

1. Представление знаний, рассуждений и задач. Классификация методов обработки знаний.

Представление (representation) в работе Уинстона [Winston, 1984] определяется как «множество синтаксических и семантических соглашений, которое делает возможным описание предмета».

Человек одновременно использует самые различные методы представления знаний: языковое описание, графическая информация, математические формулы и т.д. В зависимости от специфики той или иной области деятельности один или несколько видов описания будут превалировать над остальными. Например, в математике формулы и графики будут преобладать над текстовой информацией, в юриспруденции доминирует текстовая информация, в искусствоведении преобладает текстовая и графическая информация. В случае выбора средств представления знаний в технических системах такой универсализм представления знаний невозможен, поскольку потребует аппаратной и программной реализации интеллектуальных функций, недостижимых в настоящее время. Поэтому, существует несколько базовых специализированных машинных языков представления знаний, каждый из которых наиболее предпочтителен для той или иной предметной области

Язык продукционных правил.

Продукционные правила – это правила, имеющие форму: ЕСЛИ «Условие» – ТО «Событие». Продукционные правила описывают знания в виде взаимосвязей типа: «причина» – «следствие», «явление» – «реакция», «признак» – «факт» и т.п

Язык семантических сетей.

Знаниями можно назвать описания отношений между абстрактными понятиями и сущностями, являющимися конкретными объектами реального мира. Изначально семантические сети разрабатывались как модели долговременной человеческой памяти в психологии, но впоследствии эта модель перекочевала в инженерию знаний.

Язык логики предикатов.

Логика предикатов является разделом математики – математической логикой, имеющей большую историю. Данная область математики традиционно составляла математический фундамент, закладываемый в основу формального описания систем.

Язык фреймов

Фреймовая система представления знаний является моделью описания человеческих знаний в виде связанной совокупности крупных структурных единиц, каждая из которых содержит данные, описывающие определенную ситуацию. Во фреймовой системе единицей представления является объект, называемый фреймом. Фрейм содержит совокупность некоторых понятий и сущностей, с помощью которой можно описать конкретную ситуацию. Фрейм имеет уникальное имя и внутреннюю структуру, состоящую из множества упорядоченных элементов – слотов. Каждый слот имеет уникальное в пределах своего фрейма имя и содержит определенную информацию. Таким образом, каждый фрейм это структура данных, описывающая определенную ситуацию, место, объект и т.п. Структура данных внутри фрейма может иметь различный вид: граф, таблица и т.п.

2. Системы принятия решений, основанные на знаниях. Формализация интеллектуальных задач

Системы, основанные на знаниях, — это компьютерные программы, спроектированные таким образом, чтобы воспроизводить работу экспертов в заданных областях знания.

По связи с реальным временем выделяют статические ЭС и динамические ЭС.

По степени интеграции в другие ИС выделяют автономные и интегрированные ЭС

По используемым методам решения задач выделяют логические и логико-математические (гибридные или расчетно-логические) ЭС. Первые являются классическим примером систем, основанных на знаниях. Вторые стали ответом на расширяющиеся потребности, которые возникают, в частности в интегрированных системах, а также при поддержке принятия решений, когда необходимы не только знания экспертов, но и математические обоснования.

Типы решаемых задач с помощью систем, основанных на знаниях: интерпретация символов или сигналов, диагностика, мониторинг, проектирование, прогнозирование, планирование, управление, обучение

Основные типы систем, основанных на знаниях.

1. Экспертные системы моделируют процесс принятия решения, свойственный человеческому мозгу. Они используются для того, чтобы действовать подобно экспертам-людям. Хотя экспертные системы служат для того, чтобы воспроизвести процесс принятия решений, свойственный человеку, алгоритмы и правила, на основе которых принимаются решения, являются статическими.
2. Нейронные сети моделируют работу человеческого мозга на биологическом уровне. Это означает, что нейронные сети обладают встроенной способностью к самообучению: они могут учиться распознавать шаблоны, а затем на основе шаблонов распознавать образы.
3. Системы, основанные на прецедентах, моделируют человеческую способность принимать решение по аналогии с уже имеющимися прецедентами
4. Системы, построенные на генетических алгоритмах, как явствует из названия, моделируют процесс развития биологических организмов на клеточном уровне.
5. Интеллектуальные агенты — это программы, для которых задана конечная цель, но они могут самостоятельно выбирать пути достижения этой цели, гибко изменяя алгоритм своего поведения. В качестве примеров интеллектуальных агентов можно назвать интернет-роботы поисковых систем, которые непрерывно просеивают миллионы страниц различных сай-тов, чтобы поддерживать данные поисковой машины в актуальном состоянии.
6. Системы добычи данных. Термин «добыча данных» используют для того, чтобы описать процесс обнаружения знаний путем нахождения ранее неизвестных взаимосвязей между данными. В базах, в которых хранятся большие массивы данных, можно выявить неявные закономерности, имеющие практическую пользу.

Основными задачами в процессе формализации являются проблемы структуризации исходной задачи и знаний в выбранном (разработанном) формализме, а именно:

структуризация общей задачи на связанные подзадачи;

структуризация предметной области на основе иерархии классов;

структуризация знаний на декларативные и процедурные;

структуризация приложения на основе иерархии «часть/целое».

3. Модели представления знаний: алгоритмические, логические, сетевые и продукционные модели; сценарии

продукционные модели – модели основанные на правилах, позволяют представить знание в виде предложений типа: «ЕСЛИ условие, ТО действие». Продукционные модели обладает тем недостатком, что при накоплении достаточно большого числа правил, они начинают противоречить друг другу;

сетевые модели (или семантические сети) – в инженерии знаний под ней подразумевается граф, отображающий смысл целостного образа. Узлы графа соответствуют понятиям и объектам, а дуги – отношениям между объектами. Обладает тем недостатком, что однозначного определения семантической сети в настоящее время отсутствует;



Первый подход, называемый эмпирическим, основан на изучении принципов организации человеческой памяти и моделировании механизмов решения задач человеком.

Второй подход можно определить как теоретически обоснованный, гарантирующий правильность решений.

Логическая модель знаний Основная идея– вся информация, необходимая для решения прикладных задач, рассматривается как совокупность фактов и утверждений, которые представляются как формулы в некоторой логике. Знания отображаются совокупностью таких формул, а получение новых знаний сводится к реализации процедур логического вывода.

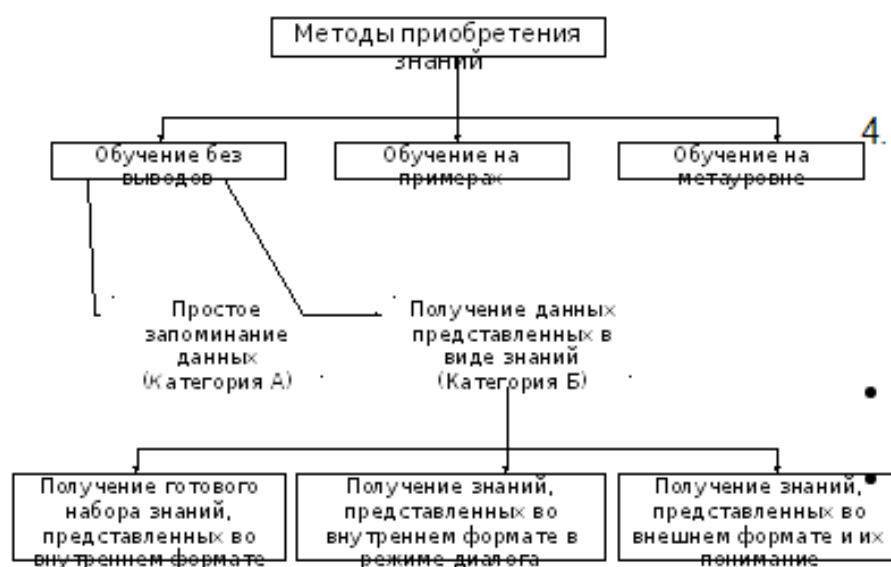
В логических моделях знаний слова, описывающие сущности предметной области, называются термами (константы, переменные, функции), а слова, описывающие отношения сущностей – предикатами.

Алгоритмическая модель также может отображаться графом, где в корневой вершине находится исходная решаемая задача, в промежуточных вершинах – подзадачи, в конечных – элементарные подзадачи. Дуги отображают операции программирования типа <соглашение>. Алгоритмическая модель представления знаний является основополагающей по отношению ко всем остальным моделям представления знаний. Кроме этого, алгоритмическая модель представления знаний позволяет представить в памяти ЭВМ процедурные знания.

Сценарием называется формализованное описание стандартной последовательности взаимосвязанных фактов, определяющих типичную ситуацию предметной области.

4. Методы приобретения знаний. Методы выявления знаний при работе с экспертами и базами данных

Приобретение знаний реализуется с помощью двух функций: получения информации извне и ее систематизации.



5. Базы знаний. Системы логического вывода

База знаний — база данных, содержащая правила вывода и информацию о человеческом опыте и знаниях в некоторой предметной области. В самообучающихся системах база знаний также содержит информацию, являющуюся результатом решения предыдущих задач.

База знаний — важный компонент интеллектуальной системы. Наиболее известный класс таких программ — это экспертные системы. Они предназначены для поиска способов решения проблем из некоторой предметной области, основываясь на записях БЗ и на пользовательском описании ситуации.

Простые базы знаний могут использоваться для создания экспертных систем хранения данных в организации: документации, руководств, статей технического обеспечения. Главная цель создания таких баз — помочь менее опытным людям найти уже существующее описание способа решения какой-либо проблемы.

Двумя наиболее важными требованиями к информации, хранящейся в базе знаний интеллектуальной системы, являются:

1. Достоверность конкретных и обобщённых сведений, имеющихся в базе данных;
2. Релевантность информации, получаемой с помощью правил вывода базы знаний.

Ниже перечислены некоторые из особенностей, которые могут (но не обязаны) быть у системы, оперирующей базами знаний.

- Автоматическое доказательство (вывод). Способность системы выводить новые знания из старых, находить закономерности в БЗ. Часто принимается, что база знаний отличается от базы данных именно наличием механизма вывода.
- Доказательство заключения. Способность системы после выдачи ответа «объяснить» ход её рассуждений, причем «по первому требованию».
- Интроспекция. Нахождение противоречий, нестыковок в БЗ, контроль правильной организации БЗ.
- Машинное обучение. Превращение БЗ в гибкую систему, адаптация к проблемной области. Аналогична человеческой способности «набирать опыт».

Машина вывода — программа, которая выполняет логический вывод из предварительно построенной базы фактов и правил в соответствии с законами формальной логики.

Правило в данном случае — параметризованное утверждение, состоящее из двух частей: условия и результата, например: «Если X является потомком человека, то X является человеком». В данном примере правило определяет условие «X является потомком человека». Если для значения параметра X это условие истинно, то правило превращается в факт «X является человеком».

Для построения базы фактов и правил применяются формальные языки, обычно напоминающие естественный язык, но гораздо более строгие и ограниченные.

Машина вывода обычно может выполнять одну или обе из следующих операций: Проверка истинности некоторого факта — истинным считается факт, если он может быть выведен по законам формальной логики из имеющейся базы фактов и правил. Нахождение множества значений параметра некоторого правила, при котором данное правило превращается в истинный факт. Механизм вывода — это обобщенная процедура поиска решения задачи, которая на основе базы знаний и в соответствии с информационной потребностью пользователя строит цепочку рассуждений (логически связанных единиц знаний), приводящую к конкретному результату

6. Языки искусственного интеллекта - общие черты и различия.

Это прежде всего Лисп (LISP) и Пролог (Prolog) – наиболее распространенные языки, предназначенные для решения задач искусственного интеллекта. Есть и менее распространенные языки искусственного интеллекта, например РЕФАЛ, разработанный в России.

Универсальность этих языков меньшая, нежели традиционных языков, но ее потерю языки искусственного интеллекта компенсируют богатыми возможностями по работе с символьными и логическими данными, что крайне важно для задач искусственного интеллекта. На основе языков искусственного интеллекта создаются специализированные компьютеры (например, Лисп-машины), предназначенные для решения задач искусственного интеллекта. Недостаток этих языков – неприменимость для создания гибридных экспертных систем.

Лисп

Концепция, которую он представляет, называется функциональным программированием, она является прямым продолжением обычного алгоритмического подхода. Лисп-программа представляет собой функцию, результат вычисления которой - это результат работы программы, а аргументы, чаще всего - другие вызовы функций.

Пролог

использует подход к программированию, принципиально отличный от алгоритмического и называемый целевым или декларативным программированием. При алгоритмическом программировании мы задаем последовательность действий, которые должна выполнять программа, т.е. описываем, КАК она должна работать. При декларативном программировании мы описываем, ЧТО программа должна делать, а то, как будут осуществлены эти действия - дело Пролог-системы. В задачах, связанных с заданием отношений между объектами, Пролог гораздо мощнее алгоритмических языков типа Паскаля или Си. Если добавить к этому, что база данных Пролога (содержащая факты и правила) может динамически изменяться во время выполнения самой же программой или пользователем, становится ясно, насколько полезен Пролог для разработки в области искусственного интеллекта. Для того чтобы инициировать вычисления, выполняется специальный запрос к базе знаний, на которые система логического программирования генерирует ответы «истина» и «ложь». Для обобщённых запросов с переменными в качестве аргументов созданная система Пролог выводит конкретные данные в подтверждение истинности обобщённых сведений и правил вывода.

7. Язык исчисления предикатов. Интерпретация и истинность.

С помощью предикатов можно формулировать содержательные утверждения в различных областях знания. Поэтому важно дать средства построения осмысленных выражений с предикатами и приписывания им истинностных значений подобно тому, как это было сделано в исчислении высказываний.

Выберем множество истинностных значений V . Также, выберем некоторое предметное множество D . n -местным предикатом мы назовем функцию из D^n в V . Как и раньше, мы ограничимся классическим множеством V -- истина и ложь, но оставляем потенциальную возможность его расширить.

Предикаты могут быть 0-местными, в этом случае это хорошо нам известные пропозициональные переменные, принимающие какие-то истинностные значения, в происхождение которых мы не вникаем.

символы переменных

логические операции

Символ	Значение
\neg	Отрицание («не»)
\wedge	Конъюнкция («и»)
\vee	Дизъюнкция («или»)
\rightarrow	Импликация («если ..., то ...»)

• кванторы:

Символ	Значение
\forall	Квантор всеобщности
\exists	Квантор существования

Формула называется истинной, если она выполнена на любом наборе элементов.

8. Язык ЛИСП, ПРОЛОГ.

Это прежде всего Лисп (LISP) и Пролог (Prolog) – наиболее распространенные языки, предназначенные для решения задач искусственного интеллекта. Есть и менее распространенные языки искусственного интеллекта, например РЕФАЛ, разработанный в России.

Универсальность этих языков меньшая, нежели традиционных языков, но ее потерю языки искусственного интеллекта компенсируют богатыми возможностями по работе с символьными и логическими данными, что крайне важно для задач искусственного интеллекта. На основе языков искусственного интеллекта создаются специализированные компьютеры (например, Лисп-машины), предназначенные для решения задач искусственного интеллекта. Недостаток этих языков – неприменимость для создания гибридных экспертных систем.

Лисп

Концепция, которую он представляет, называется функциональным программированием, она является прямым продолжением обычного алгоритмического подхода. Лисп-программа представляет собой функцию, результат вычисления которой - это результат работы программы, а аргументы, чаще всего - другие вызовы функций.

Пролог

Пролог (Prolog) — язык логического программирования, основанный на логике дизъюнктов Хорна, представляющей собой подмножество логики предикатов первого порядка.

использует подход к программированию, принципиально отличный от алгоритмического и называемый целевым или декларативным программированием. При алгоритмическом программировании мы задаем последовательность действий, которые должна выполнять программа, т.е. описываем, КАК она должна работать. При декларативном программировании мы описываем, ЧТО программа должна делать, а то, как будут осуществлены эти действия - дело Пролог-системы. В задачах, связанных с заданием отношений между объектами, Пролог гораздо мощнее алгоритмических языков типа Паскаля или Си. Если добавить к этому, что база данных Пролога (содержащая факты и правила) может динамически изменяться во время выполнения самой же программой или пользователем, становится ясно, насколько полезен Пролог для разработки в области искусственного интеллекта. Для того чтобы инициировать вычисления, выполняется специальный запрос к базе знаний, на которые система логического программирования генерирует ответы «истина» и «ложь». Для обобщённых запросов с переменными в качестве аргументов созданная система Пролог выводит конкретные данные в подтверждение истинности обобщённых сведений и правил вывода.

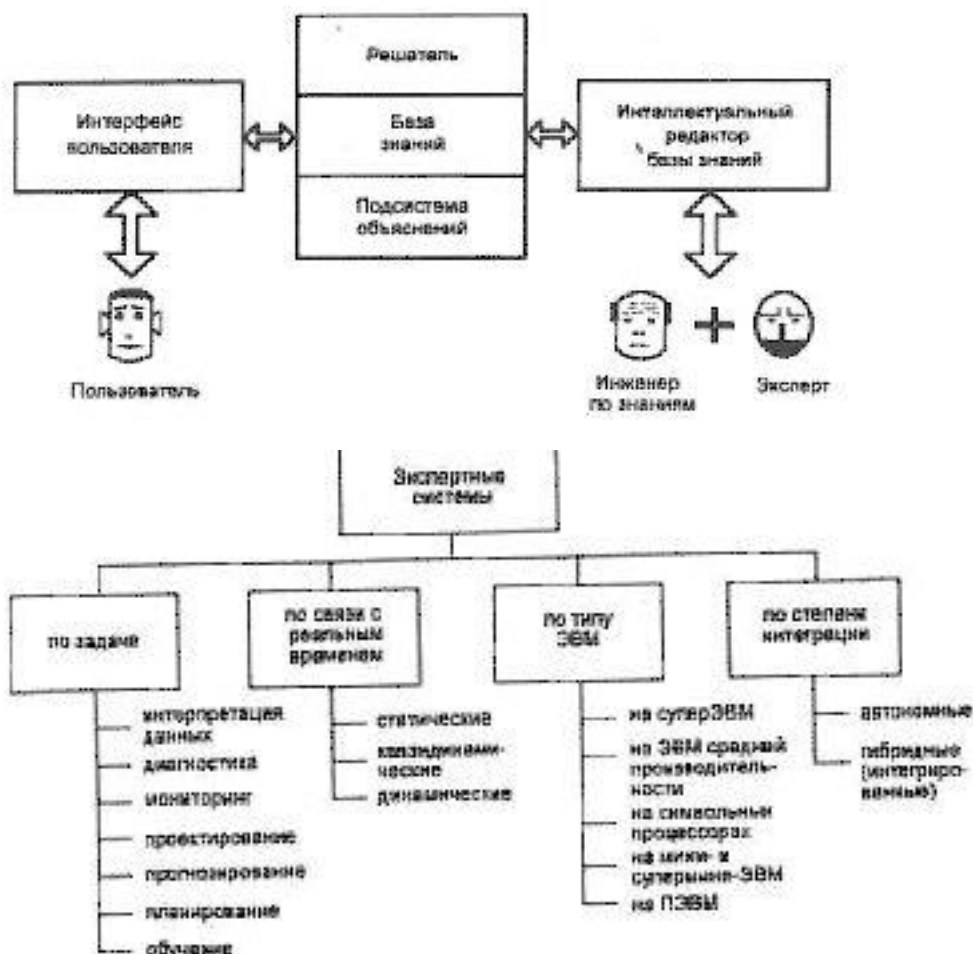
9. Экспертные системы: классификация и структура.

Экспертные системы – это направление исследований в области искусственного интеллекта по созданию вычислительных систем, умеющих принимать решения, схожие с решениями экспертов в заданной предметной области.

Как правило, экспертные системы создаются для решения практических задач в некоторых узкоспециализированных областях, где большую роль играют знания специалистов. Экспертные системы были первыми разработками, которые смогли привлечь большое внимание к результатам исследований в области искусственного интеллекта.

Экспертные системы имеют одно большое отличие от других систем искусственного интеллекта: они не предназначены для решения каких-то универсальных задач, экспертные системы предназначены для качественного решения задач в определенной разработчиками области, в редких случаях – областях.

Экспертное знание – это сочетание теоретического понимания проблемы и практических навыков ее решения, эффективность которых доказана в результате практической деятельности экспертов в данной области. Фундаментом экспертной системы любого типа является база знаний, которая составляется на основе экспертных знаний специалистов. Правильно выбранный эксперт и удачная формализация его знаний позволяет наделить экспертную систему уникальными и ценными знаниями. Поэтому ценность всей экспертной системы как законченного продукта на 90% определяется качеством созданной базы знаний.



10. Экспертные системы: инструментальные средства проектирования, разработки и отладки

Традиционные языки программирования

В эту группу инструментальных средств входят традиционные языки программирования (C, C++, Basic, SmallTalk, Fortran и т.д.), ориентированные в основном на численные алгоритмы и слабо подходящие для работы с символьными и логическими данными. Поэтому создание систем искусственного интеллекта на основе этих языков требует большой работы программистов. Однако большим достоинством этих языков является высокая эффективность, связанная с их близостью к традиционной машинной архитектуре. Кроме того, использование традиционных языков программирования позволяет включать интеллектуальные подсистемы (например, интегрированные экспертные системы) в крупные программные комплексы общего назначения. Среди традиционных языков наиболее удобными считаются объектно-ориентированные (SmallTalk, C++). Это связано с тем, что парадигма объектно-ориентированного программирования тесно связана с фреймовой моделью представления знаний. Кроме того, традиционные языки программирования используются для создания других классов инструментальных средств искусственного интеллекта.

Языки искусственного интеллекта

Это прежде всего Лисп (LISP) и Пролог (Prolog) - наиболее распространенные языки, предназначенные для решения задач искусственного интеллекта. Есть и менее распространенные языки искусственного интеллекта, например РЕФАЛ, разработанный в России. Универсальность этих языков меньшая, нежели традиционных языков, но ее потерю языки искусственного интеллекта компенсируют богатыми возможностями по работе с символьными и логическими данными, что крайне важно для задач искусственного интеллекта. На основе языков искусственного интеллекта создаются специализированные компьютеры (например, Лисп-машины), предназначенные для решения задач искусственного интеллекта. Недостаток этих языков - неприменимость для создания гибридных экспертных систем.

Специальный программный инструментарий

В эту группу программных средств искусственного интеллекта входят специальные инструментарии общего назначения. Как правило, это библиотеки и надстройки над языком искусственного интеллекта Лисп: KEE (Knowledge Engineering Environment), FRL (Frame Representation Language), KRL (Knowledge Representation Language), ARTS и др., позволяющие пользователям работать с заготовками экспертных систем на более высоком уровне, нежели это возможно в обычных языках искусственного интеллекта.

"Оболочки"

Под "оболочками" (shells) понимают "пустые" версии существующих экспертных систем, т.е. готовые экспертные системы без базы знаний. Примером такой оболочки может служить EMYCIN (Empty MYCIN - пустой MYCIN), которая представляет собой незаполненную экспертную систему MYCIN. Достоинство оболочек в том, что они вообще не требуют работы программистов для создания готовой экспертной системы. Требуется только специалисты в предметной области для заполнения базы знаний. Однако если некоторая предметная область плохо укладывается в модель, используемую в некоторой оболочке, заполнить базу знаний в этом случае весьма не просто.

11. Экспертные системы: этапы разработки; примеры реализации.

Этапы разработки экспертных систем

Этап идентификации.

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

Этап концептуализации.

На данном этапе проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач. Этот этап завершается созданием модели предметной области (ПО), включающей основные концепты и отношения.

Этап формализации.

Теперь все ключевые понятия и отношения выражаются на некотором формальном языке, который либо выбирается из числа уже существующих, либо создается заново. Другими словами, на данном этапе определяются состав средств и способы представления декларативных и процедурных знаний, осуществляется это представление и в итоге формируется описание решения задачи ЭС на предложенном (инженером по знаниям) формальном языке. Выходом этапа формализации является описание того, как рассматриваемая задача может быть представлена в выбранном или разработанном формализме.

Этап выполнения.

Цель этого этапа — создание одного или нескольких прототипов ЭС, решающих требуемые задачи. Затем на данном этапе по результатам тестирования и опытной эксплуатации создается конечный продукт, пригодный для промышленного использования. Разработка прототипа состоит в программировании его компонентов или выборе их из известных инструментальных средств и наполнении базы знаний. Главное в создании прототипа заключается в том, чтобы этот прототип обеспечил проверку адекватности идей, методов и способов представления знаний решаемым задачам.

Этап опытной эксплуатации

На этом этапе проверяется пригодность ЭС для конечного пользователя. Пригодность ЭС для пользователя определяется в основном удобством работы с ней и ее полезностью. Под полезностью ЭС понимается ее способность в ходе диалога определять потребности пользователя, выявлять и устранять причины неудач в работе, а также удовлетворять указанные потребности пользователя (решать поставленные задачи).

Примеры реализации

Домашний доктор - простая медицинская экспертная система. Она определяет характер заболевания, основываясь на ответах пользователя, полученных в результате диалога. База знаний включает порядка 100 распространенных заболеваний.

FALCON. Экспертная система определяет вероятные причины отклонений параметров процесса от нормы на химическом заводе, интерпретируя данные, состоящие из числовых значений показаний приборов, положения переключателей и состояния аварийных датчиков.

12. Проектирование экспертных систем

Первые ЭС были статического типа. Типичная статическая ЭС должна включать следующие компоненты:

- базу знаний (БЗ);
- базу данных (рабочую память);
- решатель (интерпретатор);
- систему объяснений;
- компоненты приобретения знаний;
- интерфейс с пользователем.

БЗ ЭС предназначена для хранения долгосрочных данных, описывающих рассматриваемую область, и правил, описывающих целесообразные преобразования данных этой области.

БД ЭС служит для хранения текущих данных решаемой задачи.

Решатель формирует последовательность применения правил и осуществляет их обработку, используя данные из рабочей памяти и знания из БЗ.

Система объяснений показывает, каким образом система получила решение задачи, и какие знания при этом использовались. Это облегчает тестирование системы и повышает доверие пользователя к полученному результату.

Компоненты приобретения знаний необходимы для заполнения ЭС знаниями в диалоге с пользователем-экспертом, а также для добавления и модификации заложенных в систему знаний.

К разработке ЭС привлекаются специалисты из разных предметных областей, а именно:

- эксперты той проблемной области, к которой относятся задачи, решаемые ЭС;
- инженеры по знаниям, являющиеся специалистами по разработке интеллектуальных информационных систем (ИИС);
- программисты, осуществляющие реализацию ЭС.

Любая ЭС должна иметь, по крайней мере, два режима работы:

- режим приобретения знаний;
- режим консультаций.

Динамические ЭС, наряду с компонентами статических ЭС, должны содержать:

- подсистему моделирования внешнего мира;
- подсистему связи с внешним окружением.

Подсистема моделирования необходима для прогнозирования, анализа и адекватной оценки состояния внешней среды. Изменения окружения решаемой задачи требуют изменения хранимых в ЭС знаний, для того чтобы отразить временную логику происходящих в реальном мире событий.

13. Нейронечеткие экспертные системы. Комитеты экспертов

Нечеткая логика (fuzzy logic) является надмножеством классической булевой логики. Она расширяет возможности классической логики, позволяя применять концепцию неопределенности в логических выводах.

Особенности нечетких систем

возможность оперировать нечеткими входными данными: например, непрерывно изменяющиеся во времени значения (динамические задачи), значения, которые невозможно задать однозначно (результаты статистических опросов, рекламные компании и т.д.);

возможность нечеткой формализации критериев оценки и сравнения: оперирование критериями "большинство", "возможно", "преимущественно" и т.д.;

возможность проведения качественных оценок как входных данных, так и выходных результатов: вы оперируете не только значениями данных, но и их степенью достоверности и ее распределением;

возможность проведения быстрого моделирования сложных динамических систем и их сравнительный анализ с заданной степенью точности: оперируя принципами поведения системы, описанными fuzzy-методами, вы во-первых, не тратите много времени на выяснение точных значений переменных и составление описывающих уравнений, во-вторых, можете оценить разные варианты выходных значений.

14. Многоагентные интеллектуальные системы

В классической теории искусственного интеллекта решение какой-либо задачи сводится к созданию некоторой одной интеллектуальной системы, называемой агентом, которая, имея в своем распоряжении все необходимые знания, способности и вычислительные ресурсы, способна решить некоторую глобальную проблему.

Многоагентные системы или мультиагентные системы – это направление искусственного интеллекта, которое для решения сложной задачи или проблемы использует системы, состоящие из множества взаимодействующих агентов

В теории многоагентных систем (также часто встречается название «мультиагентные системы») за основу берется противоположный принцип. Считается, что один агент владеет всего лишь частичным представлением о глобальной проблеме, а значит, он может решить лишь некоторую часть общей задачи. В связи с этим для решения сложной задачи необходимо создать некоторое множество агентов и организовать между ними эффективное взаимодействие, что позволит построить единую многоагентную систему. В многоагентных системах весь спектр задач по определенным правилам распределяется между всеми агентами, каждый из которых считается членом организации или группы. Распределение заданий означает присвоение каждому агенту некоторой роли, сложность которой определяется исходя из возможностей агента.

Для организации процесса распределения задачи в многоагентных системах создается либо система распределенного решения проблемы либо децентрализованный искусственный интеллект. В первом варианте процесс декомпозиции глобальной задачи и обратный процесс композиции найденных решений происходит под управлением некоторого единого «центра». При этом многоагентная система проектируется строго сверху вниз, исходя из ролей определенных для агентов и результатов разбиения глобальной задачи на подзадачи. В случае использования децентрализованного искусственного интеллекта распределение заданий происходит в процессе взаимодействия агентов и носит больше спонтанный характер. Нередко это приводит к появлению в многоагентных системах резонансных, синергетических эффектов.

Мультиагентные системы подразделяются на кооперативные, конкурирующие и смешанные. Агенты в кооперативных системах являются частями единой системы и решают подзадачи одной общей задачи. Понятно, что при этом агент не может работать вне системы и выполнять самостоятельные задачи. Конкурирующие агенты являются самостоятельными системами, хотя для достижения определенных целей они могут объединять свои усилия, принимать цели и команды от других агентов, но при этом поддержка связи с другими агентами не обязательна. Под смешанными агентами понимаются конкурирующие агенты, подсистемы которых также реализуются по агентной технологии. Кроме общения с другими агентами должна быть реализована возможность общения с пользователем

15. Интеллектуальные динамические системы.

Динамическая система — множество элементов, для которого задана функциональная зависимость между временем и положением в фазовом пространстве каждого элемента системы. Данная математическая абстракция позволяет изучать и описывать эволюцию систем во времени.

таких систем, которые, с одной стороны удовлетворяют определениям динамических систем в самом общем виде, с другой, - их динамика описывается правилами, часто используемыми в интеллектуальных системах, а состояния – параметрами и переменными, имеющими символичный характер.

Системы, которые, с одной стороны удовлетворяют определениям динамических систем, с другой - являются интеллектуальными системами, так как их динамика описывается средствами представления знаний, принятыми в интеллектуальных системах, а состояния - параметрами и переменными, имеющими логический или лингвистический характер. Рассматривается случай, когда в качестве способа представления знаний используется система правил.

Важной областью применения описываемых моделей являются двухуровневые системы управления, потребность в которых назрела в связи с развитием, например, беспилотных средств транспортного и иного назначения. В частности, в беспилотных автономных самолетах и вертолётах одним из уровней управления должен являться стратегический (или, как иногда говорят, делиберативный) уровень управления, решающий задачи, например, планирования полёта или выбора траектории или выбора цели. Другой уровень управления - назовем его активным - реализует требуемые действия. Например, на делиберативном уровне управления беспилотным вертолётom принимается решение о зависании над целью, тогда на активном уровне начинает работать математическая модель зависания, вырабатывающая требуемые управления для исполнительных механизмов. Динамическая система такого вида должна включать множество правил, образующих базу знаний системы и некоторые функции, заданные подходящим способом. Кроме того, для хранения текущих значений переменных состояния потребуется рабочая память, а для управления работой правил – стратегии применения правил.