

1. КРАТКИЕ СВЕДЕНИЯ ОБ ЭКСПЕРТНЫХ СИСТЕМАХ

Искусственный интеллект ориентирован на создание методов дублирования функций живых интеллектуальных систем искусственными системами. Технология разработки интеллектуальных программных средств основана на том, что знания о решении задач отделяются от программ и реализуются в виде базы знаний, а в программах реализуется алгоритм манипулирования этими знаниями. Этот алгоритм называют механизмом логического вывода. Знания являются явными и доступными, что отличает интеллектуальные системы от большинства традиционных программных средств.

К знаниям относят информацию о логике решения задач, а к данным – информацию, которая должна быть проанализирована в соответствии с этой логикой. Поэтому имеются специфические признаки, отличающие знания от данных:

1. внутренняя интерпретируемость, означающая, что в знаниях находится информация, раскрывающая смысл элементов знаний;
2. структурированность знаний, заключающаяся в возможности декомпозиции сложных объектов на более простые и установление соответствующих связей между ними;
3. связность знаний отражает причинно-следственные и временные отношения между фактами, процессами и явлениями;
4. активность знаний, так как они могут содержать планы действий и управляющие процедуры.

Формализованные знания формулируются в виде общих и строгих суждений (законов и формул), отражающих универсальные знания.

Неформализованные знания отличаются конкретностью, субъективностью и приблизительностью и обычно представляют многообразие эмпирических приемов и правил. В зависимости от того, какие знания преобладают в той или иной предметной области, ее относят к формализованным или неформализованным областям.

Основными особенностями интеллектуальных систем являются:

1. алгоритм решения неизвестен заранее, а строится самой системой с помощью рассуждений, базирующихся на эвристических приемах;
2. способность анализа и объяснений своих действий и знаний;
3. способность приобретения новых знаний от пользователя – эксперта, не знающего программирования, и изменения в соответствии с ними своего поведения.

С точки зрения пользователя наиболее важными свойствами интеллектуальных систем являются:

1. возможность получения подробных объяснений полученных решений;

2. возможность постепенного наращивания баз знаний без перепрограммирования;
3. знания из баз знаний обладают самостоятельной ценностью и могут распространяться.

Основные виды интеллектуальных систем, отличающиеся целью и реализацией знаний:

1. интеллектуальные информационные поисковые системы, обеспечивающие в процессе диалога взаимодействие конечных пользователей с базами знаний на профессиональных языках пользователей;
2. интеллектуальные пакеты прикладных программ, в которых, в отличие от обычных пакетов, реализуются знания о процессе решения задач предметной области;
3. экспертные системы, обеспечивающие возможность решения задач в полностью неформализованных областях, причем вся информация о решении задач предметной области реализуется в виде баз знаний.

Наиболее распространенными моделями представления знаний в экспертных системах являются:

1. логическая модель;
2. продукционная модель;
3. фреймовая модель;
4. модель представления знаний в виде семантической сети;
5. модель доски объявлений;
6. модель представления знаний в виде сценария.

Экспертные системы (ЭС) представляют собой класс компьютерных программ, которые выдают советы, проводят анализ, выполняют классификацию и ставят диагноз. Они ориентированы на решение задач, обычно требующих проведения экспертизы человеком-специалистом. В отличии от машинных программ, использующих процедурный анализ, ЭС решают задачи в узкой предметной области на основе дедуктивных суждений.

Классификация ЭС по решаемой задаче:

- 1) *Интерпретация*. Это анализ исходных данных с целью определения их смысла.
- 2) *Диагностика*. Это процесс поиска неисправности в системе, который основан на интерпретации данных
- 3) *Мониторинг (наблюдение)*. Это задача непрерывной интерпретации сигналов и выдачи оповещений в тех случаях, когда контролируемые параметры выходят за допустимые пределы. Мониторинг является частью диагностической системы, которая работает в реальном масштабе времени.

- 4) *Прогнозирование.* Это предсказание хода развития системы в будущем на основании ее поведения в прошлом и настоящем. Эти системы содержат блоки обработки статистики, блоки принятия решения на основе неполной информации, блоки генерации альтернативных путей развития системы.
- 5) *Планирование.* Под планированием понимается определение планов действий, относящихся к объектам, способным выполнять некоторые функции.
- 6) *Проектирование.* Цель: помочь человеку при нахождении им эвристических решений в процессе творчества или автоматизировать рутинную работу. Проектирование состоит в подготовке спецификаций на создание “объектов” с заранее определенными свойствами.
- 7) *Обучение.* Под обучением понимается использование компьютера для обучения какой-то дисциплине или предмету. Системы обучения диагностируют ошибки при изучении какой-либо дисциплины с помощью ЭВМ.
- 8) *Управление.* Под управлением понимается функция организованной системы, поддерживающая определенный режим деятельности. Такие ЭС осуществляют управление поведением сложных систем в соответствии с заданными спецификациями.
- 9) *Поддержка принятия решений.* Поддержка принятия решений – это совокупность процедур, обеспечивающая лицо, принимающее решения, необходимой информацией и рекомендациями, облегчающими процесс принятия решения. Эти ЭС помогают специалистам выбрать и сформировать оптимальное решение в конкретной ситуации.

Все ЭС включают в себя по крайней мере три основных элемента: базу знаний (БЗ), машину вывода, интерфейс пользователя (рис. 1).

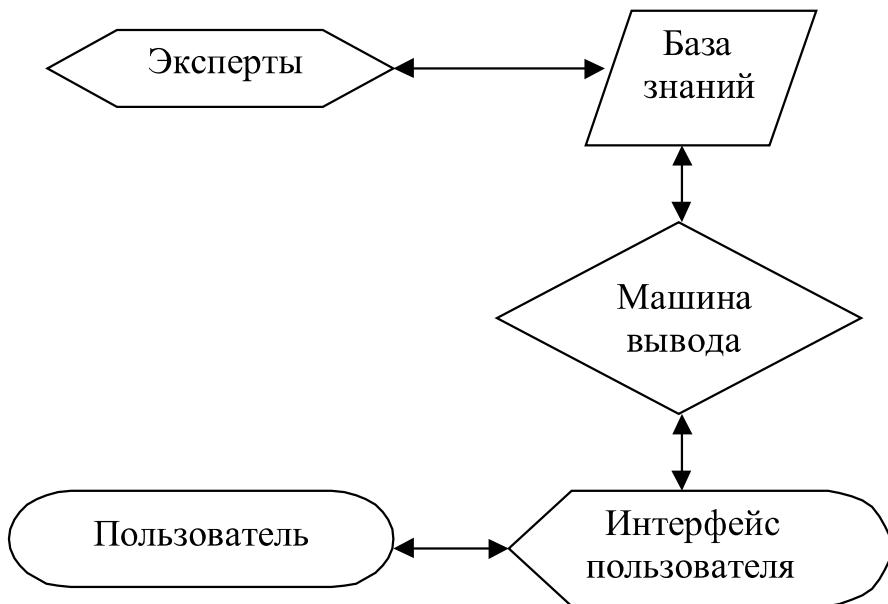


Рис. 1 Структура экспертной системы

Группа экспертов или иной источник экспертизы обеспечивает загрузку в базу знаний фактов, наблюдений и способов анализа ситуаций. Пользователь запрашивает систему о конкретных проблемах через интерфейс, который допускает общение с использованием обычных выражений. Информация, содержащаяся в базе знаний, обрабатывается с помощью машины вывода, которая использует эмпирические ассоциации—или правила «ЕСЛИ-ТО» для формирования и проверки возможных решений. Интерфейс пользователя в доступной форме передает полученные результаты оператору.

База знаний содержит известные факты, выраженные в виде объектов, атрибутов и условий. Помимо описательных представлений о действительности, она включает выражения неопределенности—ограничения на достоверность факта. В этом отношении база знаний отличается от традиционной базы данных. При обработке информации в базе данных пользуются заранее определенными логическими правилами. Соответственно, база знаний, представляющая более высокий уровень абстракции, имеет дело с классами объектов, а не с самими объектами.

Главным в ЭС является машина вывода, осуществляющая поиск в базе знаний по правилам рациональной логики для получения решений. Она приводится в действие при получении запроса пользователя и выполняет следующие задачи: сравнивает информацию, содержащуюся в запросе пользователя, с информацией базы знаний, ищет определенные цели или причинные связи, оценивает относительную определенность фактов, основываясь на соответствующих коэффициентах доверия, связанных с каждым фактом. Действие машины вывода аналогично рассуждениям

эксперта-человека, который оценивает проблему и предлагает гипотетические решения. В поиске целей на основе предложенных правил, машина вывода обращается к базе знаний до тех пор пока не найдет вероятный путь к получению приемлемого результата.

Задача интерфейса пользователя состоит в организации обмена информацией между оператором и машиной вывода. Интерфейс с использованием естественного языка создает видимость беседы.

Эффективность применения ЭС существенно возросла с введением понятия нечетких знаний.

Нечеткая логика - надмножество Булевой логики, которая была расширена с целью обработки концепции частичной правды - значения истинности между "полностью истинным" и "полностью ложным". Она была введена доктором Лотфи Заде (США) в 1965 году как способ моделирования неопределенностей естественного языка. Его основная идея состояла в том, что человеческий способ рассуждений, опирающийся на естественный язык, не может быть описан в рамках традиционных математических понятий. Этим понятиям присуща строгая однозначность интерпретации, а все, что связано с использованием естественного языка, имеет многозначную интерпретацию.

Таким образом, основной целью введения нечеткой логики является создание аппарата, способного моделировать человеческие рассуждения и объяснить человеческие приемы принятия решений в ходе решения различных задач. В настоящее время нечеткая логика применяется при разработке систем, понимающих тексты на естественном языке, при создании планирующих систем, опирающихся на неполную информацию, для обработки зрительных сигналов, при управлении техническими, социальными и экономическими системами, в системах искусственного интеллекта и робототехнических системах.

Определим понятие нечеткого множества на конкретном примере. Рассмотрим множество X всех чисел от 0 до 10. Определим подмножество A множества X всех действительных чисел от 5 до 8 ($A = [5,8]$). Введем функцию принадлежности элемента множества X множеству A . Эта функция ставит в соответствие число 1 или 0 каждому элементу в X , в зависимости от того принадлежит данный элемент подмножеству A или нет.

Можно интерпретировать элементы, которым поставлена в соответствие 1, как элементы, находящиеся в множестве A , а элементы, которым поставлен в соответствие 0, как элементы, не находящиеся в множестве A . Эта концепция используется во многих областях приложений. Но существуют ситуации, в которых данной концепции будет недоставать гибкости.

В следующем примере опишем множество молодых людей. Более формально можно записать так: $B = \{\text{множество молодых людей}\}$. Так как возраст начинается с 0, то нижний предел этого множества должен быть

нулем. Верхний предел определить сложнее. Установим верхний предел равным 20 годам. Таким образом, получаем В как четко ограниченный интервал $B = [0,20]$.

Возникает вопрос: почему кто-то в свой двадцатилетний юбилей - молодой, а сразу на следующий день уже не молодой? Очевидно, если передвинуть верхнюю границу в другую точку, то возникает точно такой же вопрос. Более естественный путь получения множества В состоит в ослаблении строгого разделения на молодых и не молодых. Сделаем это, вынося не только четкие суждения «Да, он принадлежит множеству молодых людей» или «Нет, он не принадлежит множеству молодых людей», но и более гибкие формулировки «Да, он принадлежит к достаточно молодым людям» или «Нет, он не очень молод».

В первом примере мы кодировали все элементы множества с помощью 0 или 1. Простой способ обобщить данную концепцию - ввести значения между 0 и 1. В действительности может существовать бесконечное число значений в интервале $[0, 1]$. При этом число 1 ставится в соответствие тому элементу, который принадлежит множеству В, а 0 означает, что элемент точно не принадлежит множеству В. Все другие значения определяют степень принадлежности элемента множеству В. Тогда утверждение о том, что человек в возрасте 25 лет все еще молод истинно, например, на 50%.

Для комбинирования нецелочисленных значений истинности в нечеткой логике определяются эквиваленты операций И, ИЛИ и НЕТ.

$$\begin{aligned} p1 \text{ И } p2 &= \min(p1, p2) \text{ (т.е. меньшее)} \\ p1 \text{ ИЛИ } p2 &= \max(p1, p2) \text{ (т.е. большее)} \\ \text{НЕ } p1 &= 1 - p1 \text{ (т.е. обратное значение).} \end{aligned}$$

Существенной в нечеткой логике является проблема взвешивания сведений. Предположим, что имеется следующий набор продукционных правил:

Правило 1:

ЕСЛИ х программирует на ЭВМ
И х получает новую информацию через Интернет
ТО х выберет специальность по информатике.

Правило 2:

ЕСЛИ х не склонен к изучению гуманитарных наук
И х не любит доказывать теоремы
ТО х выберет специальность по информатике.

Предположим, что мы видели, как x программировал задачу на компьютере (определенность равна 1) и вполне уверены (0,8), что x получает новую информацию через Интернет. Тогда, условия, входящие в правило 1, имеют совместное значение степени истинности, равное 0.8, поскольку в случае логической функции И мы используем операцию \min .

Для правила 2 мы знаем, что «x не склонен к изучению гуманитарных наук» (0.5) и «x не любит доказывать теоремы» (степень истинности 0.25), тогда степень истинности заключения «x выберет специальность по информатике» равна 0.25 (меньшему из значений).

Таким образом, возникает проблема определения результирующей степени истинности заключения, на основании приведенных двух правил. Следует отметить, что исследование таких проблем относится в большей степени к теории свидетельств, чем к нечеткой логике.

Схема, использующая свидетельства, для получения степени уверенности была предложена Шортлиффом и применяется в ЭС MYCIN. Она основывается на коэффициентах уверенности, предназначенных для измерения степени доверия к заключению, которое является результатом полученных свидетельств. Коэффициент уверенности – это разность между двумя мерами:

$$КУ[h:e] = МД[h:e] - МНД[h:e],$$

где: КУ[h:e] – уверенность в гипотезе h с учетом свидетельств e; МД[h:e] – мера доверия гипотезе h при заданном e; МНД[h:e] – мера недоверия h при свидетельствах e.

Коэффициент КУ может изменяться от -1 (абсолютная ложь) до 1 (абсолютнаястина). Значения МД и МНД могут изменяться только от 0 до 1. Заметим, что эта формула не позволяет отличать случай противоречащих свидетельств, когда МД и МНД велики, от случая недостаточной информации, когда МД и МНД очень малы. КУ, МД и МНД не являются вероятностными мерами .

Шортлиффом была предложена формула уточнения, по которой новую информацию можно сочетать со старыми результатами. Она применяется к мерам доверия и недоверия, связанным с каждой гипотезой. Формула для меры доверия имеет следующий вид:

$$МД[h:e1,e2]=МД[h:e1]+МД[h:e2](1-МД[h:e1]),$$

где запятая между e1 и e2 означает, что e2 следует за e1. Аналогичным образом уточняются значения меры недоверия.

Смысл формулы состоит в том, что влияние второго свидетельства e_2 на гипотезу h при заданном свидетельстве e_1 оказывается в смещении меры доверия в сторону полной определенности на расстояние, зависящее от второго свидетельства. Эта формула имеет два важных свойства:

- а) она симметрична относительно следования e_1 и e_2 ;
- б) по мере накопления подкрепляющих свидетельств МД (или МНД) движется в сторону полной определенности.

Рассмотрим пример, указывая в скобках значение МД для свидетельств.

Правило 1:

ЕСЛИ x программирует на ЭВМ (0.75)

И x не любит теоретические дисциплины (0.6)

ТО x выберет специальность по информатике.

Правило 2:

ЕСЛИ x любит увлекаться точными науками (0.5)

ИЛИ любит практику на ЭВМ (0.7)

ТО x выберет специальность по информатике.

Операция И в первом правиле определяет минимальное из значений 0.75 и 0.6 , т.е. 0.6. Операция ИЛИ во втором правиле требует взятия максимального из значений 0.5 и 0.7, т.е. 0.7.

Тогда гипотеза, что “ x выбирает специальность по информатике” поддерживается на уровне 0.6 правилом 1 и на уровне 0.7 правилом 2. Применяя приведенную формулу получаем

$$\text{МД[информатика: правило 1, правило 2]} = \text{МД[информатика: правило 1]} + \\ + \text{МД[информатика: правило 2]} (1 - \text{МД[информатика: правило 1]}) = 0.88.$$

Таким образом, объединенная мера доверия оказывается выше, чем при учете каждого свидетельства, взятого отдельно. Это согласуется с ожидаемым нами результатом, поскольку несколько показывающих одно и то же направление свидетельств подкрепляют друг друга. Следует отметить, что если поменять порядок правил 1 и 2, то на результате это не отразится. Такой набор правил с успехом использовался в ЭС MYCIN, что привело к их широкому применению в последующих разработках.

2. ПРИНЦИПЫ ПОСТРОЕНИЯ КОМПОНЕНТОВ ЭКСПЕРТНЫХ СИСТЕМ

Идеальная ЭС должна содержать пять основных подсистем: интерфейс с пользователем, систему логического вывода (механизм вывода), базу знаний, составляющих ядро любой ЭС, а также модуль приобретения знаний, модуль отображения и объяснения решений (рис. 4.1) [4].

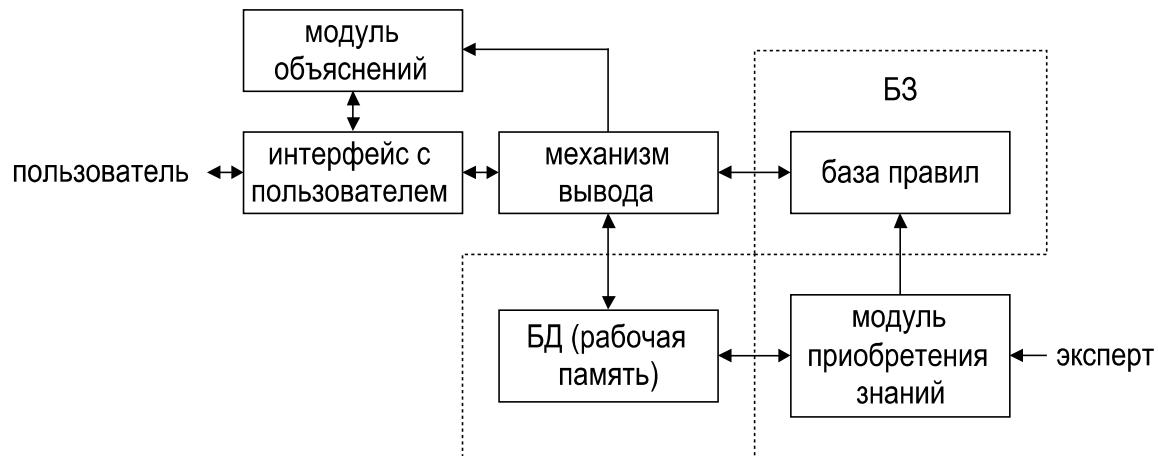


Рис. 4.1. Структура связей между подсистемами ЭС

2.1. Механизм вывода (интерпретатор правил)

Механизм вывода (интерпретатор правил) выполняет две функции: во-первых, просмотр существующих фактов из рабочей памяти (БД) и правил из БЗ, добавление (по мере возможности) в рабочую память новых фактов и, во-вторых, определение порядка просмотра применения правил. Этот механизм управляет процессом консультации, сохраняя для пользователя информацию о полученных заключениях, и запрашивает у него информацию, когда для очередного правила в рабочей памяти оказывается недостаточно данных [4].

В некоторых системах принят прямой порядок вывода: от фактов, которые находятся в рабочей памяти, к заключению. В других системах вывод осуществляется в обратном порядке: заключения просматриваются последовательно до тех пор, пока не будут обнаружены в рабочей памяти или получены от пользователя факты, подтверждающие одно из них. В подавляющем большинстве систем, основанных на знаниях, механизм вывода представляет собой небольшую по объему программу. Основную же часть памяти компьютера занимают правила.

Механизм вывода включает в себя два компонента: один из них реализует собственно вывод, другой управляет этим процессом. Компонент вывода выполняет первую задачу, просматривая имеющиеся правила и факты из рабочей памяти, и добавляет в нее новые факты при срабатывании

какого-нибудь правила. Управляющий компонент определяет порядок применения правил. Представляет интерес рассмотреть каждый из этих компонентов более подробно.

2.1.1. Компонент вывода

Его действие основано на применении правила вывода, суть которого состоит в следующем:

Пусть известно, что истинно утверждение *A* и существует правило вида «ЕСЛИ *A*, ТО *B*», тогда утверждение *B* также истинно. Правила срабатывают, когда находятся факты, удовлетворяющие их левой части: если истинна посылка, то должно быть истинно и заключение.

Хотя в принципе на первый взгляд кажется, что такой вывод легко может быть реализован на компьютере, тем не менее на практике человеческий мозг все равно оказывается более эффективным при решении задач. Например:

“Мэри искала ключ”.

Для слова «ключ» допустимы как минимум два значения: «родник» и «ключ от квартиры». В следующих же двух предложениях одно и то же слово имеет совершенно разные значения:

“Мы заблудились в чащे”.

“Нужно чаще ходить в театры”.

Понять факты становится еще сложнее, если они являются составными частями продукции. Например:

ЕСЛИ Белый автомобиль легко заметить ночью

И Автомобиль Джека белый

ТО Автомобиль Джека легко заметить ночью.

Это заключение легко выведет даже ребенок, но оно оказывается не под силу ЭС.

Подводя итоги, можно сказать, что человек способен вывести большое число заключений с помощью очень большой базы знаний, которая хранится в его памяти; экспертные же системы могут вывести только сравнительно небольшое число заключений, используя заданное множество правил.

Компонент вывода должен обладать способностью функционировать в условиях недостатка информации. Механизм вывода должен быть способен продолжить рассуждения и со временем найти решения даже при недостатке информации. Это решение может и не быть точным, однако система ни в коем случае не должна останавливаться вследствие отсутствия какой-либо части входной информации.

2.1.2. Управляющий компонент

Управляющий компонент определяет порядок применения правил, а также устанавливает, имеются ли еще факты, которые могут быть изменены

в случае продолжения консультации. Управляющий компонент выполняет четыре функции:

- Сопоставление – образец правила сопоставляется с имеющимися фактами.
- Выбор – если в конкретной ситуации могут быть применены сразу несколько правил. В этом случае из них выбирается одно, наиболее подходящее по заданному критерию (разрешение конфликта).
- Срабатывание – если образец правила при сопоставлении совпадает с какими-либо фактами из рабочей памяти, то правило срабатывает.
- Действие – рабочая память подвергается изменению путем добавления в нее заключения сработавшего правила. Если в правой части правила содержится указание на какое-либо действие, то оно выполняется (как, например, в системах обеспечения безопасности информации).

Интерпретатор правил работает циклически. В каждом цикле он просматривает все правила, чтобы выявить среди них те, посылки которых совпадают с известными на данный момент фактами из рабочей памяти. Интерпретатор определяет также порядок применения правил. После выбора правило срабатывает, его заключение заносится в рабочую память, и затем цикл повторяется сначала.

В одном цикле может сработать только одно правило. Если несколько правил успешно сопоставлены с фактами, то интерпретатор производит выбор по определенному критерию единственного правила, которое и сработает в данном цикле. Цикл работы интерпретатора схематически представлен на рис. 4.2.

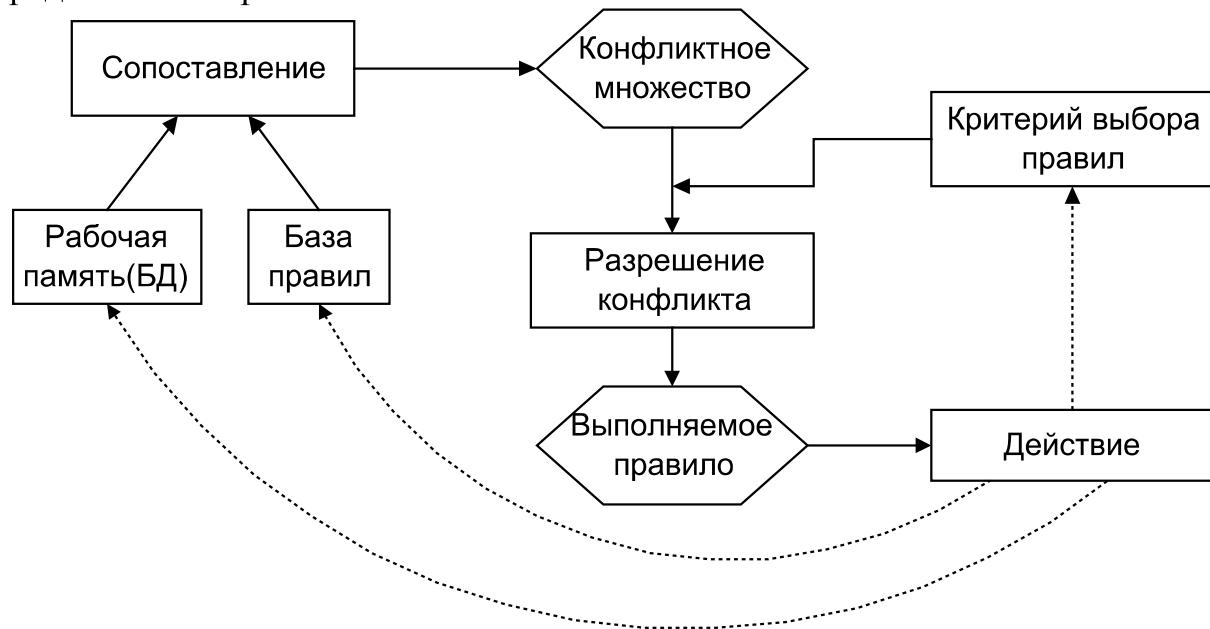


Рис. 4.2. Цикл работы интерпретатора правил

Информация из рабочей памяти последовательно сопоставляется с правилами для выявления успешного сопоставления. Совокупность отобранных правил составляет так называемое конфликтное множество. Для

разрешения конфликта интерпретатор имеет критерий, с помощью которого он выбирает единственное правило, после чего он срабатывает. Это выражается в занесении фактов, образующих заключение правила, в рабочую память или в изменении критерия выбора конфликтующих правил. Если же в заключение правила входит какое – либо действие, то оно выполняется.

Новые данные, введенные в систему сработавшим правилом, в свою очередь, могут изменить критерий выбора правила. В действительности ЭС не располагают процедурами, которые могли бы построить в пространстве состояний сразу весь путь решения задачи. Более того, зачастую даже не удается определить, имеется ли вообще какое-нибудь решение задачи. Тем не менее поиск решения выполняется, поскольку движением в пространстве состояний управляют скрытые или виртуальные процедуры. Такие управляющие процедуры получили название *недетерминированных* процедур. Это означает, что траектория поиска решения в пространстве состояний полностью определяется данными. Следует отметить, что при разработке управляющего компонента механизма вывода существенным является определение критерия выбора правила, которое будет применено в конкретном цикле.

2.2. Взаимодействие пользователей с экспертной системой

Взаимодействие с ЭС осуществляют различные типы пользователей: пользователи – неспециалисты, пользователи – специалисты, эксперты, пользователи – студенты, инженеры по знаниям [9].

Задача пользователей, не являющихся специалистами в области экспертизы, состоит в получении от ЭС решения некоторой задачи.

Задача пользователей – специалистов в области экспертизы заключается в использовании ЭС для сокращения трудоемкости получения результата или повышения его качества.

Задача пользователей – студентов состоит в обучении с помощью ЭС методам решения задач из области экспертизы.

Задача экспертов, т.е. высококвалифицированных специалистов, заключается в обнаружении недостающих знаний и вводе их в систему, т.е. осуществлении отладки знаний.

Задача инженеров по знаниям заключается в отладке управляющего механизма, анализе и модификации ЭС.

Каждый из перечисленных типов пользователей предъявляет свои специфические требования к общению, но всех их объединяет следующее:

- языком общения является ограниченный естественный язык, а не формальный язык программирования;
- процесс взаимодействия пользователей любого типа с ЭС не сводится к обмену изолированными парами предложений («запрос–ответ»), а

представляет собой разветвленный диалог, в котором инициатива переходит от одного участника к другому.

Сложность и обособленность задач, решаемых в процессе общения пользователей с ЭС, приводит к необходимости выделения в структуре ЭС компоненты взаимодействия.

Назначение компоненты взаимодействия состоит в следующем:

- организовать диалог «пользователь – ЭС», т.е. распределить функции участников общения в ходе кооперативного решения задачи и отслеживать состояние диалога как функцию текущих целей участников и фазы решения задачи;
- осуществить обработку отдельного сообщения с учетом текущего состояния диалога, т.е. осуществить преобразование сообщения из естественно-языковой формы в форму внутреннего представления или обратное преобразование.

Общая схема компоненты взаимодействия, состоящей из диалоговой подсистемы и подсистемы анализа и синтеза, приведена на рис. 4.3.

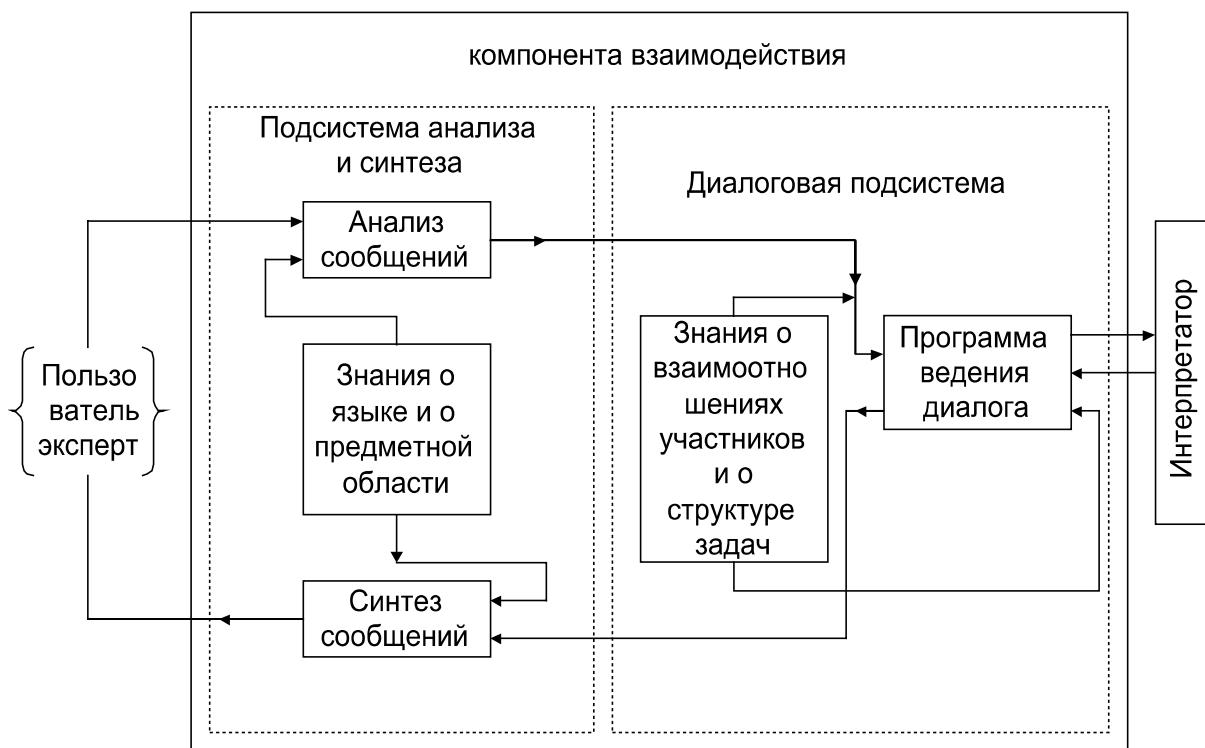


Рис. 4.3. Схема компоненты взаимодействия

В настоящее время не существует полной общепринятой модели диалога. Диалог рассматривается как процесс непротиворечивого взаимодействия участников, в котором они преследуют свои цели с помощью обмена сообщениями на установленном языке общения [9].

Диалог как процесс непротиворечивого взаимодействия участников возможен благодаря согласованности целей участников. При этом цели

известны участникам заранее и дополняют друг друга, например диалог покупка (покупатель/продавец) и т.д. Именно пара целей определяет тип диалога, т.е. множество диалогов, преследующих данные цели, вне зависимости от конкретных участников общения и решаемой ими задачи.

Информация о типе диалога может быть охарактеризована следующими компонентами: множеством параметров, описаниями параметров и макроструктурой диалога.

Множество параметров несет информацию о том, в каких пределах может варьироваться данный тип диалога. Обычно достаточно охарактеризовать любой диалог тремя параметрами: двумя параметрами, определяющими участников (их роли), и предметом (темой) диалога.

Описания параметров содержат множество утверждений о параметрах. В первую очередь эти утверждения определяют цели и состояния участников, которые должны соблюдаться в ходе ведения диалога данного типа. Описания характеризуют те аспекты диалога, которые остаются неизменными на протяжении всего диалога данного типа. Аспекты диалога, изменяемые систематическим образом, представляются в виде общей структуры (макроструктуры) диалога. Макроструктура задается в виде множества подцелей участников, частично упорядоченных во времени.

Элементарную единицу диалога называют *шагом диалога*. Последовательность нескольких шагов диалога называют *диалоговой последовательностью*. Шаг диалога состоит из действия первого участника (инициатора действия) и следующей за ним реакции второго участника. Действие всегда составляет первую часть шага, а реакция – вторую. Термин “реакция” является более предпочтительным, чем термин “ответ”, так как реакция по форме может быть не ответом (в смысле ответа на вопрос), а, например, вопросом. Действие состоит из подготовки и осуществления сообщения инициатором. Реакция состоит из подготовки и осуществления сообщения вторым участником.

При рассмотрении последовательности шагов диалога для определения того, от кого исходит действие, а от кого – реакция, необходимо учитывать смысл. Участники могут перехватывать инициативу, т.е. вместо реакции в ответ на действие первого участника второй участник может совершить действие. Например, вместо ответа на поставленный вопрос задается встречный вопрос. Перехват инициативы необходим при возникновении непонимания, несогласия или недоверия к действиям собеседника.

Шаг диалога характеризуется следующими параметрами:

- инициатор и тип инициирования;
- способ и форма влияния действия на реакцию;
- способ спецификации шага (подзадачи).

Инициатором шага диалога может быть пользователь или система. При этом действия пользователя всегда обозначают выбор или явную формулировку определенной задачи. Действия системы обычно

подразделяются на запрос и предложение. В случае действия-запроса система предлагает пользователю определить (ввести) задачу. В случае действия-предложения система предлагает пользователю выбрать из некоторого ограниченного множества задач интересующую его задачу.

Влияние действия на реакцию обычно представляют в виде следующих форм: команды (задание действия), “меню” (предложение для выбора реакции), анкеты (предложения для выбора значений некоторых сущностей), взаимодействия на естественном языке.

По способу влияния действия на реакцию выделяют:

- свободный (неограниченный) выбор, т.е. действие не накладывает ограничений на вид реакции;
- ограниченный выбор, т.е. действие ограничивает разнообразие реакций. Ограничения на возможные реакции могут быть заданы либо путем указания множества выбора (как в “меню”), либо заданием жесткого формата, который должен быть соблюден в реакции.

По способу спецификации шага можно говорить об автоматической (однозначной) спецификации задачи, обсуждаемой на данном шаге, и о возможной, но не обязательно однозначной спецификации задачи. Например, “меню” и синтаксически правильная “команда” вызывают однозначную спецификацию задачи системой, а высказывания на ограниченном естественном языке не гарантируют однозначной спецификации.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ

Целью лабораторной работы является создание студентом экспертной системы (ЭС).

Отчет по лабораторной работе должен содержать:

1. Цель работы.
2. Постановку задачи.
3. Метод решения задачи.
4. Структурную схему алгоритма.
5. Листинг программы.
6. Результаты работы экспертной системы.
7. Выводы.

Для создания экспертной системы рекомендуется использование языков программирования: Турбо-Пролога, C++, JAVA.

Ниже излагаются варианты тем для разработки экспертных систем. Выбор варианта производится в соответствии с желанием студента, на основании его знаний о предметной области.

4. ВАРИАНТЫ ЗАДАНИЙ

1. ЭС, рекомендующая распределение времени при подготовке к экзаменам.
2. ЭС по выбору темы для бакалаврской работы.
3. ЭС по диагностике состояния здоровья пациента.
4. ЭС по выбору ВУЗА и специальности для абитуриента.
5. ЭС, определяющая тип темперамента человека
6. ЭС по выбору маршрута и способа передвижения из одного населенного пункта в другой.
7. ЭС по принятию финансовых решений в области малого предпринимательства.
8. ЭС по выбору места работы после окончания ТПУ.
9. ЭС, определяющая неисправность автомобиля и дающая рекомендации по ее устраниению.
10. ЭС по выбору автомобиля.
11. ЭС для принятия решения о приеме на работу в компьютерную фирму нового сотрудника.
12. ЭС поиска неисправностей в компьютере.
13. ЭС по выбору стиральной машины.
14. ЭС, рекомендующая конфигурацию персонального компьютера.
15. ЭС, прогнозирующая исход футбольного матча.
16. ЭС по выбору системы защиты информации.
17. ЭС оценки качества программного обеспечения.
18. ЭС, принимающая решения о формировании бюджета семьи.
19. ЭС по определению оптимального маршрута движения автомобиля “Скорой помощи” по вызовам.
20. ЭС по определению типа геологической породы.
21. ЭС, рекомендующая конфигурацию сервера локальной вычислительной сети.
22. ЭС по выбору инструментальных средств при создании WEB сайтов.

ЛИТЕРАТУРА

1. Гавrilova T.A. , Хорошевский В.Ф. Базы знаний интеллектуальных систем. – Санкт-Петербург: Питер, 2000. – 382 с.
2. Корнеев В.В., Гареев А.Ф., Васютин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. – М.: “Нолидж”, 2000. – 352 с.
3. Змитрович А.И. Интеллектуальные информационные системы. –

- Минск: Тетра Системс, 1997. – 367 с.
4. Таунсенд К., Фохт Д. Проектирование и программная реализация экспертных систем на персональных ЭВМ / Пер.с англ. – М.: Финансы и статистика, 1990. – 320 с.
 5. Искусственный интеллект: Кн. 1. Системы общения и экспертные системы. Справочник / Под ред. Э.В. Попова. – М.: Радио и связь, 1990. – 464 с.
 6. Ин Ц., Соломон Д. Использование Турбо – Пролога / Пер. с англ. – М.: Мир, 1993. – 608 с.
 7. Марселлус Д. Программирование экспертных систем на Турбо Прологе / Пер.с англ. – М.: Финансы и статистика, 1994. – 256 с.
 8. Осуга С. Обработка знаний / Пер. с японск. – М.: Мир, 1989. – 293 с.
 9. Попов Э.В. Экспертные системы. – М.: Наука, 1987, – 288 с.
 10. Экспертные системы. Принцип работы и примеры / Под ред. Р. Форсайда; Пер.с англ. – М.: Радио и связь, 1987. – 221 с.
 11. Куффлер С.В., Николс Дж.Г. От нейрона к мозгу. – М: Мир. 1978. – 439 с.
 12. Джексон П. Введение в экспертные системы / Пер. с англ. – М.: Издательский дом “Вильямс”, 2001. – 624 с.
 13. <http://softlab.od.ua/algo/neuro/fuzzy-intro/index.htm> – Кряжевских С.В. Введение в нечеткую логику и системы нечеткого управления. 1997.
 14. Спицын В.Г. Базы знаний и экспертные системы: Учеб. пособие – Томск: Изд. ТПУ, 2001. – 88 с.