количественно оценить расстояние от текущей вершины до конечной. Из двух вершин при одинаковой глубине перспективней та, от которой меньше расстояние до цели. Для многих приложений, в частности для экспертных систем, применение количественных оценок не позволяет эффективно направлять процесс поиска.

Поиск методом "генерация - проверка"

Процесс поиска может быть сформулирован в терминах "генерация - проверка". Действительно, пространство поиска (пространство состояний или И/ИЛИ граф), как правило, явно не задано. Поэтому для осуществления процесса поиска необходимо генерировать очередное возможное решение (состояние или подзадачу) и проверить, не является ли оно результирующим. Разумно потребовать, чтобы генератор удовлетворял требованиям полноты и избыточности. Говорят, что генератор является полным, если он обеспечивает генерацию всех возможных решений. Генератор является избыточным, если он генерирует каждое решение только один раз. Обеспечение свойства не избыточности является важным, но трудновыполнимым, так как в соответствии с этим требованием не допускается генерация не только тождественных, но и синонимичных решений. Например, если задача генератора - синтезировать все фразы русского языка, то весьма трудно (если вообще возможно) сделать такой генератор избыточным. При генерации текущего возможного решения (состояния или подзадачи) возникает проблема распределения знаний между генератором и устройством проверки. При слепом и эвристическом поиске генератор имеет минимальные знания о проблемной области, достаточные для генерации всех возможных решений (состояний или подзадач), а устройство проверки определяет, не является ли очередное решение целевым. В принципе некоторые знания можно перенести из устройства проверки в генератор, чтобы он не генерировал решения, которые заведомо не могут привести к успеху. По сути дела, увеличение знаний генератора о проблемной области приводит к сокращению пространства, в котором осуществляется поиск. Однако при этом повышаются затраты на генерацию каждого очередного состояния (подзадачи).

Можно выделить важную форму метода "генерация - проверка", называемую иерархическая "генерация — проверка". В этом случае на верхнем уровне генератор вырабатывает не полное, а частично определенное решение (будем для краткости называть такие решения частичными). Каждое частичное решение описывает, например, не все состояние, а только его некоторую часть, определяя таким образом класс возможных состояний. Идея состоит в том, что устройство проверки может уже по виду частичного решения определить, что оно (а следовательно, и все полные решения, которые могут быть получены из него) не ведет к успеху. Если же проверка не отвергает частичное решение, то на следующем уровне генератор продолжает вырабатывать из данного частичного решения все полные решения, а устройство проверки определяет, являются ли они целевыми.

Ограничения

Поиск в иерархии пространств

Методы поиска в одном пространстве не позволяют решать сложные задачи, так как с увеличением размера пространства время поиска экспоненциально растет. При большом размере пространства поиска можно попробовать разбить общее пространство на

подпространства и осуществлять поиск сначала в них. Можно сказать, что в данном случае пространство поиска представлено иерархией пространств. В иерархии присутствуют не только конкретные, но и абстрактные пространства, т.е. пространства которые имеют описание только наиболее важных сущностей. Использование неполных описаний позволяет сократить пространство и (или) сделать операторы в нем более мощными, что приводит к дополнительному ускорению решения задачи.

В качестве классического примера использования абстрактных пространств можно привести задачу определения кратчайшего пути на карте. Пусть требуется переехать из центра города А в центр города В. Если осуществлять поиск требуемого пути на детальной карте, содержащей все улицы во всех городах, встретившихся по дороге, то задача может стать практически неразрешимой. При определении пути из города А в город В целесообразно спланировать маршрут по крупномасштабной карте (т.е. осуществить поиск в абстрактном пространстве), а затем по детальной карте спланировать выезд из города А и въезд в город В.

Поиск в факторизованном пространстве

Во многих приложениях требуется найти все решения. Примерами таких областей являются интерпретация данных, постановка диагноза и др. Действительно, в случае постановки диагноза нас интересуют все, а не некоторые болезни пациента. Однако пространство поиска в практических приложениях бывает столь велико, что не позволяет применить слепые методы поиска. Применение эвристических методов в данном случае, как правило, также исключено, так как они не обеспечивают получение всех возможных решений. Если пространство поиска удастся факторизовать, то поиск даже в очень большом пространстве можно организовать эффективно. Пространство называется факторизованным, если оно разбивается на непересекающиеся подпространства (классы) частичными (неполными) решениями. Причем по виду частичного решения можно определить, что оно не приведет к успеху, т.е. что все полные решения, образованные из него, не приведут к целевым решениям. Поиск в факторизованном пространстве осуществляется на основе метода иерархическая "генерация-проверка". Генератор вырабатывает текущее частичное решение, затем проверяется, может ли это решение привести к успеху. Если текущее частичное решение отвергается, то из рассмотрения без генерации и проверки устраняются все полные решения этого класса. Если текущее частичное решение не отвергается, то генератор вырабатывает на его основе все полные решения, а устройство проверки определяет, являются ли эти решения целевыми.

Поиск в фиксированном множестве пространств

Применение рассмотренного выше метода факторизации пространства ограничено тем, что для ряда областей не удастся по частичному решению сделать заключение о его непригодности. Примерами таких областей являются задачи планирования и конструирования. Действительно, как правило, по фрагменту плана или конструкции нельзя сказать, что этот фрагмент не может являться частью полного решения. В данном и последующих двух пунктах будут рассмотрены методы поиска, использующие идею абстрактного пространства. Методы различаются предположениями о природе этого пространства. Абстракция должна подчеркнуть важные особенности рассматриваемой задачи, позволить разбить задачу на более простые подзадачи и определить последовательность подзадач (план решения), приводящую к решению основной задачи. В простейшем случае пространство поиска разбивается на фиксированную последовательность подзадач (подпространств), с помощью которых можно решить любую входную задачу.

Поиск в изменяющемся множестве иерархических пространств

В ряде приложений не удастся все решаемые задачи свести к фиксированному набору подзадач. Примерами таких приложений являются задачи планирования перемещений в пространстве. План решения задачи в данном случае должен иметь переменную структуру и не может быть сведен к фиксированному набору подзадач. Для решения подобных задач может быть использован метод нисходящего уточнения (top-down refinement). Для того чтобы упростить процесс решения некоторой задачи в сложном пространстве, целесообразно получить обобщенное (и, следовательно, более простое) пространство и попробовать получить решение в этом пространстве. Указанный прием можно повторять многократно. При этом полный процесс решения задачи можно представить как "нисходящее" движение в иерархии пространств от наиболее абстрактного к конкретному, в котором получается окончательное решение. Существенной характеристикой такого процесса является поиск решения задачи в абстрактном пространстве, преобразование этого решения в решение более низкого уровня и т.д. Причем на каждом уровне вырабатывается окончательное решение и только затем осуществляется переход на следующий, более конкретный уровень. Внутри каждого уровня подзадачи рассматриваются как независимые, что создает частичное упорядочение абстрактных состояний. Формирование более абстрактного пространства осуществляется путем игнорирования части описаний менее абстрактного пространства (на первом шаге - конкретного пространства). Игнорирование описаний осуществляется на основе ранжирования описаний по степени важности. Часто ранжирование осуществляется на основе учета степени неизменности фактов (наиболее абстрактны те описания, которые не могут изменяться). При этом абстрактные пространства, с одной стороны, должны для упрощения решения задачи обеспечивать значительное упрощение исходного пространства, а с другой стороны, должны быть подобны друг другу и конкретному пространству, чтобы процесс "нисходящего" переноса решения из более абстрактных пространств в менее абстрактные не требовал больших вычислительных затрат.

Завершая описание метода "нисходящего уточнения", отметим, что абстрактные пространства здесь создаются индивидуально, в соответствии с решаемой задачей. Необходимо отметить, что метод базируется на следующих предположениях: 1) возможно осуществить частичное упорядочение понятий области, приемлемое для всех решаемых задач; 2) решения, принимаемые на верхних уровнях, нет необходимости отменять на более нижних.

Принцип наименьших свершений

Основной недостаток метода "нисходящего уточнения" состоит в том, что он не имеет обратной связи. Метод предполагает, что одни и те же решения должны приниматься в одинаковых ситуациях при решении любой задачи.

При решении ряда задач детализация решения, полученного на абстрактном уровне, оказывается невозможна, так как при построении абстрактного плана были опущены детали, препятствующие его уточнению, т.е. требуется пересмотр абстрактного плана (решения). В подобных ситуациях целесообразно применение принципа наименьших свершений. В соответствии с данным принципом решение не строится сразу до конца на верхних уровнях абстракции. Частичное решение детализируется постепенно, по мере появления информации, подтверждающей возможность решения и вынуждающей принять решение. Рассуждение, основанное на использовании принципа наименьших свершений, требует, чтобы система была в состоянии: 1) определить, когда накопилось достаточно информации, для принятия решения; 2) приостанавливать работу над некоторой подзадачей, когда для решения нет достаточной информации; 3) переходить с одной подзадачи на другую, возобновляя выполнение приостановленной подзадачи при появлении недостающей информации; 4) объединять информацию, полученную различными подзадачами.

Итак, принцип наименьших свершений координирует процесс поиска решения с наличием необходимой информации и в соответствии с доступной информацией перемещает фокус активности по решению задачи от одной подзадачи к другой. Данный подход непригоден, когда существует много возможностей, но нет надежных оснований для выбора решения. В этих случаях необходимо использовать некоторые формы правдоподобных рассуждений или переходить на использование другой модели. Необходимо отметить, что для управления процессом решения задачи принцип наименьших свершений требует больших знаний, чем метод нисходящего уточнения.

Метапространство в иерархии пространств

При решении любой задачи многократно возникает вопрос: "что делать на следующем шаге"? В простейшем случае решение о том, что делать на следующем шаге, предопределено методом поиска решения. При поиске в абстрактных и конкретных пространствах на каждом шаге решался вопрос о том, какой из операторов, существующих в проблемной области, применить к текущему состоянию проблемной области. Вопрос о том, как решающая программа это сделает, не обсуждался. Можно оказывать не явное, а косвенное влияние на определение того, "что делается на следующем шаге в проблемной области" путем выбора того или иного метода, известного решателю. Подобный подход требует явного разграничения знаний о процессе решения и знаний о проблемной области. В последнее время наблюдается интерес к разработке систем, которые могут рассуждать о процессе получения ими решения, например планировать процесс своих рассуждений. В такой постановке речь идет о решении задачи на метауровне. Решатель в метапространстве содержит явное описание процесса организации поиска, т.е. описание состояний, операторов, условий применимости операторов, описание доступных методов (стратегий) поиска и способов их взаимодействия. Получить решение в метапространстве - это значит определить, какой метод (программа) будет применен на следующем шаге, т.е. составить метаплан решения задачи. Заметим, что метаплан, в отличие от абстрактного плана, выражается не в терминах операторов проблемной области, а в терминах методов (программ), известных решателю.

Не существует причин ограничивать метазнания одним уровнем.

Завершая описания методов поиска в иерархии пространств, подчеркнем, что в рассмотренных подходах используются пространства трех видов: конкретные, абстрактные и метапространства. В одной системе могут использоваться пространства всех трех типов. Необходимо также отметить, что поиск решения в пределах некоторого пространства может осуществляться любым из рассмотренных выше методов.

Поиск в альтернативных пространствах

Методы, рассмотренные выше, исходят из предпосылки, что знания о проблемной области и данные о решаемой задаче являются точными и полными. Однако при решении любых практических задач и особенно при решении неформализованных задач распространена обратная ситуация. Эксперту приходится работать в условиях неполноты и неточности знаний (данных) и, как правило, в условиях дефицита времени. В этом случае эксперт делает пробные правдоподобные предположения, которые он не может доказать; тем самым вопрос об их истинности остается открытым. Все утверждения, полученные на основе этих правдоподобных предположений, также не могут быть доказаны.

Один из способов обоснования предположений заключается в том, чтобы рассматривать их как возможные значения, задаваемые по умолчанию. Обычно человек знает, что некоторые

предположения верны только при определенных условиях. Если информация, указывающая на нарушение этих условий, отсутствует, то предположение может быть высказано. Другое обоснование предположений базируется на рассмотрении рассуждения как процесса с ограниченными ресурсами. Так, можно считать, что предположение (X) имеет место, если, используя ограниченные ресурсы, нельзя доказать истинность противоположного утверждения. Предположение и выводы, сделанные на его основе, должны устраняться, если появилась информация, показывающая ошибочность этого предположения. Этот аспект в построении умозаключений с использованием предположений называется немонотонностью. Немонотонные рассуждения особенно важны при решении задач планирования и конструирования. В этих задачах пространство поиска иногда очень велико, и нет возможности предвидеть все последствия сделанного выбора. Поэтому при конструировании предположения выступают в виде пробных решений, последствия которых затем анализируются с точки зрения их пригодности (непригодности). Если последствия не противоречат тому, что хотел конструктор, то процесс конструирования продолжается дальше, возможно, с выдвижением новых предположений. В противном случае необходимо устранить все последствия и сделать альтернативное предположение и т.д.

Итак, для того чтобы система могла делать умозаключения, основанные на здравом смысле, при работе с неполными (неточными) данными и знаниями, она должна быть способна делать предположения, а при получении новой информации, показывающей ошибочность предположений, отказываться как от сделанных предположений, так и от умозаключений, полученных на основе этих предположений. Мнение системы о том, какие факты имеют место, изменяется в ходе рассуждения, т.е. можно говорить о ревизии мнения. Таким образом, даже если рассматривать проблемную область как статическую, неполнота (и неточность) знаний и данных влечет за собой рассмотрение этой области при различных (и даже противоположных) предположениях, что в свою очередь приводит к представлению области в виде альтернативных возможных пространств, соответствующих различным, возможно, противоречивым и (или) взаимодополняющим предположениям и мнениям.

Будем говорить, что множество мнений, свойственных некоторому индивиду (системе), составляют его систему мнений. Основываясь на некоторой системе мнений, можно образовать пространство поиска, предназначенное для решения каких-либо задач. В ходе рассуждений человек (система) может менять свои мнения, образуя различные системы мнений. Совокупность мнений, которой система придерживается в текущий момент, будем называть активной системой мнений. Каждой из систем мнений соответствует свое пространство поиска, а все вместе они образуют альтернативные пространства.

Для того чтобы изменить мнение, система должна быть способна рассуждать о зависимостях, существующих в активной системе мнений. Новые мнения могут быть следствием новой информации, полученной извне или выведенной. Зависимости в системе мнений должны содержать сведения о мнениях, правилах вывода и обоснованиях (поддержках) мнений. Простейшим видом обоснования (justification) может являться информация о том, на каких мнениях основывается факт.

Обоснование может использоваться для поддержания или изменения текущей системы мнений. Этот процесс обычно называют ревизией мнений (belief revision). Все мнения, входящие в активную систему мнений, имеют законные обоснования. Обоснование считается законным, если каждое мнение, которое в нем использовано, входит в активную систему мнений. Система мнений изменяется, когда обоснования добавляются или модифицируются.

Для того чтобы сделать предположение (угадывание), можно использовать специальный вид обоснования, называемый немонотонным обоснованием: обоснованием может служить не только мнение о некоторых утверждениях, но и отсутствие такого мнения.

В ранее описанных методах при достижении терминального состояния необходимо вернуться в некоторое предыдущее состояние пространства и продолжить поиск в новом направлении. Обычно при реализации поиска использовался бэктрекинг. Этот механизм восстанавливает состояние, непосредственно предшествующее текущему, и затем выбирает очередную альтернативу. В связи с тем, что бэктрекинг работает по принципу "last-in, first-out", т.е. сначала устраняется последнее рассматриваемое состояние, для его реализации может быть применен стек.

Применение бэктрекинга при поиске в альтернативных мирах будет приводить к излишней неэффективности, так как все неудачи, возникшие при поиске в одном направлении, не запоминаются при переходе к поиску в другом направлении. Та же самая причина неудачи может заново обнаруживаться и на новом направлении. Традиционный бэктрекинг отбрасывает слишком много информации. Осуществлять возврат целесообразно не к состоянию, непосредственно предшествующему данному, а к тому состоянию, которое является причиной возникновения неудачи. В используемых нами терминах причиной неудач являются предположения, т.е. недоказуемые утверждения. Поэтому при обнаружении неудачи необходимо возвращаться в состояние, где это предположение было сделано, и испытывать другое предположение. Данный метод называют управляемым зависимостью бэктрекингом, а традиционный бэктрекинг называют хронологическим (временным) бэктрекингом.

Поиск с использованием нескольких моделей

При решении сложных задач в условиях ограниченных ресурсов использование нескольких моделей может значительно повысить мощность системы. В основу взаимодействия моделей лежит принцип косвенного взаимодействия, т.е. взаимодействие источников знания осуществляется только через общую память, называемую "доской объявлений" (blackboard). Механизм, используемый для обеспечения взаимодействия источников знаний, представляет собой парадигму построения гипотез и их проверки. Поиск решения рассматривается как итеративный процесс, в котором имеют место два типа действий источников знаний: 1) создание на доске гипотез, т.е. высказывание предположения о некотором аспекте задачи; 2) проверка правдоподобности некоторых гипотез. Уровень и время являются двумя измерениями пространства гипотез, делящими доску на части и используемыми для адресации гипотез. Если временные интервалы двух и более гипотез одного и того же уровня существенно перекрывают друг друга, то эти гипотезы являются конкурирующими. Гипотезы различных уровней связаны в направленный граф, что позволяет описывать гипотезы одного уровня через гипотезы более низкого уровня. Частичное решение (т.е. группа гипотез) одного уровня может использоваться для поиска решения на соседних уровнях.

Сложность и многообразие вариантов анализа, возникающих вследствие неопределенностей во входных сигналах и ошибок источников знаний, не позволяют при ограниченности вычислительных ресурсов рассматривать все варианты обработки. Поиск решения осуществляется с помощью управляющего процесса, называемого приспособляющимся планированием. Такое планирование объединяет идеи принципа наименьших свершений со стратегиями управления ограниченными вычислительными ресурсами (варьированием ширины поиска и комбинированием поиска от целей с поиском от данных). Приспособляющийся планировщик автоматически адаптируется к изменению условий

неопределенности изменением ширины поиска. Основным механизмом приспособления является взаимодействие назначения оценок правдоподобности гипотез, определяющих активацию соответствующих источников знаний, с назначением приоритетов источникам знаний, ожидающих активации. Если наибольшую оценку имеют несколько гипотез, то для раскрытия многозначности ЭС расширяет зону поиска. При этом выбор между равноправными конкурирующими гипотезами откладывается до тех пор, пока не будет получена дополнительная информация. Среди тех источников знания, которые подходят к гипотезе с наивысшей оценкой правдоподобия, выбирается источник знания с наивысшим приоритетом. Вычисление приоритета соответствует попытке оценить полезность действия источника знания для достижения общей цели и осуществляется на основе анализа фреймов стимула и реакции.

Разнообразие уровней "доски" обеспечивает разбиение большого пространства поиска на абстрактные подпространства, а разнородность уровней обеспечивает интерпретацию входных сигналов с различных точек зрения. Приспосабливающееся планирование объединяет идеи принципа наименьших свершений со способностью управлять ограниченными вычислительными ресурсами.

Выбор метода решения задачи

Простые задачи характеризуются следующими значениями параметров: малый (не слишком большой) размер пространства поиска; точность и полнота данных; статичность области; возможность адекватного описания области с помощью модели. На практике встречается мало приложений, удовлетворяющих перечисленным требованиям. Эти методы объединяют то, что все они используют одно пространство поиска.

Сложные задачи характеризуются тем, что значение хотя бы одного из перечисленных параметров оказывается в них более сложным.

Для преодоления трудностей, вызванных большим пространством поиска, используются методы, основанные на введении иерархии пространств (конкретных, абстрактных и метапространств). Простейший из этих методов основывается на факторизуемости пространства решений, что позволяет производить раннее "отсечение". Метод обеспечивает получение всех решений. Если пространство поиска не удастся факторизовать, но при этом не требуется получать все решения или выбирать лучшее, то могут быть применены методы, использующие иерархию однородных абстрактных пространств. Если пространство поиска таково, что любая задача может быть сведена к известной заранее последовательности подзадач, то используется фиксированное абстрактное пространство. Эффективность этого метода определяется возможностью использовать безвозвратную стратегию. В тех случаях, когда решение задачи не может быть получено без бэктрекинга, применяются более сложные методы. Метод "нисходящего уточнения" применим в том случае, когда все задачи не могут быть сведены к фиксированному набору подзадач, однако существует фиксированная упорядоченность понятий области и фиксированный частичный порядок между подзадачами. В случае, если подзадачи взаимозависимы, т.е. для решения некоторой подзадачи может требоваться информация, получаемая другой подзадачей, и подзадачи не могут быть упорядочены, целесообразно применять принцип наименьших свершений. Этот подход позволяет приостанавливать решение подзадачи, для которой не достает информации, переходить к решению другой подзадачи и возвращаться к исходной задаче, когда отсутствующая информация станет доступной. Использование данного подхода требует более разнообразных знаний о решении задачи, чем в предыдущих случаях. При использовании разнообразных знаний о процессе решения становится целесообразным объединять принцип наименьших свершений с методами, использующими метазнание.

Принцип наименьших свершений может приводить к образованию тупиков в процессе решения задачи, что препятствует использованию этого принципа в чистом виде. Для преодоления тупиков используют предположения или применяют метод нескольких моделей.

Для преодоления трудностей, вызванных неполнотой и (или) неточностью данных (знаний), используют вероятностные, нечеткие и точные методы. Все эти методы основываются на идее увеличения надежности путем комбинирования фактов и использования метазнаний о возможностях комбинирования фактов. Неточные подходы (вероятностные, псевдовероятностные, нечеткие) используют разнообразные априорные оценки, условные вероятности и нечеткие переменные; точные подходы используют предположения и ревизию мнений при немонотонных рассуждениях.

Для преодоления трудностей, вызванных изменениями проблемной области, используются методы поиска в динамических пространствах. Необходимо отметить, что существующие средства учета динамики проблемных областей далеки от совершенства и не удовлетворяют требованиям практики.

Для преодоления неадекватности модели проблемной области используются методы, ориентированные на использование нескольких моделей. Эти методы позволяют объединить возможности различных моделей, описывающих проблемную область с различных точек зрения. Кроме того, использование нескольких моделей позволяет уменьшить вероятность потери хорошего решения, несмотря на неполноту поиска, вызванную ограниченностью вычислительных ресурсов.

Завершая рассмотрение методов поиска, отметим, что ускорение работы экспертных систем может быть достигнуто не только за счет сокращения пространства поиска, но и за счет использования специализированных структур знаний. Единообразное представление знаний, применяемое в большинстве существующих систем, становится неэффективным при увеличении базы знаний.

№5. Представление знаний в экспертных системах

Специалистами предложены десятки различных толкований понятия «знание». Определение знаний вне контекста ИИ звучит следующим образом: «Проверенный практикой результат познания действительности, верное ее отражение в мышлении человека».

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

Существуют так называемые метазнания (знания о знаниях). Понятие «метазнания» указывает на знания, касающиеся способов использования знаний и свойств знаний. Это понятие необходимо для управления базой знаний, логическим выводом, отождествления, обучения и т. п.

Процедуральные и декларативные знания.

Чем же знания отличаются от данных?

Можно выделить еще пять свойств, отличающих знания:

- | | | | | | |
|----|------------|----------------|--------------|---|---------------------|
| 1) | | внутренняя | | | интерпретируемость, |
| 2) | | рекурсивная | | | структурируемость, |
| 3) | | | взаимосвязь | | единиц, |
| 4) | наличие | семантического | пространства | с | метрикой, |
| 5) | активность | | | | |

Знания в СИИ представляются некоторой знаковой (семиотической) системой. В характеристиках семиотических систем выделяют три аспекта: синтаксический, семантический и прагматический.

Синтаксис описывает внутреннее устройство знаковой системы, т. е. правила построения и преобразования знаковых выражений. Семантика определяет отношения между знаками и их концептами, т. е. задает смысл или значения конкретных знаков. Прагматика определяет знак с точки зрения конкретной сферы его применения.

В соответствии с перечисленными аспектами семиотических систем можно выделить три типа знаний: Синтаксические знания характеризуют синтаксическую структуру описываемого объекта или явления, которая не зависит от смысла и содержания используемых при этом понятий; Семантические знания содержат информацию, непосредственно связанную со значениями и смыслом описываемых явлений и объектов; Прагматические знания описывают объекты и явления с точки зрения решаемой задачи с учетом действующих в ней специфических критериев.

Типам знаний соответствуют три типа моделей для их представления. Наличие семантических и(или) прагматических моделей - наиболее существенный признак, отличающий интеллектуальные системы от других.

Модели 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. Область определения предиката задается в виде фактов, действия описываются как правила, определяющие логическую формулу вывода фактов из других фактов.

Пример:

Выбор(ФИО) если

(претендент(ФИО, Образование, Стаж) и

вакансия(Должность, Образование, Опыт) и

Стаж \geq Опыт)

Претендент ("Иванов", "среднее", 10)

Претендент ("Петров", "высшее", 12)

Вакансия ("менеджер", "высшее", 10)

Вакансия ("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 $\text{vibor}(X, Y)$ получим: $X = \text{"Петров"}$, $Y = \text{"менеджер"}$.

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 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 помощью которого могут быть 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 описания объектов фигурируют только в качестве переменных ("атрибут - значение") внутри правил. В основном 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. Так, 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 также 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.

В общем виде под продукцией понимается выражение следующего вида:

(i); Q; P; A=>B, N.

Здесь *i*—имя продукции, с помощью которого данная продукция выделяется из всего множества продуктов. В качестве имени может выступать некоторая лексема, отражающая суть данной продукции (например, «покупка книги» или «набор кода замка»), или порядковый номер продукции в их множестве, хранящемся в памяти системы.

Элемент Q характеризует сферу применения продукции. Такие сферы легко выделяются в когнитивных структурах человека. Наши знания как бы «разложены по полочкам». На одной «полочке» хранятся знания о том, как надо готовить пищу, на другой—как добраться до работы и т. п. Разделение знаний на отдельные сферы позволяет экономить время на поиск нужных знаний. Такое же разделение на сферы в базе знаний ИС целесообразно и при использовании для представления знаний продукционных моделей.

Элемент P есть условие применимости ядра продукции. Обычно P представляет собой логическое выражение (как правило, предикат). Когда P принимает значение «истина», ядро продукции активизируется. Если P ложно, то ядро продукции не может быть использовано. Например, если в продукции «НАЛИЧИЕ ДЕНЕГ; ЕСЛИ ХОЧЕШЬ КУПИТЬ ВЕЩЬ X, ТО ЗАПЛАТИ В КАССУ ЕЕ СТОИМОСТЬ И ОТДАЙ ЧЕК ПРОДАВЦУ» условие применимости ядра продукции ложно, т. е. денег нет, то применить ядро продукции невозможно.

Основным элементом продукции является ее ядро: A=>B. Интерпретация ядра продукции может быть различной и зависит от того, что стоит слева и справа от знака секвенции =>. Обычное прочтение ядра продукции выглядит так: ЕСЛИ A, ТО B, более сложные конструкции ядра допускают в правой части альтернативный выбор, например, ЕСЛИ A, ТО B1, ИНАЧЕ B2, Секвенция может истолковываться в обычном логическом смысле как знак логического следования B из истинного A (если A не является истинным выражением, то о B ничего сказать нельзя). Возможны и другие интерпретации ядра продукции, например A описывает некоторое условие, необходимое для того, чтобы можно было совершить действие B.

Элемент N описывает постусловия продукции. Они актуализируются только в том случае, если ядро продукции реализовалось. Постусловия продукции описывают действия и процедуры, которые необходимо выполнить после реализации B. Например, после покупки некоторой вещи в магазине необходимо в описи товаров, имеющихся в этом магазине, уменьшить количество вещей такого типа на единицу. Выполнение N может происходить не сразу после реализации ядра продукции.

Классификация ядер продукции. Ядра продукции можно классифицировать по различным основаниям. Прежде всего все ядра делятся на два больших типа: детерминированные и недетерминированные. В детерминированных ядрах при актуализации ядра и при выполнимости A правая часть ядра выполняется обязательно; в недетерминированных ядрах B может выполняться и не выполняться. Таким образом, секвенция => в детерминированных ядрах реализуется с необходимостью, а в недетерминированных — с возможностью. Интерпретация ядра в этом случае может, например, выглядеть так: ЕСЛИ A, ТО ВОЗМОЖНО B.

Возможность может определяться некоторыми оценками реализации ядра. Например, если задана вероятность выполнения B при актуализации A, то продукция (в простейшем случае продукция состоит лишь из ядра) может быть такой: ЕСЛИ A, ТО С ВЕРОЯТНОСТЬЮ p РЕАЛИЗОВАТЬ B. Оценка реализации ядра может быть лингвистической, связанной с

понятием терм-множества лингвистической переменной, например: ЕСЛИ А, ТО С БОЛЬШОЙ ДОЛЕЙ УВЕРЕННОСТИ В. Возможны иные способы задания оценки реализации ядра.

Детерминированные продукции могут быть однозначными и альтернативными. Во втором случае в правой части ядра указываются альтернативные возможности выбора, которые оцениваются специальными весами выбора. В качестве таких весов могут использоваться вероятностные, лингвистические, экспертные оценки и т.п. (например, ЕСЛИ А ТО ЧАЩЕ ВСЕГО НАДО ДЕЛАТЬ В1, РЕЖЕ В2)

Особым типом являются прогнозирующие продукции, в которых описываются последствия, ожидаемые при актуализации А, (например, ЕСЛИ А, ТО С ВЕРОЯТНОСТЬЮ р можно ожидать В).

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:

$$\frac{P(H/E)}{P(\bar{H}/E)} = \frac{P(E/H) * P(H)}{P(E/\bar{H}) * P(\bar{H})}$$

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:

$$O = \frac{P}{1-P}; P = \frac{O}{O+1}$$

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:

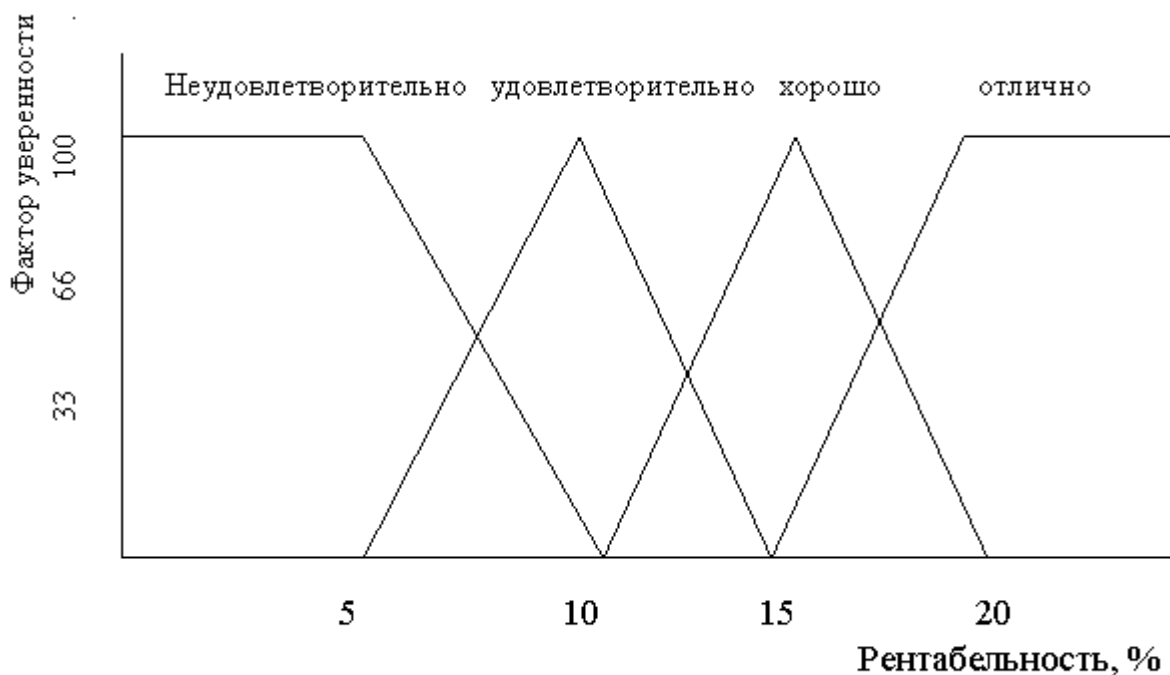
$$O(H/E_1, E_2) = 1 * 10 * 100 = 1000;$$

$$P(H / E_1, E_2) = \frac{1000}{1001} = 0,99$$

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 не обязательно 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. (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): $\min(\text{cf}_A, \text{cf}_B)$ или $\text{cf}_A * \text{cf}_B / 100$

Дизъюнкция (A или B): $\max(\text{cf}_A, \text{cf}_B)$ или $\text{cf}_A + \text{cf}_B - \text{cf}_A * \text{cf}_B / 100$

Объединение факторов уверенности в посылках правил осуществляется чаще всего, 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" cf90

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. 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 производственные 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"

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 объектных методах представления знаний.

Семантические сети. Объектные методы представления знаний берут начало от семантических сетей, в которых типизируются отношения между объектами. Элементарной единицей знаний в семантической сети служит триплет, в котором имя предиката представляет помеченную дугу между двумя узлами графа, соответствующими двум связанным объектам.

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. 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.

Объектно-ориентированная модель, аналогичная во многих отношениях фреймовой модели, также предусматривает инкапсуляцию процедур в структуры данных и механизм наследования. Отличия заключаются в четком различии понятий класс объектов и экземпляр объекта, а также в способе активации процедур к объектам. Для объектно-ориентированной модели характерны такие черты, как скрытие данных и их доступность только через методы (присоединенные процедуры) класса, наследование как атрибутов, так и методов (в последнем случае обеспечивается необходимый уровень абстракции данных и полиморфизм использования процедур). Обращение к объектам, то есть вызов методов класса, осуществляется либо из внешних программ, либо из других объектов путем отправки сообщений.

Состав знаний ЭС

Рассмотрим вопрос о том, какими (по составу) знаниями должна обладать экспертная система для выполнения стоящих перед ней задач. Не вызывает сомнения необходимость представления знаний, описывающих область экспертизы. Вопрос состоит в том, какие еще знания необходимо представлять в экспертной системе и от чего зависит состав этих знаний.

По-видимому, состав знаний зависит 1) от проблемной области, 2) от структуры экспертной системы, 3) от требований и целей пользователей и 4) от языка общения.

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

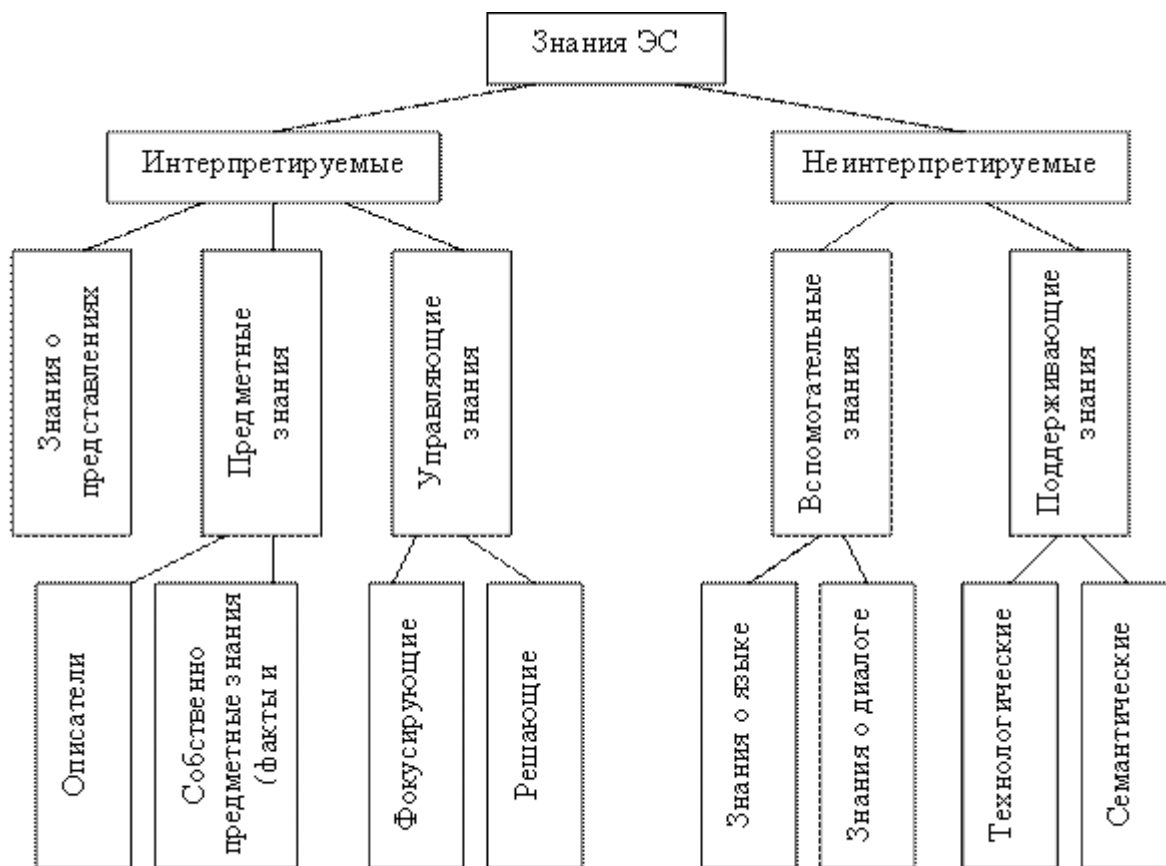
- 1) знания о процессе решения задачи, (т.е. управляющие знания), используемые интерпретатором;
- 2) знания о языке общения и способах организации диалога, используемые лингвистическим процессором;
- 3) знания о способах представления и модификации знаний, используемые компонентой приобретения знаний (структурные);
- 4) поддерживающие структурные и управляющие знания, используемые объяснительной компонентой.

Рассмотрим классификацию знаний с точки зрения проблемной области и с точки зрения архитектуры экспертной системы. Предметную область можно характеризовать совокупностью объектов, характеристик объектов и отношений между объектами. В зависимости от особенностей предметной области данные об образующих ее сущностях могут быть точными/ошибочными, приблизительными, многозначными и полными/неполными. Следует отметить, что, строго говоря, данные всегда неточны, но системы этот факт зачастую игнорируют. Знания, хранимые экспертной системой, тоже, как правило, неточны в силу их экспертного характера. Ошибочность/приблизительность/многозначность данных и знаний (правил, модулей) приводит к тому, что экспертная система имеет дело не с одной, а как бы с несколькими альтернативными областями,

представляемыми в системе в виде альтернативных пространств. При этом в задачу экспертной системы может входить либо рассмотрение всех возможных альтернативных пространств и выдача всех решений, либо рассмотрение всех пространств и выбор некоторых (одного или нескольких) решений. Неполнота данных и знаний приводит к необходимости использовать не одну, а несколько моделей и не один, а несколько источников знаний. Наличие нескольких моделей приводят к представлению в экспертной системе одной предметной области в виде нескольких взаимно дополняющих пространств

Необходимо отметить, что точность/ошибочность, полнота/неполнота, статичность/динамичность данных и знаний являются слабо зависимыми свойствами проблемной области, что приводит к произвольному сочетанию этих свойств. Могут быть точные статические области, неточные динамические области, альтернативные статические и (или) динамические области и т.п.

Все знания делятся 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 структуру и содержание. Если эти знания и используются какой-либо компонентой системы, то она не "осознает" этих знаний. Не интерпретируемые знания подразделяются на вспомогательные знания, хранящие информацию о лексике и грамматике языка общения и 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 исполняют 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 операционные знания. Факты определяют возможные значения сущностей и характеристик предметной области. Операционные знания содержат информацию о том, как можно изменять описание предметной области в ходе решения задач. Говоря другими словами, операционные знания - это знания, задающие процедуры обработки. Однако мы избегаем использовать термин процедурные знания (вместо операционные), так как хотим подчеркнуть, что эти знания могут быть заданы не только в процедурной форме, но и в декларативной.

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 гипотезах и о знаниях, которые наиболее целесообразно использовать при проверке соответствующих гипотез. В первом случае внимание фокусируется на элементах рабочей памяти, во втором - на элементах базы знаний. Решающие знания содержат информацию, используемую для выбора

способа интерпретации знаний, подходящего к текущей ситуации. Эти знания используются 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 использоваться 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.

Организация знаний в ЭС

В данном разделе будут рассмотрены вопросы организации (структурирования) знаний, не зависящие от используемой модели представления знаний. Выделим следующие аспекты проблемы организации знаний: 1) организация знаний по уровням представления и по уровням детальности; 2) организация знаний в рабочей памяти; 3) организация знаний в базе знаний.

Организация знаний по уровням представления.

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. Говоря другими словами, экспертная система должна иметь знания о том, как представлены ее знания о проблемной области. Если знания о проблемной области 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 отсылок к знаниям нулевого уровня, знания первого уровня независимы от проблемной области.

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: 1) общую организацию знаний, 2) логическую организацию и 3) физическую организацию отдельных структур 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 памяти. 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 и уровень данных предметной области (уровень решений).

В существующих 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. 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. В случае РП, состоящей из изолированных элементов, операцию сопоставления данных правилам называют операцией сопоставления, а в случае РП, содержащей сложный элемент, эту операцию называют операцией классификации. Отметим, что алгоритм классификации более сложен.

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. Это приводит к появлению в РП объектов со списками свойств, содержащими имена характеристик и их значения. Так, например, используется тройка: "объект - атрибут - значение".

В случае РП, состоящей из сложного элемента, связь между отдельными объектами указана 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 знаний. Рассмотрим организацию предметных и управляющих знаний. Особое внимание уделяется именно этим знаниям по следующим причинам: во-первых, это основные знания системы как по значимости, так и по объему; во-вторых, именно эти знания оказывают наибольшее влияние на качество и эффективность работы системы; в-третьих, вопросы организации неинтерпретируемых знаний не представляются сложными в связи с простыми способами обработки этих знаний.

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. С точки зрения доступа к знаниям можно выделить три аспекта проблемы связность знаний (и, в частности, данных); механизм доступа к знаниям и способ сопоставления.

Связность (агрегация) знаний является основным способом, обеспечивающим ускорение поиска релевантных знаний. Большинство специалистов пришло к убеждению, что знания следует организовывать вокруг наиболее важных объектов (сущностей) предметной области. Все знания, характеризующие некоторый объект области связываются и представляются в виде отдельного блока (unit). При подобной организации знаний, если системе потребовалась информация о некоторой сущности, то она ищет блок, описывающий эту сущность, а затем уже внутри блока отыскивает информацию о данной сущности. В блоках целесообразно выделить два типа связей между элементами: внешние и внутренние. Внутренние связи объединяют элементы в единый блок и предназначены для выражения структуры блока. Внешние связи отражают взаимозависимости, существующие между блоками 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 уместных знаний. Введение ассоциативных связей обосновывается тем, что логические связи не могут успешно решать и задачи семантической обработки (понимания и вывода), и задачи поиска.

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 назначения, 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 объекта, имеющемуся в рабочей памяти, найти в базе знаний блоки, удовлетворяющие этому описанию. Очевидно, что упорядочение и структурирование знаний может значительно ускорить процесс поиска. Нахождение нужных 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 нужных объектов. На втором этапе путем выполнения операции сопоставления потенциальных кандидатов с описаниями кандидатов осуществляется окончательный выбор нужных объектов. 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. Данная операция 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. Иногда операция сопоставления может быть использована для коррекции, осуществляемой на основе вида неудачи, возникшей при сопоставлении.

Операции сопоставления весьма разнообразны. Можно выделить следующие их формы: синтаксическое сопоставление, параметрическое сопоставление, **семантическое сопоставление и принуждаемое** сопоставление. В случае синтаксического сопоставления соотносят формы (образцы), а не содержание объектов. Термин "форма" (образец) употребляется в классическом смысле. Чтобы упростить процесс сопоставления, все образцы хранятся в едином каноническом представлении. Успешным считается 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, накладываются 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.

№6. Схема функционирования управляющей компоненты

В данной главе будет рассмотрена работа управляющей компоненты, т.е. вопросы о том, как осуществляет решение задачи сама ЭС. Необходимость использовать в ЭС нетрадиционные методы управления вызвана в первую очередь неформализованностью решаемых ими задач. Особенности неформализованных задач с точки зрения организации управления приводят к тому, что процесс решения таких задач не удастся представить в виде детерминированной последовательности программных модулей. Здесь в некоторый текущий момент к исполнению пригодно несколько модулей (или один модуль, но над разными данными), причем не существует надежной информации, позволяющей предпочесть один модуль другому. Задача управляющей компоненты состоит в том, чтобы обеспечить функционирование системы в подобных условиях. Неформализованную задачу можно характеризовать двумя величинами: 1) средним числом модулей, готовых к исполнению в текущий момент; 2) средней мощностью каждого модуля, готового к исполнению. Степень неформализованности задачи прямо пропорциональна среднему числу модулей, готовых в текущий момент к исполнению, и обратно пропорциональна средней мощности. В традиционном программировании модули (программы) вызываются по имени, поэтому программист в ходе составления и отладки программы должен выявить множество всех мыслимых ситуаций, которые возникнут в ходе работы общей программы при различных входных данных; и в каждой точке, где завершается работа одного модуля, в явном виде (указав имя модуля и перечень используемых им данных) запрограммировать однозначный переход к очередному модулю. Такая организация управления не позволяет решать неформализованные задачи.

Основное отличие управляющей компоненты экспертных систем от традиционных механизмов управления состоит в следующем: 1) отдельные модули вызывают не по имени, а по описанию ситуации; 2) способ взаимосвязи модулей формируется в процессе решения задачи, так как выбор очередного модуля зависит от текущей ситуации и не может быть сформирован заранее.

Классическая схема управления экспертной системой

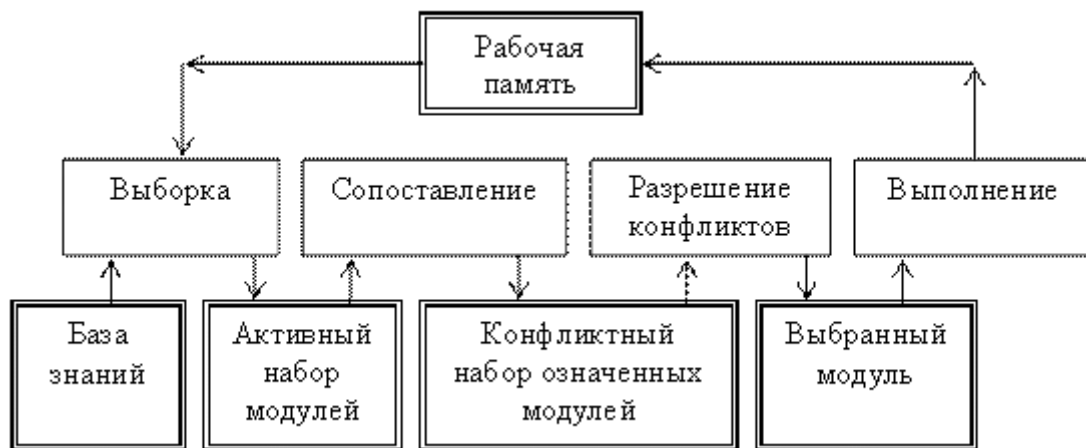
Общая схема функционирования управляющей компоненты экспертной системы приведена на рис.

Управляющую компоненту экспертных систем часто называют интерпретатором. Задача интерпретатора состоит в том, чтобы на основании текущего состояния рабочей памяти

определить, какой модуль и с какими данными будет работать, т.е. выработать на очередном цикле работы управляющее и информационное взаимодействия модулей.



По окончании работы текущего модуля интерпретатор проверяет условия окончания задачи, и если они не удовлетворены, то выполняется очередной цикл. Модули хранятся в базе знаний экспертной системы. Модули обычно реализуются в виде программ или правил. Каждый модуль снабжается образцом, т.е. описанием, указывающим, при выполнении каких условий этот модуль может приступить к



работе.

В общем случае работа интерпретатора в каждом цикле состоит в последовательном выполнении четырех этапов: выборки, сопоставления, разрешения конфликтов, выполнения (действия). Иногда цикл работы интерпретатора делят на два этапа: распознавание и действие, в этом случае в этап распознавания включают выборку, сопоставление и разрешение конфликтов. При этом говорят, что задача "расознавания" состоит в разрешении конфликта.

С точки зрения теории работа интерпретатора зависит только от состояния рабочей памяти и от состава базы знаний. На практике обычно учитывается история работы интерпретатора, т.е. поведение интерпретатора в предшествующих циклах. Информация о поведении интерпретатора запоминается в так называемой памяти состояний интерпретатора. Обычно память состояний содержит в том или ином виде протокол работы системы. В общем случае каждый из этапов использует в своей работе три источника знаний: 1) рабочую память; 2) базу знаний; 3) память состояний интерпретатора. Каждый из этапов направляет свою работу с помощью стратегий, т.е. с помощью определенного набора эвристических правил. Возможности стратегий по управлению процессом функционирования зависят от того, какие функции интерпретатора встроены в него жестко, а какие могут изменяться. Встраивание определенных функций в интерпретатор повышает эффективность его работы, но ограничивает степень воздействия на процесс функционирования. Как правило, в интерпретатор встраивают общую схему поиска решения (т.е. метод), а через стратегии управляют деталями поиска.

Для более глобального влияния на процесс управления необходимо иметь возможность выбирать ту или иную схему интерпретации. Общая схема работы интерпретатора в процессе его работы может изменяться.

На этапе выборки осуществляется определение подмножества элементов рабочей памяти и подмножества модулей базы знаний, которые могут быть использованы в текущем цикле. Иногда этап выборки делается один раз на несколько следующих друг за другом циклов. При реализации этапа выборки обычно используется один из двух подходов.

Первый подход, называемый иногда синтаксической выборкой, выполняет грубый отбор знаний (данных и/или модулей), которые могут быть полезны в текущем цикле. Основанием для выборки знаний в данном случае являются формальные (синтаксические) знания, встроенные в систему разработчиком.

Второй подход, называемый **иногда семантической выборкой**, осуществляет отбор знаний на основании таких сведений, как: модель предметной области, разбиение задачи на подзадачи, текущие цели и т.п. Семантические знания, используемые на этапе выборки, вводятся в систему экспертом, например, в виде метаправил. В результате работы этапа выборки происходит выделение активного набора данных и активного набора модулей, т.е. осуществляется фокусирование внимания системы на определенном ограниченном количестве данных и модулей.

На этапе сопоставления определяется, какие активные модули и на каких активных данных готовы к работе. Модуль готов к работе, если среди активных данных есть данные, удовлетворяющие условиям этого модуля, указанным в его образце. Такие модули называются означенными. Результатом работы этапа сопоставления является набор означенных модулей. Набор означенных модулей часто называют конфликтным набором, подчеркивая этим тот факт, что к работе готовы все модули набора, но интерпретатор не знает еще, какой из них предпочесть. Теоретически сопоставление выполняется в каждом цикле работы интерпретатора над всеми активными знаниями, т.е. образцы всех активных модулей сопоставляются со всеми активными данными. На практике в целях повышения эффективности все означивания не вырабатываются заново на каждом очередном цикле.

На этапе разрешения конфликтов интерпретатор выбирает из конфликтного набора то означивание, которое будет выполняться в текущем цикле. На данном этапе интерпретатор оценивает означенные модули с точки зрения их полезности при достижении текущей цели. Подчеркивая этот факт, данный этап иногда называют этапом планирования.

На этапе выполнения осуществляется исполнение модулей, выбранных этапом разрешения конфликтов. В ходе этого этапа осуществляется модификация рабочей памяти, выполняются операции ввода/вывода и изменяется память состояний интерпретатора.

При реализации общей схемы интерпретации используются две основные архитектуры. Различия в реализации общей схемы являются в первую очередь следствием различной сложности используемых модулей. В одном подходе модулями являются относительно небольшие автономные фрагменты знаний, представляемые в виде правил (в частном случае - в виде продукционных правил), которые понятны пользователю (эксперту), незнакомому с программированием. Этот подход часто называют подходом, использующим управляемые образцами правила, а системы, основанные на данном подходе, системами, управляемыми правилами. Отметим, что частным видом таких систем являются продукционные системы.

Во втором подходе в качестве модулей используются большие сложные автономные фрагменты знаний, представленные в виде программ, смысл которых, конечно, не может быть понятен непрограммисту. Этот подход называют подходом, использующим управляемые образцами модули. Оба подхода используют управление, основанное на сопоставлении образцов, т.е. по окончании работы одного модуля его преемником является один из тех модулей, образцы которых будут означены элементами рабочей памяти. В обоих подходах взаимосвязь модулей (информационная и управляющая) осуществляется через общую память.

Во втором подходе используется несколько отличная терминология. Рабочая память называется "доской объявлений", конфликтный набор называется агендой. Программы, разрешающие конфликты, называются политическими модулями. Модули называются источниками знаний. Каждый источник знания имеет образец. Если образец некоторого источника знаний сопоставляется с данными на "доске", то этот означенный источник знания заносится в агенду. Агенда представляет собой упорядоченный список работ, готовых к выполнению. Под работой понимается источник знания с описанием данных, которые он может обрабатывать в текущий момент. В каждый момент времени с агендой работает один из политических модулей. Выбор политического модуля зависит от обрабатываемой гипотезы. Политический модуль переупорядочивает агенду и выбирает некоторую работу на исполнение. Результатом выполнения работы является изменение содержимого "доски". Политические модули, являясь источниками знания, также заносятся в агенду и выбираются на исполнение. Множество политических модулей обеспечивает разнообразие способов выработки управляющих решений. Модификация систем подобного типа достигается за счет независимости источников знания.

Концепция управляемых образцами модулей позволяет решать более сложные задачи и строить более эффективные системы. Однако этот подход затрудняет реализацию объяснительных способностей и способностей по приобретению новых знаний. Использование в данном подходе больших фрагментов знаний связано с разработкой для каждой проблемной области своих политических модулей, осуществляющих детальное планирование и использование знаний. Кроме того, возможности данного подхода к решению задач различных классов ограничены номенклатурой имеющихся модулей и способами их взаимодействия, предусмотренными политическими модулями.

Концепция управляемых образцами правил позволяет (за счет ограниченной сложности используемых фрагментов знания и представления их в понятном для пользователя виде) решать разнообразные задачи, обеспечивая развитые объяснительные способности и способности по приобретению знаний. Однако ограниченная сложность правил приводит к "комбинаторному взрыву" при решении сложных задач.

Стратегии как механизм управления

Основным механизмом, обеспечивающим разнообразное управление в рамках общей схемы работы интерпретатора, являются стратегии. Перед тем как перейти к описанию отдельных этапов, детализируем понятие "стратегия" и приведем классификацию стратегий. Стратегии можно рассматривать по крайней мере с трех точек зрения: 1) как средство разрешения конфликтов; 2) как способ представления метазнания; 3) как средство повышения эффективности метода, встроенного в интерпретатор.

В первом приближении стратегией можно называть знания о том, какой модуль следует выбрать при наличии нескольких модулей, пригодных к работе. В более общих терминах стратегии можно рассматривать как метазнания о том, как и когда использовать различные

источники знаний объектного уровня (т.е. знания о предметной области). Метазнания могут выражать знания о разбиении задачи на подзадачи, знания о кооперации источников знания, знания о наличии различных стратегий поиска. Необходимо подчеркнуть важность явного задания подобных знаний, т.е. задания знаний в такой форме, которую система может анализировать. Явное задание знаний обеспечивает гибкое поведение системы. Кроме того, стратегии традиционно рассматривают как средство повышения эффективности некоторого общего метода. В более общих терминах стратегии можно рассматривать как любые знания о том, как, когда и какие модули (правила) использовать, т.е. стратегии можно рассматривать не только как средство оптимизации некоторого метода, но и как средство для выбора (и даже для определения) метода.

Приведем классификацию стратегий, используемых в экспертных системах, по следующим параметрам: общность; явность; содержание.

С точки зрения общности стратегии можно классифицировать следующим образом:

- 1) стратегии, не зависящие от способа представления знаний (стратегии поиска от целей или от данных);
- 2) стратегии, не зависящие от проблемной области;
- 3) стратегии, учитывающие специфику проблемной области;
- 4) стратегии, учитывающие специфику цели.

Стратегии, учитывающие специфику проблемной области, имеют более ограниченное применение, чем стратегии первых двух типов, однако именно они позволяют использовать знания о конкретной области для получения качественных и эффективных решений. Стратегии, учитывающие специфику цели, позволяют управлять процессом решения в зависимости от текущих задач системы.

С точки зрения явности стратегии можно разделить на заданные явно и неявно. Стратегия задана явно, если она может быть идентифицирована как отдельная сущность системы, т.е. смена стратегии проходит для системы безболезненно.

Не касаясь специфики проблемной области, содержание стратегии можно охарактеризовать по крайней мере тремя независимыми параметрами: 1) масштабом; 2) составом знаний, используемых стратегией; 3) "полезностью" стратегии.

Параметр "масштаб" делит все стратегии на локальные и глобальные. Локальными стратегиями будем называть те стратегии, которые определяют поведение интерпретатора в текущем цикле его работы. Глобальными стратегиями будем называть те стратегии, которые определяют некоторую линию рассуждений интерпретатора, т.е. последовательность выполняемых (предполагаемых к выполнению) циклов (шагов).

Параметр "состав используемых знаний" делит знания на два класса: 1) знания о текущем цикле работы интерпретатора; 2) знания об истории работы интерпретатора. Заметим, что теоретически для работы экспертной системы достаточно только текущих знаний, однако на практике в целях повышения эффективности систем используют и знания об истории работы.

Параметр "полезность" подразделяется на индивидуальную и сравнительную полезность. Индивидуальная полезность характеризует некоторое знание само по себе вне сравнения его с другими знаниями. Сравнительная полезность характеризует ценность некоторого знания в сравнении с другим знанием.

В связи с тем, что перечисленные выше параметры стратегий являются независимыми, они могут встречаться в произвольных сочетаниях. Необходимо подчеркнуть, что все приведенные сочетания параметров используются на практике.

Этап выборки

Задача данного этапа состоит в сокращении количества возможных путей поиска решения за счет того, что в каждом цикле работы интерпретатора рассматриваются не все, а только некоторая часть данных и правил (модулей), называемых активными данными и правилами. Некоторые авторы объединяют этап выборки с этапом сопоставления. Можно сказать, что эффект выборки состоит в фокусировании внимания системы на каждом шаге работы интерпретатора на тех данных и правилах, которые кратчайшим путем ведут к решению.

На этапе выборки осуществляют отбор объектов следующих видов: данных, правил, метаданных и метаправил. Обычно на каждом цикле осуществляется отбор как среди данных (метаданных), так и среди правил (метаправил). Осуществляя выбор данных, система фокусирует свое внимание на целях (гипотезах), а осуществляя выбор правил, — на способах обработки данных.

Обычно выделяют два типа выборки: простую выборку и иерархическую выборку. Простая выборка характеризуется тем, что выбираемые сущности рассматриваются как сущности одного уровня. При появлении нового элемента в рабочей памяти те правила, которые содержат этот элемент в условии правила (при поиске от данных), помечаются как активные. При удалении элемента из памяти метки у соответствующих правил снимаются. Выборка в данном случае сводится к выбору из всего множества правил тех, которые помечены. Обычно, если в системе используется простая выборка, то этап выборки рассматривают как составную часть этапа сопоставления.

При иерархической выборке объекты (правила, данные) разбиваются на иерархические подмножества (классы). Выборка в данном случае состоит в использовании метаправила для выбора одного из классов. При этом классы могут быть как непересекающиеся, так и пересекающиеся. Здесь метаправила проверяют условия X, Y и на основании их значений выбирают соответствующее подмножество правил (класс). Предполагается, что символы в условии правил связаны конъюнктивно.

Следует отметить, что введение иерархии правил неизбежно влечет за собой и иерархию данных, что, к сожалению, не всегда явно признается разработчиками систем. Действительно, метаправила, в отличие от правил, применяются не к объектам предметной области, а к метаданным, т.е. уместно говорить о появлении в рабочей памяти данных и метаданных. В системах, использующих управляемые образцами модули, указанная иерархия данных на "доске" (аналоге рабочей памяти) осуществляется явно путем введения различных уровней.

Объекты, подлежащие выборке на текущем цикле, задаются либо по имени, либо по описанию свойств. При задании по имени указывается либо перечень объектов (данных, правил), либо перечень имен классов, описывающих объекты. При задании объектов через описание свойств указывают не имена объектов (классов объектов), а перечень свойств, которыми эти объекты должны обладать.

На этапе выборки используются довольно простые операции: замена, добавление, удаление, упорядочивание и т.п. Операция замены состоит в замене одних активных объектов (данных, правил) или классов объектов на другие. Операция добавления

(устранения) состоит в добавлении к перечню (устранении из перечня) активных объектов тех объектов, которые указаны в текущей выборке. Операция упорядочивания состоит в изменении порядка активных объектов (данных, правил), что влияет на порядок их обработки и как следствие на получаемый конечный результат.

Этап сопоставления

Задача данного этапа состоит в том, чтобы сопоставить выбранное множество активных правил (P_u) выбранному подмножеству элементов рабочей памяти (F_v) и определить конфликтный набор правил, т.е. определить, какие правила из P_u и на каких данных из F_v удовлетворены. Напомним, что правило удовлетворено, если выполнены (являются истинными) требования, указанные в условии правила. Следует подчеркнуть, что одно правило может входить в конфликтный набор несколько раз, так как оно может быть удовлетворено на различных данных (элементах рабочей памяти).

Операция сопоставления требует значительных вычислений, так как для

определения конфликтного набора надо проверить условия всех активных правил на всех сочетаниях активных данных. Сложность операции сопоставления усугубляется тем, что условия правил могут содержать не только константы, но и переменные. Для проверки таких условий необходимо сопоставлять с ними все возможные значения из рабочей памяти. В связи с тем, что скорость работы является одной из главных проблем экспертных систем, необходимо обеспечить эффективность операции сопоставления в условиях большого количества правил и данных.

Для того чтобы конкретизировать описание процесса сопоставления, выберем некоторый способ представления правил и данных. Будем полагать, что рабочая память содержит объекты и связанные с ними пары "свойство - значение". Объект вместе с его парами "свойство — значение" будем называть элементом рабочей памяти. Правило будем представлять в виде "условие \rightarrow действие". Условие состоит из последовательности образцов, которые могут рассматриваться как частичные описания элементов рабочей памяти. Если некоторый образец P описывает некоторый элемент E , то говорят, что P сопоставляется E . Во введенном представлении условие правила удовлетворяется, если все образцы сопоставляются с элементами рабочей памяти (с учетом логических связей между ними).

Будем описывать операцию сопоставления в предположении, что правила применяются слева направо, т.е. вывод осуществляется от данных. Отметим, что если действия правил представляются в виде образцов, то практически все приведенные ниже рассуждения применимы и для вывода от целей.

Способы ускорения операции сопоставления. Наиболее общий подход к ускорению операции сопоставления состоит в комбинировании двух процессов: 1) процесса индексирования (выборки) и 2) процесса интерпретации (означивание) условий правил. В простых областях индексирование может заменять операцию выборки. Простейшая форма индексирования состоит в том, что интерпретатор извлекает одну или более характеристик (например, свойств) из каждого элемента рабочей памяти и использует их для того, чтобы выделить подмножество правил, условия которых могут быть удовлетворены. Затем интерпретатор исследует условия этих выделенных правил и определяет те из них, которые удовлетворяются. В более эффективных формах индексирования с каждым образцом в правиле связывается счетчик. В начале работы значения всех счетчиков равны нулю. При введении некоторого элемента в рабочую память функция "индексирования" определяет,

какие образцы сопоставляются с этим элементом, и добавляет к их счетчикам по единице. При устранении некоторого элемента из рабочей памяти единица вычитается из тех счетчиков, образцы которых сопоставлялись с этим элементом. При выполнении сопоставления интерпретатор рассматривает только те правила, в условии которых все образцы имеют не нулевые значения счетчиков. Подобный подход к операции сопоставления находит широкое применение.

Неэффективность простейших алгоритмов сопоставления является следствием того, что в каждом цикле работы интерпретатора для получения конфликтного набора заново осуществляется просмотр рабочей памяти и памяти правил. Очевидно, что значительная часть означиваний, полученных на предыдущем цикле работы интерпретатора, может быть сохранена и использована в последующем цикле. Задача состоит в том, чтобы избежать на каждом цикле повторных просмотров рабочей памяти и памяти правил.

Для того чтобы избежать излишних просмотров рабочей памяти, необходимо запомнить определенную информацию и передавать ее от цикла к циклу. Шаг, который может потребовать повторения, состоит в определении для некоторого образца, того какие элементы рабочей памяти ему соответствуют. Для ответа на этот вопрос простейший интерпретатор будет сопоставлять данный образец со всеми активными элементами рабочей памяти. Повторных просмотров можно избежать, связав с каждым образцом список элементов, с которыми он сопоставляется. Этот список должен модифицироваться при каждом изменении рабочей памяти. Если некоторый элемент вводится в рабочую память (или модифицируется), то интерпретатор находит все образцы, которые сопоставляются с этим элементом, и добавляет его к спискам соответствующих образцов. При удалении некоторого элемента из рабочей памяти интерпретатор снова находит все образцы, которые сопоставляются с этим элементом, и устраняет этот элемент из списков образцов. Таким образом, интерпретатор, запоминая указанную информацию, никогда не просматривает рабочую память. Для того чтобы избежать лишних сопоставлений, образцы правил организуются в сетевую структуру, которая выполняет функции индексирования правил. Сеть образцов представляет собой разновидность дискриминационной сети. В вершинах дискриминационной сети проверяются характеристики элементов. В зависимости от результатов проверок признак, поступивший на вход сети, пройдет через сеть по тому или иному пути (путям) и как результат сообщит, какие правила удовлетворяют этому признаку. Сеть составляется специальной программой (компилятором) на основе анализа условий правил, хранимых в рабочей памяти. Если после всех проверок некоторый признак попадает в некоторую терминальную вершину, то данный признак будет содержать означивание соответствующего правила. Признак, полученный терминальной вершиной, содержит информацию об изменении конфликтного набора.

Этапы разрешения конфликтов и выполнения

Результатом этапа сопоставления является конфликтный набор, т.е. множество означиваний тех правил, которые удовлетворены в текущем цикле. В ходе этапа разрешения конфликтов интерпретатор выбирает одно или несколько означиваний, которые должны быть выполнены в текущем цикле.

Критерии оценки адекватности стратегий разрешения конфликтов. Для того чтобы система могла демонстрировать разумное поведение в динамически изменяющемся окружении, она должна удовлетворять по крайней мере двум требованиям. Во-первых, система должна быть способна реагировать на изменение окружения, т.е. система должна быть способна обнаружить новые требования окружения, решить, какие действия необходимы для удовлетворения новым требованиям, и выполнить необходимые действия. Во-вторых,

система должна быть способна к обучению (к приобретению новых знаний) в тех случаях, когда возникают новые аспекты окружения или когда становится ясно, что предыдущее поведение не является адекватным.

Экспертная система "осознает" изменение окружения только на фазе распознавания цикла "распознавание-действие", поэтому для удовлетворения требованию "реагировать на окружение" количество означенных правил в каждом цикле должно быть ограничено. Возможно наиболее очевидный способ ограничить количество означиваний состоит в выделении взаимоисключающих ситуаций для правил. Однако это эквивалентно указанию области применения каждого правила, что ограничивает способности системы к обучению. Для того чтобы система могла добавлять правила в рабочую память, не требуя модификации введенных ранее правил, правила должны быть в значительной степени автономны. Наличие конфликтов является неизбежной платой за автономность правил. Механизм разрешения конфликтов позволяет поддерживать автономность правил за счет того, что интерпретатор хранит глобальные знания, т.е. знания о нескольких (о всех) циклах работы.

В связи с тем, что сложность правила ограничена, возникает необходимость координировать действия нескольких правил, так как требуемое поведение системы сложнее действия некоторого одного правила. Наиболее очевидным способом координации является явное указание для каждого правила его преемника. Однако этот способ противоречит автономности правил. Выход, предлагаемый механизмом разрешения конфликтов, состоит в том, что указывается только ЧТО должно быть сделано, а не КТО должен делать. Отсутствие явного указания на того, кто должен делать, не ограничивает возможностей по приобретению новых правил.

Говорят, что система обладает свойством "чувствительности", если она способна реагировать на требования окружения. Если система демонстрирует преемственность поведения, то говорят о "стабильности". Можно сказать, что задача механизма разрешения конфликтов состоит в обеспечении свойств "чувствительности" и "стабильности" без потери свойства автономности правил. Понятие "чувствительность" можно уточнить через множество характеристик, которым должна удовлетворять система для того, чтобы функционировать в динамическом окружении. Перечислим эти характеристики:

Интерпретатор должен помогать системе обеспечивать "чувствительность" к различным аспектам окружения, в то время как система фокусирует свое внимание на решении частной задачи, т.е. интерпретатор не должен терять "чувствительность", обеспечивая "стабильность".

2. Интерпретатор должен помогать системе обеспечивать "чувствительность" к ее собственным действиям. Например, система должна быть чувствительной к заикливаниям.

3. Интерпретатор должен помогать системе осуществлять "разумное" поведение при конфликтных данных в рабочей памяти, т.е. интерпретатор должен отличать релевантную информацию от информации, которая более не является релевантной,

4. При наличии нескольких означиваний, пригодных для текущей ситуации, интерпретатор должен выбрать некоторое целесообразное действие.

5. Интерпретатор должен избегать принятия случайных решений, т.е. выбор одного означивания из многих должен осуществляться без использования механизма случайности.

Хотя "чувствительность" и "стабильность" не содержат явно противоречивых требований, достижение значительных успехов в одном из этих свойств обычно ослабляет другое. В частности, если "чувствительность" не потеряна, то разработчик экспертной системы должен очень тщательно распределить ответственность за "стабильность" между интерпретатором и

системой. Конструктор не может возложить всю ответственность за стабильность на интерпретатор без потери чувствительности. Для обеспечения стабильности целесообразно определить формы необходимой координации между означиванием и минимальную поддержку, обеспечиваемую интерпретатором для выполнения этих форм. Формы координации, используемые в экспертных системах, подобны основным управляющим конструкциям, используемым в языках программирования.

Стратегии разрешения конфликтов. В этом пункте будут рассмотрены стратегии пяти классов, широко используемые в экспертных системах, которые отличаются по селективности или по используемым ими источникам знаний. Стратегии значительно различаются по селективности: одни устраняют только малую часть означиваний из конфликтного набора, другие устраняют почти все означивания. Все рассмотренные ниже стратегии используют один или несколько из следующих источников знаний: рабочая память; память правил; память состояний, поддерживаемая интерпретатором. Рабочая память хранит элементы данных и некоторую информацию о них (например, относительный возраст элементов). Память правил содержит правила и некоторую информацию о них (например, порядок в котором правила введены в систему). Память состояний интерпретатора хранит историю работы, т.е. информацию об удовлетворенных правилах и о подмножестве данных, которые означивали правила.

Выделим следующие классы стратегий: 1) стратегии упорядочивания правил; 2) стратегии специальных случаев; 3) стратегии возраста элементов; 4) стратегии различий; 5) случайные стратегии.

Стратегии упорядочивания правил используют в качестве критерия для выбора означиваний из конфликтного набора приоритеты, приписанные правилам, т.е. источником знаний для стратегий этого класса является память правил. Между правилами может быть установлено отношение общего порядка, например правило P1 доминирует над всеми остальными, P2 доминирует на всеми правилами, кроме P1, и т.д. Использование стратегий упорядочивания правил требует определить механизм назначения приоритета для каждого вновь вводимого правила.

Стратегии специальных случаев используют в качестве критерия выбора означиваний отношение "специальный случай". В связи с тем, что для многих пар означиваний отношение "специальный случай" не выполняется, данные стратегии обладают малой селективностью. Источником знаний в стратегиях этого класса может быть память правил, рабочая память и оба эти источника одновременно. Отношение "специальный случай" может определяться по-разному, однако дальнейшие действия после определения специального случая во всех стратегиях этого класса одинаковы. Если означивание KS является специальным случаем означивания KG то KS является предпочтительным. Кроме того, если существует означивание KN, которое не является ни общим, ни специальным случаем для любого другого означивания, то KN является предпочтительным. Стратегии этого класса устраняют только те означивания, которые являются общими случаями некоторых других означиваний. Примером стратегии специальных случаев является стратегия, которая определяет над данными означиваний (т.е. источником знаний является рабочая память) следующее отношение специального случая: означивание K_g является специальным случаем другого означивания KG, содержит все элементы, содержащиеся в KG.

Стратегии возраста используют в качестве критерия выбора время нахождения элементов в рабочей памяти. Обычно используется одна из мер времени: 1) возраст элемента измеряется числом циклов интерпретатора, которые прошли со времени создания этого элемента; 2) возраст элемента измеряется числом действий, которые были совершены после действия,

создавшего этот элемент. Использование первой меры может привести к ситуации, когда два элемента имеют одинаковую меру (т.е. эти элементы были созданы в одном цикле работы интерпретатора разными действиями). При использовании второй меры элементов, имеющих одинаковый возраст, быть не может. Опыт использования экспертных систем показывает, что по ряду различных причин более разумно выбирать "молодые" элементы.

Рассмотрим три возрастные стратегии. Стратегия C-ВОЗР1 использует в качестве меры возраста некоторого элемента число действий, которые были выполнены с момента создания этого элемента. Эта стратегия упорядочивает означивания на основе наиболее молодого элемента, содержащегося в них. Если наиболее молодой элемент одного означивания K1, младше, чем наиболее недавний элемент другого означивания K2, то предпочтение отдается означиванию K1. Так как все означивания могут быть попарно сравнимы, то данная стратегия обладает высокой селективностью. Однако если различные означивания содержат наиболее молодой элемент, то все они будут признаны предпочтительными. Стратегия C-ВОЗР2 упорядочивает означивания по наименее молодому элементу, т.е. эта стратегия предпочитает означивание K_g означиванию

K_G, если наименее молодой (т.е. наиболее старый) элемент из K_s моложе

наименее молодого элемента из K_G. Стратегия C-ВОЗР3 наиболее полно использует информацию о возрасте. Возраст элементов определяется по числу действий, которые были выполнены с момента создания данного элемента. Данная стратегия рассматривает возраст всех элементов, входящих в некоторое означивание. Для того чтобы упорядочить два означивания, сначала сравнивают их наиболее молодые элементы; если они равны, то сравнивают следующие по возрасту элементы и т.д.

Стратегии различий (C-РАЗЛ) используют в качестве критерия выбора подобие (или отличие) означиваний из текущего конфликтного набора тем означиваниям, которые были выполнены в предыдущих циклах. Источником знаний в стратегиях этого класса является память состояний интерпретатора. Рассмотрим две стратегии из этого класса. C-РАЗЛ 1 рассматривает два означивания как различные, если различны правила этих означиваний. Данная стратегия предпочитает те означивания, которые отличаются от означиваний, выполненных в предыдущем цикле. C-РАЗЛ 2 использует более сильный критерий. Эта стратегия считает два означивания различными, если различны их правила или данные. Кроме того, C-РАЗЛ2 использует полную историю системы при выборе предпочтительных означиваний. C-РАЗЛ2 препятствует вторичному использованию одинаковых означиваний.

Случайные стратегии являются нежелательными, но к ним приходится прибегать в тех случаях, когда после применения других стратегий не происходит выбора одного (при использовании одного процессора) означивания.

Завершая рассмотрение способов реализации этапа разрешения конфликтов, необходимо подчеркнуть, что обеспечение адекватного поведения системы в динамическом окружении возможно только при использовании всех источников знаний (рабочей памяти, памяти правил и памяти состояний интерпретатора). Любая отдельная стратегия использует только часть доступных знаний и не может обеспечить гибкое управление в динамическом окружении.

Этап выполнения. На данном этапе выполняются действия правил (при поиске от данных), что обычно приводит к изменению рабочей памяти. Различают три основных вида действий: ввод, вывод, преобразование. Ввод осуществляет изменение рабочей памяти за счет влияния

окружения. Вывод не меняет рабочую память. Преобразование осуществляет над некоторыми элементами рабочей памяти те операции, которые указаны в действии правила.

После выполнения действий осуществляется проверка условий окончания, т.е. определяется, не является ли текущее состояние рабочей памяти целевым. Кроме того, на этапе выполнения осуществляются изменения памяти состояний интерпретатора, соответствующие выполненным действиям.

Развитие традиционной схемы управления

В литературе описан ряд подходов к построению экспертных систем. Кратко рассмотрим два таких подхода. Первый из подходов базируется на использовании множества независимых агентов (акторов). В последнее время снова возник интерес к подобной архитектуре построения систем. Поясним этот подход на примере контрактных сетей. Вершинами контрактной сети являются исполнители (агенты). Каждый из них обладает управляющей информацией о том, какие задачи он способен решить, каковы способы оценки затрат на решение задачи и каковы способы выбора "компаньона" при решении задачи. Вершины сети могут заключать контракты с другими вершинами на правах подрядчика и субподрядчика, причем каждая вершина может одновременно выступать в нескольких ролях в соответствии с заключенными контрактами. Общение между вершинами осуществляется посредством посылки сообщений, состав и номенклатура которых заранее оговорены. Решая задачу, исполнитель, соответствующий вершине сети, выявляет в задаче части, решить которые он не способен. Исполнитель формирует сообщения ко всем вершинам сети, каждое из которых содержит описание задачи с предложением принять участие в ее решении. Другие исполнители определяют, могут ли они принять участие в решении задачи, и если могут, то посылают соответствующее сообщение с предложением услуг и оценкой затрат. Исходная вершина, выступая в роли подрядчика, выбирает среди откликнувшихся вершин лучшего с ее точки зрения исполнителя и заключает с ним контракт на решение задачи. С помощью такого процесса происходит распределение решения задачи. Все вершины контрактной сети независимы, т.е. исходное состояние системы до решения задачи представляется графом, состоящим из изолированных вершин. Все связи между исполнителями устанавливаются только в процессе функционирования системы.

Метод контрактных сетей моделирует работу группы экспертов при решении общей задачи. При этом считается, что каждый эксперт способен решать довольно широкий класс специфических задач. Потенциально подход обладает значительной гибкостью и модифицируемостью.

Второй из рассматриваемых нами подходов обобщает традиционную схему построения в направлении создания так называемой иерархической системы. Общая схема иерархической экспертной системы (ИЭС) приведена на рис. В этой схеме традиционной экспертной системе соответствует ЭС¹. ИЭС можно определить следующим образом:



1. Вертикальная размерность на рис. соответствует иерархии знаний о предметной области, выраженных на различных уровнях абстракции.
2. Горизонтальная размерность на рис. соответствует иерархии метаправил, рассматриваемых как стратегии, используемые для разрешения конфликта.
3. Каждая компонента ЭС соответствует экспертной системе с традиционной структурой.
4. Взаимодействие компонент ограничено взаимосвязью соседних компонент. Взаимосвязь осуществляется только через рабочую память.

Введение вертикальной размерности позволяет естественно представлять различные уровни описания предметной области, такие, как: уровни на "доске" или абстрактные пространства. Введение горизонтальной размерности позволяет естественным образом расширить традиционные ЭС в направлении иерархической организации правил.

№7. Объяснительные способности ЭС

Важность объяснений в ЭС вызвана рядом факторов. Во-первых, трудно ожидать, что пользователи будут знать все возможности и понимать все действия экспертной системы. Во-вторых, значимость объяснений обусловлена тем, что экспертные системы предназначены для использования в слабоформализованных областях, т.е. для решения задачи, не имеющих алгоритмических решений. В условиях отсутствия теории, являющейся надежной гарантией правильности полученных результатов, возникает особая необходимость в разработке средств, дающих пользователям возможность убедиться в достоверности методов и знаний, используемых экспертной системой для получения решения. В качестве таких средств в ЭС используется объяснительный компонент.

Понятие "объяснение" и смежные с ним понятия "обоснование", "определение", "доказательство" изучаются в логике и теории познания. В самом общем плане объяснение трактуется как процесс раскрытия сущности изучаемого объекта путем установления тех связей и отношений, которые определяют его существенные черты. Объяснение предполагает описание объекта и анализ последнего в контексте его связей, отношений и зависимостей. "Объяснение - совокупность приемов, помогающих установить достоверность суждений относительно какого-либо неясного, запутанного дела или имеющих целью вызвать более ясное отчетливое представление о более или менее известном явлении".

Общими логическими характеристиками всякого объяснения считаются: 1) его двусоставность и 2) наличие в нем отношения логического следования. В любом объяснении должны содержаться две части, различающиеся по своим функциям: экспланандум (S) - то, что надлежит объяснить, и эксплананс (R) - совокупность объясняющих положений.

По форме объяснение является выводом S из R. С точки зрения вывода объяснения иногда разделяют на дедуктивные, индуктивные и абдуктивные выводы.

Близким к понятию "объяснение" является понятие "обоснование". "Обоснование — мыслительная процедура, основанная на использовании определенных знаний, норм и установок для принятия каких-либо утверждений, оценок или решений о практических действиях". Целью обоснований является "обсуждение и анализ правомочности и целесообразности определенных действий (решений) в некоторой ситуации, составление возможных альтернатив и выбор из них наиболее эффективных".

Применительно к экспертным системам "обосновать действия системы — это показать, что они являются разумными в рассматриваемой ситуации". "Объяснить - сделать ясным, ответить на вопрос КАК, ПОЧЕМУ; обосновать - подтвердить правильность результата". Как будет показано ниже, в экспертных системах обоснование и объяснение не удается реализовать с помощью одних и тех же знаний.

Объяснительные способности экспертных систем должны быть ориентированы на всех, кто с ними взаимодействует. В этой главе мы будем называть таких людей обобщенными термином "пользователи". Обычно выделяют следующие типы пользователей:

- 1) пользователи, не являющиеся специалистами в области экспертизы, их задача получить от экспертной системы решение некоторой задачи;
- 2) пользователи, являющиеся специалистами в области экспертизы, их задача, используя экспертную систему, сократить трудоемкость получения результата или повысить его качество;
- 3) "студенты", т.е. пользователи, которые с помощью экспертной системы хотят обучиться методам решения задач из области экспертизы;
- 4) эксперты, т.е. высококвалифицированные специалисты, в задачу которых входит обнаружить недостающие знания и ввести их в систему, т.е. осуществить отладку знаний;
- 5) инженеры по знаниям, т.е. специалисты в области инженерии знаний, в задачу которых входит отладка управляющего механизма, анализ и модификация экспертной системы.

Специфика задач, решаемых пользователями разных типов, предъявляет к объяснительным способностям экспертных систем различные требования. Действительно, основная цель использования объяснительных способностей для "студента" - обучение, для эксперта и инженера по знаниям - локализация ошибок, для пользователя-специалиста - обеспечение доверия к результату, для пользователя-неспециалиста - достижение взаимопонимания.

Итак, можно выделить следующие цели, преследуемые при использовании объяснений в экспертных системах: 1) локализовать ошибки системы путем исследования метода рассуждения; 2) повысить доверие пользователя к системе (путем объяснения способа получения результата), что в конечном счете способствует положительной оценке пользователем пригодности системы к практическому использованию; 3) достичь взаимопонимания между пользователем и системой (путем объяснения непонятных терминов, ответов и ситуаций), что повышает вероятность успешного решения поставленной пользователем задачи; 4) обучить пользователя.

Способ объяснения часто зависит от объясняемой сущности; круг объясняемых сущностей как в существующих, так и в перспективных системах не охватывает всех знаний и процедур, и необходимо явно указывать, что может быть объяснено, а что нет. Объясняемые сущности можно классифицировать по следующим аспектам: 1) область интерпретации (предметная область, система, пользователь, язык); 2) уровень общности (конкретные факты, абстрактные факты, метафакты); 3) тип сущности (процесс, объект и т.п.); 4) свойства и значения сущности (статические/динамические, точные/приближенные, полные/неполные и т.д.) и т.п.

Способ объяснения имеет несколько аспектов. Для нас будут существенны следующие три аспекта: 1) тип объяснения; 2) уровень детальности объяснения; 3) язык объяснения. В научной литературе обычно выделяют пять основных типов объяснения: причинные (каузальные), объяснения через закон, функциональные (целевые, мотивационные), структурные и генетические (исторические).

Причинные объяснения вскрывают причинные взаимосвязи между некоторыми явлениями. Объяснить в этом случае - значит вскрыть причину. Важность причинных объяснений вытекает из того, что причинность всеобща, так как нет явлений, которые не имели бы своих причин, как нет явлений, которые не порождали бы тех или иных следствий.

Наиболее развитой формой научного объяснения является объяснение на основе теоретических законов. Объяснение через закон сводится к установлению, в соответствии с каким законом, теорией, моделью возникло или происходило объясняемое явление. В этом случае объяснение можно рассматривать как логическую операцию дедукции: выведение частных следствий из общего закона (теории, модели). Например, газ ведет себя данным образом, потому что он состоит из молекул (поведение которых предсказывает кинетическая теория газа).

Функциональные объяснения сводятся к установлению функций, выполняемых той или иной частью системы. Эти объяснения строятся по принципу: "X нужно для того, чтобы могло произойти Y". Например, "мимикрия нужна насекомым для того, чтобы скрываться от врагов". Одним и тем же явлениям можно давать и причинные объяснения и функциональные. В социальных науках функциональные объяснения преобразуются в целевые (мотивационные), так как в них объектом объяснения оказываются целенаправленные действия человека или группы людей. Действие происходит потому, что человек (искусственная система) стремится к достижению некоторой цели. Сложность получения целевых объяснений заключается в том, что при анализе социальных явлений проявляются действия, а цели остаются скрытыми. В связи с этим целевое объяснение строится в предположении реальности некоторой из возможных целей. Естественно, что такие объяснения не всегда соответствуют действительности.

Структурное объяснение дается посредством описания структуры, которая обеспечивает выполнение функций и поведение объясняемой системы в целом. В отличие от других типов объяснений, использующих преимущественно какие-то отдельные аспекты явлений (причины, цели и т.д.), структурное объяснение призвано воспроизвести ситуацию в целом.

Генетическое (историческое) объяснение состоит в раскрытии условий, причин и законов, приведших к текущему состоянию системы, предмета, явления. Генетическое объяснение вскрывает происхождение сущности и способ ее образования. В существующих экспертных системах, как правило, используются целевые и причинные объяснения.

Параметр "способ объяснения" можно рассматривать и с точки зрения уровня детальности объяснений. Необходимость давать объяснения на различных уровнях детальности вызвана разнообразием целей пользователей, разнообразием их уровня знания, изменением во времени знаний конкретного пользователя и т.п. Очевидно, что на любом уровне детальности объяснения должны адекватно (для данного уровня) отражать объясняемую сущность.

Параметр "способ" включает еще один аспект - "язык объяснения". В экспертных системах объяснения, как правило, осуществляются на ограниченном естественном языке или языке графических образов.

Выделяют следующие подходы к созданию программ, обладающих объяснительными способностями: 1) записанные объяснения, т.е. использование для объяснений заранее подготовленных текстов на естественном языке; 2) генерация объяснений непосредственно из программных кодов или из трека выполнения программы;

Записанные объяснения. Простейший способ получения объяснений о том, что делает программа, состоит в запоминании ответов на все возможные вопросы, сформулированных на естественном языке. Данный подход применим в том случае, если можно предвидеть все вопросы. При этом подходе пользователь получает объяснение точно в том виде, в котором оно было записано. Типичными примерами записанных объяснений являются сообщения программ о стандартных ошибках, контекстная помощь.

Небольшую программу относительно легко снабдить записанными объяснениями. После написания программы с каждой ее частью связываются тексты, объясняющие действие соответствующей части. Когда в некоторый момент работы программы пользователь задает вопрос о том, что она делает, ему выдается текст, связанный с частью программы, выполняемой в этот момент.

Ясно, что, используя данный подход, нельзя обеспечить объяснения, учитывающие разнообразие целей и знаний пользователей.

Генерация объяснений из программ. Реализация объяснительных способностей базируется на двух предположениях. Во-первых, предполагается, что сокращенное описание действий системы может являться эффективным объяснением при правильном выборе уровня детальности, на котором рассматриваются действия. Это предположение упрощает задачу генерации объяснений, так как сводит ее к записи и последующему просмотру последовательности событий (т.е. просмотру правил в дереве целей). Данное предположение является источником основных ограничений. Действительно, не очевидно, что всегда может быть найден подходящий уровень детальности. Для того чтобы сокращенное описание действий могло быть пригодным для объяснений, выбранный уровень детальности должен удовлетворять следующим условиям: 1) достаточной детальности (чтобы действия были понятны); 2) достаточной общности (чтобы уровень содержал действия, значимые для пользователя, и игнорировал излишние детали); 3) достаточной полноты (чтобы цитируемые действия адекватно объясняли поведение системы).

Во-вторых, предполагается, что существует некоторый остов, с позиций которого можно рассматривать действия системы так, чтобы они были понятны пользователю.

Основные идеи, используемые при построении объяснительной компоненты могут быть сформулированы в виде четырех требований:

1. Необходимо определить понятие примитивного действия системы. Это задает минимальный фрагмент программы, подлежащий объяснению, тем самым определяется максимальный уровень детальности.
2. Необходимо снабдить систему способностью запоминать ее поведение на выбранном уровне детальности. Результатом является полный трек поведения системы, т.е. история активации всех правил.
3. Необходимо выбрать остов, с позиций которого трек выполнения может быть понят пользователем. Трек содержит запись поведения ЭС, а остов обеспечивает способ понимания этого поведения. Выбор остова является центральной задачей при конструировании объяснительных способностей. Для ЭС в качестве остова выбрано дерево целей, представляющее собой естественное средство описания поведения системы.

4. Необходимо разработать программу, которая предоставляет пользователю возможность анализировать трек действий системы. Эта программа должна быть способна исследовать трек в терминах целей и правил, способствующих достижению целей. Пользователю должны быть предоставлены команды (например, команды ПОЧЕМУ и КАК), позволяющие просматривать дерево целей, т.е. анализировать предыдущие, текущие и будущие состояния системы. В ходе решения задачи экспертная система может задавать пользователю вопросы. Если вопрос кажется пользователю неуместным, то он может прервать работу системы и обратиться к объяснительным средствам системы, чтобы выяснить, с какой целью система задала этот вопрос. В связи с тем, что объяснения рассматриваются в терминах движения от текущей цели по дереву целей, пользователю предоставлены две основные команды: ПОЧЕМУ и КАК. Ответ на вопрос ПОЧЕМУ соответствует движению вверх по дереву в поисках ближайшей цели, объясняющей "почему" достигается текущая цель. Ответ на вопрос КАК соответствует движению вниз по дереву целей и объясняет, "как" достигалась (будет достигаться) текущая цель. В связи с тем, что дерево целей является И/ИЛИ графом, при движении вниз обычно образуется несколько подцелей. Комбинируя последовательность вопросов ПОЧЕМУ и КАК, пользователь может просматривать дерево целей.

Объяснения могут быть представлены в текстовой, графической или смешанной формах. При использовании текстовой формы выводится список используемых правил, а при графической – дерево целей, в узлах которого содержатся означенные правила.

Обоснование правил. Объяснение (обоснование) смысла правила зависит от содержания правила. Предложена следующая классификация правил: идентифицирующие, причинные, факт мира, факт области экспертизы.

Правила, которые используют идентифицирующие свойства объектов для их классификации, называются идентифицирующими (определятельными). Такие правила основываются на свойствах некоторого класса.

Правила, предпосылки и заключения которых связаны причинными аргументами, называются причинными правилами. Причинное отношение может быть направлено от причины к следствию и от следствия к причине. Предложена следующая классификация причинных правил: 1) корреляция сущностей установлена, но направление зависимости неизвестно; 2) направление зависимости (причинности) известно, но условия процесса непонятны; 3) причинность хорошо понятна.

Правила, выражающие знания о мире, основанные на "здравом смысле", называются правилами о фактах мира. Примером такого правила является следующее правило: "Если пациент мужского пола, то он не может быть ни беременным, ни кормящим грудью".

Правила, выражающие связи гипотез проблемной области, называются правилами о фактах области. Используя этот вид правил, программа может выводить из одних свойств другие, таким образом уменьшая количество информации, запрашиваемой у пользователя.

Разумным объяснением определятельных связей, фактов мира и фактов области является простое объявление соответствующих правил, выражающих эти зависимости. Более сложные обоснования требуются в случае причинных взаимосвязей. Поясним эти сложности на примере следующего правила: "Если пациенту меньше восьми лет, то ему не следует назначать тетрациклин". Это правило не указывает причины, которые лежат в его основе. Таких причин две: 1) данное лекарство откладывается в растущих костях ребенка; 2) лекарство приводит к почернению зубов.

Следует отметить, что эксперт обычно не задумывается над обоснованием предлагаемых им правил. Однако для студента, изучающего правила новой для него области, обоснование необходимо, так как оно структурирует знания и помогает запомнить правило. Обоснование правила осуществляется путем введения промежуточных шагов между предпосылкой и заключением правила. Выбор числа шагов и используемых в них понятий в значительной степени произволен. Для того чтобы объяснить правило, необходимо знать не только промежуточные шаги, но и то, какие шаги необходимо объяснять. Говоря другими словами, необходимо на основании целей и предварительных знаний пользователя определить требуемую ему глубину понимания.

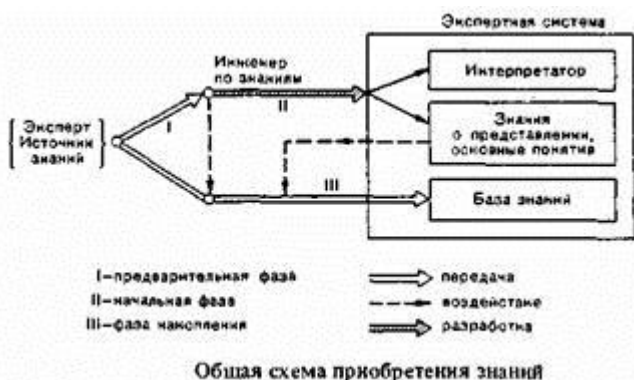
Для ликвидации избыточности объяснений может быть введена мера, которая оценивает информационную ценность правила. Используемая мера оценивается по формуле $N = -\log(CF)$. Для тривиальных правил, имеющих коэффициент определенности, равный единице, $N = 0$ (обычно это правила, содержащие определение терминов). С уменьшением значения КО правила его мера информативности возрастает.

Для того чтобы приспособлять объяснения к различным пользователям, также могут быть введены меры "сложности" и "важности". Мера "сложности" приписана значениям и правилам. В связи с тем, что без некоторых понятий, несмотря на их сложность, нельзя объяснить суть определенных явлений, значения целесообразно характеризовать мерой "важности". Эти меры могут быть статическими и динамическими. Ясно, что возможность динамической модификации мер "сложности" и "важности" повышает гибкость рассматриваемого подхода. Хотя указанные меры некоторым образом взаимосвязаны, но из значения одной значение другой прямо не следует. Состав объяснений, выдаваемых пользователю, зависит и от меры сложности и от важности понятий.

№8. Механизм приобретения знаний

Процесс получения знаний от эксперта (или из каких-либо других источников знаний) и передачи их экспертной системе называют приобретением знаний. Обычно источником знания является эксперт-человек, но могут быть и эмпирические данные и тексты, в которых содержатся сведения об области экспертизы. Процесс приобретения знаний важен и сложен. Важность этого процесса обусловлена тем, что качество и эффективность решения задач экспертной системой определяются качеством и количеством используемых ею знаний. Сложность данного процесса обусловлена как тем, что объем знаний, используемых экспертом, велик, так и тем, что знания эти не полностью осознаются экспертом.

Целесообразно осуществить разбиение процесса приобретения знаний на фазы, отражающие изменение функций участников проектирования (эксперта и инженера по знаниям) и/или экспертной системы: 1) предварительная фаза; 2) начальная фаза; 3) фаза накопления.



Предварительная фаза приобретения знаний характеризуется тем, что экспертной системы еще не существует (отсюда и название фазы). Знания приобретаются инженером по знаниям от эксперта. На этой фазе задача инженера по знаниям состоит в том, чтобы получить от эксперта основные сведения об области экспертизы (основные понятия, отношения, подзадачи и т.п.) и сформировать на их основе общее представление о структуре данных и принципах построения экспертной системы. Эта фаза приобретения знаний выполняется на этапах идентификации, концептуализации и формализации.

На начальной фазе осуществляется наполнение системы знаниями о представлении, т.е. значениями, определяющими организацию, структуру и способ представления базы знаний. В связи с тем, что для определения указанных знаний необходимо владеть основами программирования и детально понимать функционирование проектируемой экспертной системы, введение знаний на начальной фазе может осуществлять только инженер по знаниям, а не эксперт. Начальная фаза осуществляется в ходе первой стадии этапа выполнения.

В ходе фазы накопления осуществляется приобретение основных знаний об области экспертизы. На современном уровне развития приобретение знаний на этой фазе осуществляется экспертом совместно с инженером по знаниям. На фазе накопления решаются следующие задачи: 1) обнаружение неправильности, неполноты или противоречивости знаний, используемых экспертной системой; 2) извлечение новых знаний, устраняющих обнаруженную неправильность, неполноту или противоречивость; 3) преобразование новых знаний в вид, понятный экспертной системе; 4) объединение "новых" знаний со "старыми". Следует отметить, что на данной фазе приобретаются все виды знаний, необходимые для эффективного и качественного функционирования ЭС.

Модели приобретения знаний

Процесс приобретения знаний является наиболее сложным этапом разработки экспертной системы. Это объясняется тем, что обычно инженер по знаниям плохо разбирается в предметной области, а эксперт не знает программирования. В связи с этим лексика, используемая экспертом, непонятна инженеру по знаниям. Чтобы уточнить и расширить лексику, требуется совместная работа эксперта и инженера по знаниям. Одна из наиболее сложных задач, стоящих перед инженером по знаниям, заключается в том, чтобы помочь эксперту структурировать знания о проблемной области.

Процесс приобретения знаний можно свести к последовательности выполнения следующих задач:

- 1) определяется необходимость модификации (расширения) знаний;
- 2) при необходимости модификации осуществляется извлечение новых знаний, в противном случае процесс приобретения знаний заканчивается;
- 3) новые знания преобразуются в форму, "понятную" экспертной системе;
- 4) знания системы модифицируются, и осуществляется переход к первой задаче.

В выполнении перечисленных задач могут принимать участие эксперт, инженер по знаниям (программист) и экспертная система. В зависимости от того, кто выполняет задачу, можно выделить различные модели приобретения знаний.



В ранних работах по искусственному интеллекту взаимодействие с разрабатываемой системой осуществлял только программист. При

разработке системы программисты не отделяли знания (данные) от механизма вывода. В задачу программиста входило освоить с помощью эксперта предметную область и затем при разработке системы выступать в роли и эксперта, и программиста. Недостаточное знание области экспертизы не позволяло программисту гарантировать полноту и непротиворечивость приобретенных знаний. Кроме того, неизбежные модификации системы приводили (при отсутствии разделения системы на базу знаний и механизм вывода) к невозможности сохранить однажды достигнутой непротиворечивость знаний. В этой модели все перечисленные выше задачи по приобретению знаний выполнял программист.

Последующие разработки систем искусственного интеллекта основывались на отделении знаний от программ и оформлении знаний в виде простых информационных структур, называемых базами знаний. В этом случае эксперт взаимодействует с системой через



инженера по знаниям. Преимущество данного подхода состоит в том, что база знаний упрощает модификацию знаний. В данной модели первую и вторую задачи приобретения знаний выполняет эксперт с помощью инженера по знаниям, третью задачу выполняет инженер по знаниям, а четвертую - экспертная система. Важным недостатком этого подхода является его большая трудоемкость. Действительно, из четырех задач по приобретению знаний автоматизирована только одна.

Эксперт, минимально сведующий в вопросах программирования, может взаимодействовать с экспертной системой через интеллектуальный редактор без посредничества инженера по знаниям. В этой модели интеллектуальный редактор должен обладать развитыми диалоговыми способностями и значительными знаниями о структуре базы знаний (т.е. метазнаниями). Заметим, что интеллектуальный редактор может быть включен в состав экспертной системы. При использовании интеллектуального редактора эксперт (с минимальной помощью инженера по знаниям и экспертной системы) решает первую и вторую задачи приобретения знаний, третья и четвертая задачи выполняются экспертной



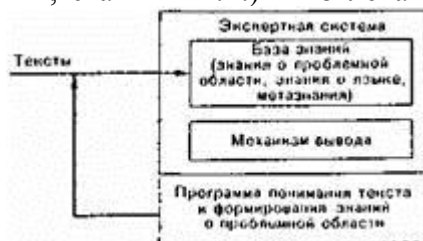
системой.



ЭС могут приобретать знания аналогично тому, как это делает эксперт-человек. В этом случае индуктивная программа будет анализировать данные, содержащие сведения о некоторой области экспертизы, автоматически формируя значимые отношения и правила, описывающие предметную область. Предполагается, что в базе знаний в явном виде хранятся конкретные факты о проблемной области, а задача индуктивной

программы — сделать значимые обобщения. Основным достоинством этого подхода является автоматизация всех перечисленных выше четырех задач по приобретению знаний.

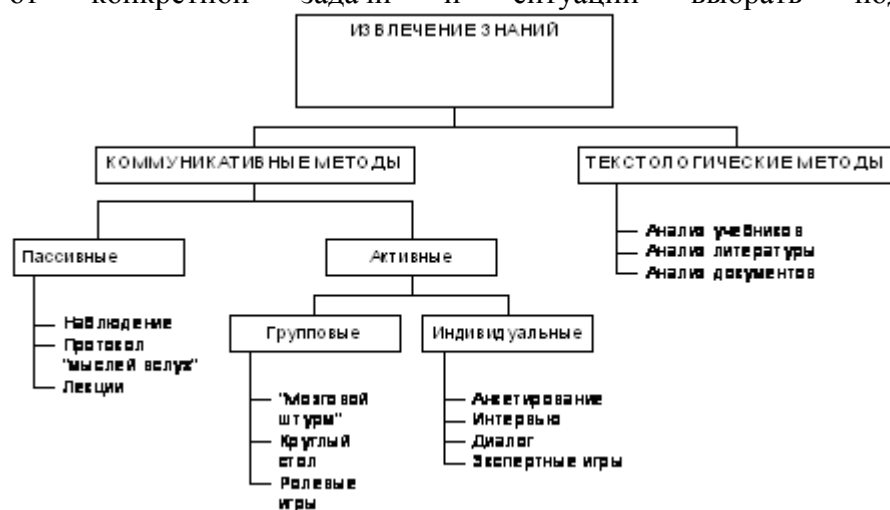
Дальнейшие перспективы развития экспертных систем связываются с приобретением знаний непосредственно из текстов на естественном языке. В данном случае требуется читать обычные печатные тексты (книги, статьи и т.п.) и извлекать из них знания, т.е. понимать



текст, схемы, графики и т.п. Сложность задачи понимания состоит не только в обработке естественного языка, но и в необходимости воссоздать по тексту модель некоторой проблемной области. Эти требования пока превосходят возможности существующих программ понимания, несмотря на то, что в данном случае речь идет об анализе текстов, ограниченных достаточно узкой областью экспертизы. Следует отметить, что приведенные выше методы (модели) приобретения знаний различаются с точки зрения их независимости от эксперта. Методы приведены в порядке возрастания этой независимости, т.е. в порядке увеличивающейся степени автоматизации процесса приобретения знаний.

Методы извлечения знаний в экспертных системах

Рассмотрим классификацию методов извлечения знаний, которые позволяют в зависимости от конкретной задачи и ситуации выбрать подходящий метод (рис.).



Основной принцип деления связан с источником знаний. Коммуникативные методы охватывают все виды контактов с источником знаний – экспертом, а текстологические касаются методов извлечения знаний из документов и специальной литературы (статей, монографий, учебников).

Коммуникативные методы можно также разделить на две группы: активные и пассивные. Пассивные методы подразумевают, что ведущая роль в процедуре извлечения как бы передается эксперту, а инженер познаниям только протоколирует рассуждения эксперта во время его реальной работы по принятию решений или записывает то, что эксперт считает нужным рассказать. В активных методах инициатива полностью в руках инженера по знаниям. Он активно контактирует с экспертом различными способами – в играх, диалогах, беседах за круглым столом и т.п.

Пассивные методы на первый взгляд достаточно просты, но на самом деле требуют от инженера по знаниям умения четко анализировать «поток сознания» эксперта и выявлять в нем значимые фрагменты знаний. Отсутствие обратной связи значительно ослабляет эффективность этих методов. Поэтому они обычно играют вспомогательную роль при активных методах.

Активные методы можно разделить на две группы в зависимости от числа экспертов, отдающих свои знания. Если их число больше одного, то целесообразно применять методы групповых обсуждений предметной области. Такие групповые методы активизируют мышление участников дискуссий и позволяют выявить тонкие аспекты их знаний. Однако индивидуальные методы на сегодняшний день остаются ведущими, поскольку «отъем знаний» не терпит лишних свидетелей.

Игровые методы сейчас широко используются в социологии, экономике, менеджменте для подготовки руководителей, учителей, врачей и др. В игре человек раскрепощается и чувствует себя намного свободнее, чем в обычной трудовой деятельности.

На выбор метода влияют три фактора: личностные особенности инженера по знаниям, эксперта и характеристика предметной области.

Одна из возможных классификаций людей по психологическим характеристикам делит всех людей на три типа:

- * мыслитель (познавательный тип);
- * собеседник (эмоционально-коммуникативный тип);
- * практик (практический тип).

Мыслители ориентированы на интеллектуальную работу и обладают такими характеристиками, как полнезависимость и рефлексивность. Собеседники – это общительные, открытые люди, готовые к сотрудничеству. Практики предпочитают действие разговорам, хорошо реализуют замыслы других, направлены на результативность работы.

Для характеристики предметных областей можно использовать следующую классификацию:

- * хорошо документированные;
- * средне документированные;
- * слабо документированные.

Эта классификация связана с соотношением двух видов знаний. Это экспертное «личное» знание и материализованное в книгах «общее» знание в данной предметной области.

Кроме этого предметные области можно разделить по критерию структурированности знаний. Под структурированностью понимают степень теоретического осмысления и выявленности основных закономерностей и принципов, действующих в данной предметной области. По степени структурированности предметные области могут быть хорошо структурированными - с четкой аксиоматизацией, широким применением математического аппарата, устоявшейся терминологией; средне структурированными – с определившейся терминологией, развивающейся теорией, явными взаимосвязями между явлениями; слабо

структурированными – с размытыми определениями, богатой эмпирикой, скрытыми взаимосвязями, с большим количеством белых пятен.

Пассивные методы

Несмотря на название эти методы требуют от инженера по знаниям не меньшей отдачи, чем активные методы игры и диалог.

Наблюдения. В процессе наблюдений инженер по знаниям находится непосредственно рядом с экспертом во время его профессиональной деятельности или имитации этой деятельности. Эксперту объясняется цель наблюдений и рекомендуется максимально комментировать свои действия.

Во время сеанса аналитик записывает все действия эксперта, его реплики и объяснения. Непременное условие этого метод – невмешательство аналитика в работу эксперта. С этой точки зрения метод наблюдений является «чистым» методом, исключая навязывание эксперту чужих представлений.

Существуют две основные разновидности проведения наблюдений: наблюдение за реальным процессом и за имитацией процесса. На практике используются обе разновидности. Наблюдение за реальным процессом позволяет глубже понять предметную область, отметить особенности процесса принятия решения. Это необходимо для проектирования эффективного интерфейса пользователя.

Наблюдение за имитацией процесса проводят при невозможности наблюдения за реальным процессом. Наблюдение за имитацией процесса обеспечивает эксперту меньшее напряжение. Однако ненастоящая работа может привести и к решению, отличному от настоящего.

Анализ протоколов мыслей вслух. Протоколирование мыслей вслух отличается от наблюдений тем, что эксперта просят не просто прокомментировать свои действия и решения, но и объяснить, как это решение было найдено, т.е. продемонстрировать всю цепочку своих рассуждений. Во время рассуждений эксперта все его слова протоколируются инженером по знаниям. Иногда этот метод называют «вербальные отчеты». Основной трудностью метода является принципиальная сложность для любого человека объяснить, как он думает. Существует экспериментальное психологическое доказательство факта, что люди не всегда в состоянии достоверно описать мыслительные процессы. Кроме того, часть знаний (например, процедурные типа как завязывать шнурки) вообще слабо коррелируют с их словесным описанием.

Удачно проведенное протоколирование «мыслей вслух» является одним из наиболее эффективных методов извлечения. В нем эксперт может проявить себя наиболее ярко, он ничем не скован и никто ему не мешает. Для большего числа экспертов это самый приятный способ извлечения знаний.

Для реализации обратной связи между интерпретацией инженера и представлениями эксперта метод потом дополняется одним из активных методов.

Лекции. Лекция является самым старым способом передачи знаний. Этим способом можно воспользоваться, если эксперт имеет опыт преподавателя или руководителя производства.

В лекции эксперту также предоставляется много степеней свободы для самовыражения. При этом необходимо сформулировать эксперту тему и задачу лекции. От инженера по знаниям в

этой ситуации требуется лишь грамотно законспектировать лекцию и в конце задать необходимые вопросы.

Искусство ведения конспекта заключается в «помехоустойчивости». Записывается главное, опускается второстепенное, выделяются фрагменты знаний (параграфы, подпараграфы), записываются только осмысленные предложения и обобщения.

Метод извлечения знаний в форме лекции, как и все пассивные методы, используют в начале разработки как эффективный способ быстрого погружения инженера по знаниям в предметную область.

Недостатки метода – слабая обратная связь, недостаток хороших лекторов среди экспертов.

Активные индивидуальные и групповые методы

Активные индивидуальные методы извлечения знаний на сегодняшний день – наиболее распространенные. Основными активными методами являются анкетирование, интервью, свободный диалог, игры с экспертом. Во всех методах инженер по знаниям выполняет активную функцию.

Анкетирование. Это наиболее стандартизованный метод. Когнитолог заранее составляет вопросник или анкету, размножает ее и использует для опроса нескольких экспертов.

Основными факторами, на которые можно существенно повлиять при анкетировании, являются средства общения (вопросник) и ситуация общения. К вопроснику предъявляются особые требования, такие например, как отсутствие монотонности, приспособленность к языку экспертов, специальная упорядоченность вопросов и оптимальная их избыточность, ясный и понятный язык и предельная вежливость.

Интервью. Это специфическая форма общения, при которой когнитолог задает эксперту серию заранее подготовленных вопросов с целью извлечения знаний о предметной области.

Основное отличие интервью от анкетирования в том, что оно позволяет аналитику опускать ряд вопросов в зависимости от ситуации, вставлять новые вопросы в анкету, изменять темп, разнообразить ситуацию общения.

Центральным звеном активных методов являются вопросы, поэтому к ним предъявляются особые требования. Существует даже специальная ветвь математической логики – эротетическая логика (логика вопросов). Однако ее результаты «страшно далеки от народа».

Все вопросы можно разбить на ряд типов.

Открытый вопрос называет тему или предмет, оставляя полную свободу эксперту по форме и содержанию ответа.

В закрытом вопросе эксперт выбирает ответ из набора предложенных. Эти вопросы легче обрабатывать. Но они закрывают ход рассуждений эксперта и программируют его ответ в определенном направлении.

Личный вопрос касается непосредственно личного опыта эксперта.

Безличный вопрос направлен на выявление общепринятых закономерностей предметной области.

Вербальные вопросы - это традиционные устные вопросы.

Зондирующие вопросы направляют рассуждения эксперта в нужную сторону, когда не срабатывают основные вопросы, направленные на выявление знаний.

Контрольные вопросы применяются для проверки достоверности и объективности информации, полученной в интервью ранее.

На качество интервью в основном, влияют три характеристики вопросов: язык вопроса (понятность, лаконичность, терминология), порядок вопросов (логическая последовательность и немонотонность); уместность вопросов (этика, вежливость).

Вопросы имеют для эксперта диагностическое значение – несколько глупых вопросов могут полностью разочаровать эксперта и отбить у него охоту к дальнейшему сотрудничеству.

Свободный диалог. Это метод извлечения знаний в форме беседы когнитолога и эксперта, в которой нет жесткого регламентированного плана и вопросника.

Этот метод требует от когнитолога высочайшей профессиональной и психологической подготовки, которая также включает составление плана проведения сеанса извлечения.

Активные групповые методы. Сюда относятся ролевые игры, дискуссии за круглым столом с участием нескольких экспертов и «мозговые штурмы». Основное достоинство групповых методов – это возможность одновременного «поглощения» знаний от нескольких экспертов.

Метод «круглого стола» предусматривает обсуждение какой-либо проблемы из выбранной предметной области, в котором принимают участие с равными правами несколько экспертов. Обычно участники вначале высказываются в определенном порядке, а затем переходят к живой свободной дискуссии. Однако в этом методе эксперты часто говорят не то, что думают. Сказывается желание произвести впечатление на других экспертов и спрашивают они не то что им интересно, а то, что демонстрирует их собственную компетентность. Научная плодотворность дискуссии делает этот метод привлекательным и для самих экспертов.

«Мозговой штурм» – один из наиболее распространенных методов раскрепощения и активизации творческого мышления. Замечено, что боязнь критики мешает творческому мышлению, поэтому основная идея штурма – это отделение процедуры генерации идей в замкнутой группе специалистов от процесса анализа и оценки высказанных идей. Участникам штурма предлагается высказывать любые идеи на заданную тему. Критика запрещена. Обычно 10-15% идей оказывается разумными, а среди них бывают весьма оригинальные. Результаты оцениваются обычно группой экспертов, не участвовавших в генерации.

Экспертные игры. Игрой называют такой вид человеческой деятельности, который отражает другие ее виды. При этом для игры характерны одновременно условность и серьезность.

Экспертная игра возникла на трех источниках: понятии деловой игры, используемой при подготовке специалистов и моделировании, понятии диагностической игры и компьютерных играх, применяемых в обучении.

Под деловой игрой чаще всего понимают эксперимент, где участникам предлагается производственная ситуация, а они на основе жизненного опыта, своих общих и специальных знаний и представлений принимают решения. Решения анализируются и вскрываются закономерности мышления участников эксперимента. Именно эта анализирующая часть полезна для извлечения знаний.

Диагностическая игра – это та же деловая игра, но применяемая конкретно для диагностики методов принятия решения (например, в медицине).

Компьютерные игры вошли в практику исследований с появлением ЭВМ. Экспертные компьютерные игры сочетают в себе свойства позиционных, динамических, зрелищных и обучающих игр. Компьютерные игры вызывают высокий интерес у эксперта, привлекают дизайном и динамикой. Однако сложность и высокая цена создания специализированных игр в конкретной предметной области ограничивают их применение.

Текстологические методы

Группа текстологических методов объединяет методы извлечения знаний, основанные на изучении специальных текстов из учебников, монографий, статей, методик и других носителей профессиональных знаний.

Среди методов извлечения знаний эта группа является наименее разработанной. Задачу извлечения знаний из текстов можно сформулировать как задачу понимания и выделения смысла текста. Сам текст на естественном языке является лишь проводником смысла, а замысел и знания автора лежат во вторичной структуре (смысловой структуре текста), настраиваемой над естественным текстом. При этом можно выделить две смысловые структуры – смысл, который пытался заложить автор (это его модель мира), и смысл, который постигает читатель, т.е. инженер по знаниям в процессе интерпретации. Сложность процесса заключается в принципиальной невозможности совпадения знаний за счет разной совокупности представлений автора и читателя. Цель извлечения знаний – достичь максимального понимания.

№9. Диалоговый компонент

Взаимодействие с ЭС осуществляют различные типы пользователей: специалисты, неспециалисты, учащиеся. Каждый из них имеет определенные требования к диалогу, но всех их объединяет следующее: языком общения должен быть ограниченный ЕЯ, или близкий к нему; процесс взаимодействия пользователей с ЭС не сводится к изолированным парам «запрос-ответ», а представляет собой разветвленный диалог, в котором инициатива переходит от одного участника к другому.

Компьютерно-лингвистический подход к диалогу

В современной научной литературе наметились два подхода к анализу и построению вопросов и ответов как основных элементов в структуре диалоговой формы передачи информации: лингвистический и компьютерный. Но многообразие видов и областей использования диалога настоятельно требуют разработок синтетического характера, в которых достижения лингвистического и компьютерного анализа вопросно-ответных структур были бы представлены в единстве. Благоприятной почвой, на которой может быть возвращено синтетическое направление в исследовании диалога, оказывается логика. При синтетическом подходе к диалогу на базе логики удастся, с одной стороны, формализовать естественно-языковые средства общения и повысить тем самым степень

детерминированности основных структурных элементов диалога, а, с другой стороны, максимально приблизить форму компьютерного представления диалога к естественно-речевому общению, увеличив тем самым эффективность восприятия информации в человеко-машинных системах. К особо важным логическим закономерностям диалога как вида человеческих рассуждений или способа информационного обмена относятся те структурно-информационные характеристики, которые инвариантны к содержанию конкретного диалога.

Лингвистический подход к диалогу

Понятие "естественный язык" знакомо любому человеку. Но в лингвистике нет бесспорного определения этому понятию. Естественный язык - это некоторая система элементов плюс система правил их функционирования.

В русском языке для письменного текста выделяется следующая система элементов: морфемы, слова, словосочетания, предложения, связанный текст (дискурс).

Рассмотренные единицы языка образуют некоторую систему разных уровней. Каждому уровню в языке соответствует группа правил. Исторически выделяются следующие основные группы правил: морфологические, синтаксические, семантические.

Морфологические правила определяют законы сочетания морфем в слове; синтаксические - законы сочетания слов или словосочетаний во фразах и предложениях; семантические - законы преобразования предложения в тексте. Выделение правил и их формализация необходимы для построения моделей ЕЯ.

Естественному языку присущи омонимия и синонимия. Омонимия позволяет использовать ограниченное число знаков для обозначения разных реальных объектов и понятий. Синонимия - характеризовать некоторую сущность различными знаками, подчеркивая определенные ее стороны.

В языке, а особенно в речи, допускаются и часто используются неполные конструкции. Одна из видов таких конструкций - эллипсис, представляющий собой предложение, в котором опущено слово или несколько слов, легко восстанавливаемых по локальному или глобальному дискурсу.

Средства отождествления и различения объектов, упоминаемых в тексте, реализуется с помощью анафор. Анафора обозначает (чаще всего) двукратное повторение в разных частях предложения или в нескольких связанных фразах словесного отображения одного и того же объекта. В частном случае анафора - это повторение одного и того же слова. Одно из таких повторений называется антецедентом, а другое - ссылкой. Чаще всего используются местоименные ссылки.

Известно, что язык задает норму. Однако она не является обязательной для речи. Типичны, например, орфографические ошибки, и др. При машинном анализе затруднения в понимании могут вызвать и такие предложения, которые, казалось бы, построены правильно. Естественный язык, кроме того, изобилует идиоматическими и фразеологическими оборотами, метафорами и ассоциациями, нечеткостями и неопределенностями и т. п. Все это затрудняет моделирование ЕЯ.

Однако выход из этой ситуации существует и состоит в том, что при построении систем общения используются подмножества ЕЯ, которые в конкретных приложениях проблемно

ориентированны и, следовательно, более просты, чем ЕЯ в полном объеме. Такие подмножества называют языками деловой прозы. Их лексика ограничена предметной областью, смыслы слов более однозначны, отсутствуют метафоры, ассоциации и т. д. Но тем не менее в них используются те же конструкции предложения, эллипсисы и анафоры и т. д. Поэтому проблема анализа текстов даже для этих языков нетривиальна.

Информационный подход к диалогу

В структуре диалога можно выделить вопрос и ответ. Вопрос - это форма мышления, в которой при наличии определенной информации, выраженной в предпосылках вопроса, требуется дать дополнительную информацию, выражаемую в ответе. В обобщенном виде под вопросом понимается некоторое неполное знание, дополнение к которому (до полного) содержится в предполагаемом истинном ответе, то есть вопрос - это часть информации ответа. Иначе говоря, вопрос - это неполная информация, в которой формулируется требование о ее дополнении до полного и однозначно правильного ответа. Вопросы делятся на корректные (заложены истинные предпосылки или на которые можно дать истинный или ложный ответ) и некорректные. Поскольку вопрос может быть задан относительно любого элемента предложения, то известную часть будем называть темой, а неизвестную, которая должна прозвучать в ответе, ремой.

Элементарный акт (шаг) диалога - это пара, состоящая из вопроса и соответствующего ему ответа. В общем случае процесс общения не может быть сведен к обмену изолированными парами высказываний вопрос-ответ. Высказывания участников общения образуют связный текст - дискурс (цепочку элементарных его актов), имеющий, как правило, достаточно сложную структуру.

Проблема формализации естественного языка

ЕЯ представляет собой сложно организованную семиотическую (знаковую) систему. В этой системе много уровней и сложных связей.

Под знаком понимаются элементы, обладающие одновременно тремя свойствами: синтаксисом, семантикой и прагматикой, отношения между которыми неоднозначны.

Для многих ЕЯ удалось описать формальную систему, лежащую в основе механизма порождения синтаксически правильных фраз. В качестве базовых элементов при этом не обязательно задавать все мыслимые в грамматике словоформы. Введя морфологический уровень, можно хранить в качестве базовых элементов лишь основы слов, а флексии подсоединить к ним при необходимости автоматически.

Семантический уровень ЕЯ формализуется значительно сложнее. Связано это с тем, что семантически правильная фраза отличается от неправильной фразы тем, что ей можно приписать некоторый смысл, другими словами, интерпретировать в некоторой модели знаний. Для каждого интерпретируемого элемента (словоформы или основы слова) необходимо задать множество значений. Для ЕЯ характерно, что все его элементы многозначны. Поэтому правила интерпретации устроены так, что значения, приписанные словам фразы, определяются лишь на основе анализа всей фразы. Существует множество подходов к построению правил для интерпретации семантики ЕЯ. Однако в настоящее время ни один из ЕЯ не удалось с необходимой полнотой описать на семантическом уровне из-за огромной семантической омонимии.

Еще более сложна формализация прагматического уровня ЕЯ. Однако при построении систем для конкретных областей формализация этого уровня сильно упрощается, т.е. прагматика текстов зависит от сферы их применения.

Обобщенная схема ЕЯ-системы

ЕЯ-система должна выполнять следующие функции:

@ ведение диалога - определение его структуры и той роли, которую система и пользователь выполняют на текущем шаге диалога;

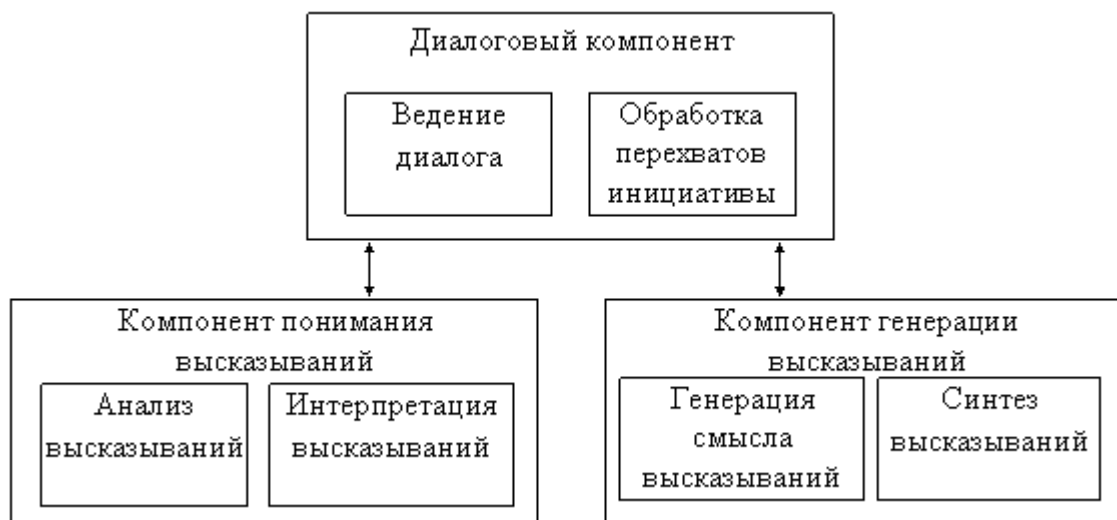
@ понимание - преобразование поступающих от пользователя высказываний на ЕЯ в высказывания на языке внутреннего представления;

@ обработка высказываний - формирование или определение задания на решение задач или подзадач на каждом шаге диалога;

@ генерация - формирование выходных высказываний на ЕЯ.

При реализации конкретных ЕЯ-систем суть этих функций может в значительной мере варьироваться.

В соответствии с этими функциями обобщенная схема ЕЯ-системы может быть представлена в виде 3-х компонентов: диалоговый; компонент понимания речи; компонент генерации высказываний. рис..



Ведение диалога выполняется по одной из схем: диалог ведет пользователь (инициатива в основном принадлежит пользователю, а система только реагирует на его требования, определяя по виду требования тип задания. Для системы весь диалог сводится к выработке реакций на текущие высказывания пользователя); диалог ведет система (система ведет диалог в соответствии с имеющимися у нее представлениями о структуре диалога и о способе обмена высказываниями).

Выделяют три режима ведения диалога:

- 1) открытый;
- 2) ограниченный;
- 3) закрытый.

Под открытым режимом диалога понимается такой диалог, в котором на ответ не накладывается никаких ограничений, кроме той информации, которая содержится в вопросе и обуславливает соответствующее содержание ответа.

Под ограниченным режимом диалога понимается такой диалог, при котором множество ответов четко ограничивается формулировкой вопроса. Информация, которая должна прозвучать в ответе, является подмножеством множества возможных ответов.

Под закрытым режимом диалога будем понимать диалог, при котором вопрос представляет из себя логическое выражение, значение которого (истина или ложь) должно прозвучать в ответе (дихотомические вопросы).

Каждый элементарный акт диалога не может быть одновременно реализован в различных режимах, однако их разумное чередование представляется вполне естественным, более того, оно позволяет разнообразить, оживить процесс диалога.

Шаг диалога (см. табл.) характеризуется следующими параметрами: инициатор и тип инициирования (вид действия); способ влияния действия на реакцию. Инициатором в общем случае может быть как пользователь, так и система. Действия пользователя подразделяются на запросы (вопросительные предложения на ЕЯ с необходимыми параметрами) и команды (служат для перехвата инициативы и выполнения технологических действий). Если инициатива принадлежит системе, то вид действия определяется исходя из того, насколько диалоговому компоненту известна функция (и/или параметры), с помощью которых осуществляется решение задачи на данном шаге. Для ввода некоторых параметров можно использовать простые вопросы. Альтернативные вопросы и вопросы с фиксированной структурой применимы, когда возможно выполнение одной из возможных функций, а выбор осуществляется в зависимости от реакции пользователя. Альтернативные вопросы ограничивают выбор явно, т. е. пользователю предоставляется множество ответов.

Основные типы шагов диалога Таблица

Инициатор шага	Вид действия	Вид реакции	Способ спецификации	
			Ограниченность пространства выбора	Однозначность определения
Пользователь	ЕЯ-запрос	Ответ на запрос	неограниченное	неоднозначное
Пользователь	Команда	Выполнение команды	ограниченное	однозначное
Система	Простой вопрос	Ответ на простой вопрос	фиксированное	однозначное
Система	Альтернативный вопрос	Выбор альтернативы	ограниченное	однозначное

Из рассмотренных основных типов шагов диалога следует, что при действиях, не ограничивающих возможные реакции, не гарантируется однозначная спецификация задачи.

Компонент понимания высказываний предназначен для выделения смысла входного высказывания и выражение этого смысла на внутреннем языке системы. Под смыслом понимается вся та семантико-прагматическая информация, которую пользователь хотел передать системе. Выявление смысла высказывания в общем случае требует его рассмотрения в контексте всего диалога.

На этапе анализа выделяются описания сущностей, упомянутых во входном высказывании, выявляются свойства этих сущностей и отношения между ними.

В методах анализа выделяются анализ слов, предложений, дискурса. Анализ слов сводится к морфологическому анализу. Анализ предложения сводится к синтаксическому и семантическому анализу. Анализ дискурса сводится к определению связей между предложениями.

Типы анализаторов (имеют семантическую и синтаксическую ориентацию):

Традиционные анализаторы. Наиболее распространенным способом анализа ЕЯ-предложения является разбор сверху вниз, слева направо, основанный на некоторой фиксированной грамматике. Анализаторы такого типа терпят неудачу при малейших отклонениях от грамматических норм.

Концептуальные анализаторы. Используют методы разбора, направляемые значениями базовых событий, обнаруженных в предложениях.

Анализаторы, использующие сопоставление по образцам. Анализ сводится к сопоставлению предложения с некоторым множеством образцов, представляющие собой последовательности из одного или нескольких слов. Гибкость таких анализаторов определяется гибкостью сопоставления. При таком подходе затруднительна обработка сложных предложений.

Анализаторы, использующие разнообразные методы. Использование в одном анализаторе нескольких методов позволяет обеспечить гибкость анализа, необходимую для обработки неграмматических конструкций.

Интерпретация заключается в отображении входного высказывания на знания системы. Основными задачами данного этапа являются: буквальная интерпретация высказывания в контексте диалога; интерпретация высказывания на намерения говорящего. Буквальная интерпретация состоит в том, чтобы, учитывая контекст диалога, идентифицировать образы тех сущностей области интерпретации, которые имел в виду пользователь. В качестве области интерпретации могут использоваться:

@ проблемная область;

@ область системы (если пользователь интересуется возможностями или состоянием системы (системный словарь));

@ область пользователя (если информация касается знаний или намерений пользователя);

@ область дискурса (если в высказываниях содержатся ссылки на предыдущие высказывания).

В общем случае процесс идентификации некоторой сущности может иметь 3 исхода:

@ однозначный - данному описанию сопоставляется единственная сущность;

@ многозначный - более одной сущности;

@ неудовлетворительный - не сопоставляется ни одна сущность.

В последних двух случаях системе необходимо осуществить перехват инициативы произвести уточняющий поддиалог. В случае многозначной интерпретации системе необходимо сгенерировать альтернативный вопрос. При неудовлетворительной - указать причины.

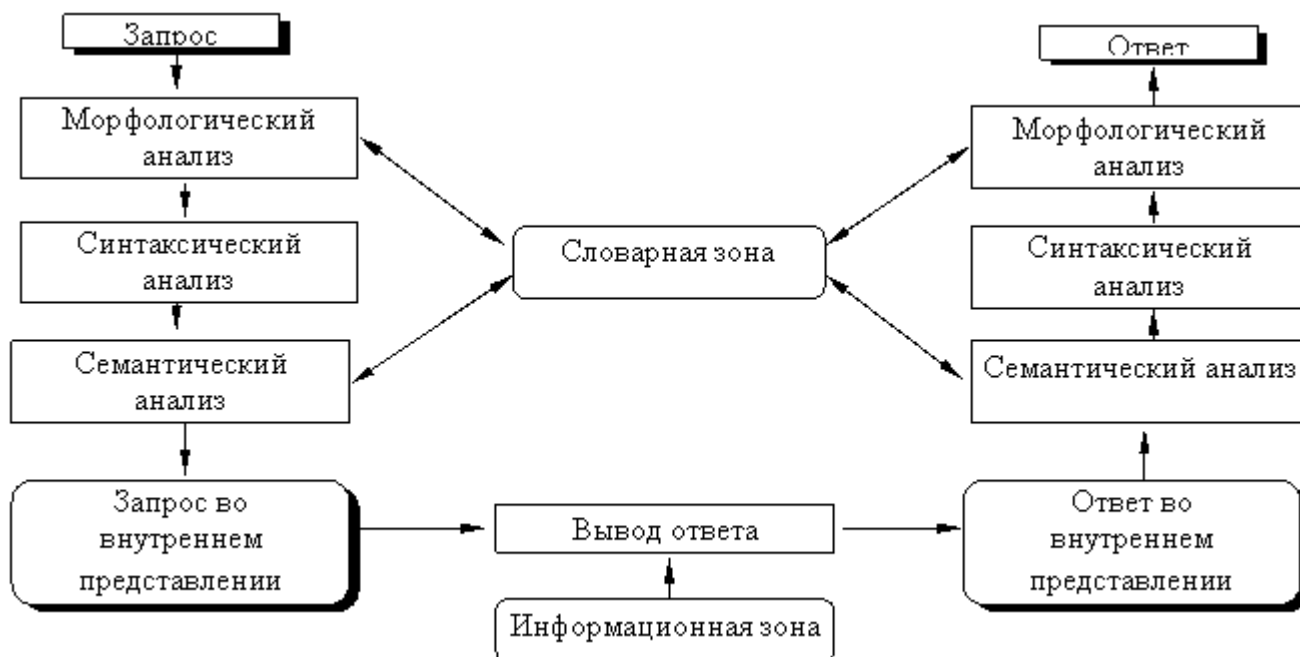
Вторая задача интерпретации состоит в том, чтобы, применяя имеющиеся у системы методы вывода, определить, как обрабатываемое высказывание соотносится с целями и планами участников общения. Задача решается совместно диалоговым компонентом и компонентом понимания высказываний. Сложность решения этой задачи заключается в том, что одно и тоже высказывание может использоваться для достижения целей, относящихся к различным областям.

Компонент генерации высказываний решает в соответствии с результатами, полученными остальными компонентами системы, две основные задачи: генерация смысла, т. е. определение типа и смысла выходного высказывания системы во внутреннем представлении; синтез высказываний, т.е. преобразование смысла в высказывание на ЕЯ. В общем случае при решении задачи формирования смысла выходного высказывания необходимо учитывать прагматический аспект, т. е. цели участников общения. Вторая задача состоит в синтезе ЕЯ-выражения, соответствующего внутреннему представлению. Естественность (степень близости к ЕЯ) и выразительная мощьность (возможность выразить разнообразные отношения, понятия и т.д.) выходных высказываний может быть различна. Высказывания могут фактически не синтезироваться, а выбираться из заранее заготовленного списка, либо иметь шаблон ответа, в который подставляются искомые слова. В ЕЯ-системах, построенных на основе СУБД результатом решения задачи может являться множество данных, формат представления которых определяется средствами генерации отчетов СУБД.

Для понимания принципов построения ЕЯ-систем важен также вопрос об используемых в системе знаниях, поскольку именно знания, представленные в различных формах, являются той базой, на которой осуществляется решение описанных выше задач.

Лингвистическая трансляция

Систему общения на ЕЯ можно рассматривать как систему перевода предложений ЕЯ в его внутримашинное представление и обратно в ЕЯ (см. рис.). Эту схему часто называют лингвистическим процессором (ЛП)[8].



Лингвистическая трансляция включает:

анализ текста на ЕЯ, который включает в себя:

- а) морфологический анализ слов, под которым понимается обработка слов и словосочетаний вне связи с контекстом, результатом которой является выделенные основы (корней) слов и приписанная им морфологическая информация (часть речи, род, число, и т.п.) Текст разбивается на словосочетания - отрезки текста между пробелами (словоформы). Есть два способа морфологического анализа: декларативный и процедурный. При декларативном способе словарь содержит все возможные словоформы каждой основы с приписываемой морфологической информацией. Анализ сводится к поиску словоформы. Требуется большой объем памяти ЭВМ. При процедурном способе словарная зона системы состоит из словаря основ и словаря аффиксов. С помощью соответствующих алгоритмов каждой словоформе приписывается морфологическая информация.
 - б) синтаксический анализ предложений, задачей которого является построение синтаксической структуры на основе морфологической информации и синтаксических правил объединения слов и словосочетаний. Она отображает связи, существующие между словами предложения.
 - в) семантическую интерпретацию, целью которой является определение смыслового содержания запроса. На выходе формируется внутреннее представление входного сообщения, отражающее знание системы о предметной области. Семантика представляется как интерпретация входного запроса моделью предметной области.
- 2) синтез ответов. Целью этого синтеза является выражение на ЕЯ сведений, нужных пользователю. Задача синтеза может быть представлена в виде двух подзадач - внелингвистический (семантический) и лингвистический (синтаксический и морфологический) синтез.

В большинстве случаев вместо полного синтеза используется синтез по шаблонам. Суть его состоит в том, чтобы для конкретной системы рассмотреть все типы сообщений, относящихся как к процессу общения, так и к процессу выдачи результатов работы и для каждого типа разработать шаблон, который заполняется при выдаче ответа. Этот метод и будет использоваться в системе, так как он наиболее простой и быстродействующий.

