

**ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ**  
(конспект рабочих материалов)

## Оглавление

1. ПОНЯТИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА .....	3
1.1. Новая технология решения задач управления .....	3
1.2. Искусственный интеллект в проблеме принятия решений .....	4
1.3. Развитие исследований в области искусственного интеллекта .....	6
1.4. Теория и практика искусственного интеллекта. ....	8
2. ПОДХОДЫ К СОЗДАНИЮ СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА .....	10
2.1. Экспертные системы .....	10
2.2. Этапы развития научного направления «мягкие вычисления» .....	12
3. ОСНОВНЫЕ НАПРАВЛЕНИЯ, ФУНКЦИИ И КЛАССИФИКАЦИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ.....	18
3.1. Области применения ИИС .....	18
3.2. Признаки классификации интеллектуальных информационных систем.....	20
3.3. Понятие экспертной системы .....	21
3.4. Самообучающиеся системы .....	23
3.5. Адаптивные информационные системы. ....	25
4. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ.....	26
4.1. Хранилища данных .....	26
4.2. Оперативная аналитическая обработка данных и интеллектуальный анализ данных .....	28
4.3. Интеллектуальные базы данных .....	31

1. ПОНЯТИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

1.1. Новая технология решения задач управления

Искусственный интеллект – это термин, который охватывает много определений. Многие специалисты согласны, что ИИ соотносится с двумя базовыми идеями. Во-первых, ИИ вовлечен в процессы изучения мыслительных процессов человека (для понимания, что представляет собой интеллект); во-вторых, ИИ имеет дело с представлением этих процессов через машины (такие как компьютеры и роботы).

Одним из многих определений ИИ является следующее: искусственный интеллект – это такое поведение машины, что если оно совершалось бы человеком, то могло бы быть названо умным, т.е. интеллектуальным.

Если исследовать значение термина *интеллектуальное поведение*, то можно определить различные способности, которые рассматриваются как признаки интеллекта:

- обучение или понимание из опыта;
- выявление смысла из двусмысленности или противоположных сообщений;
- быстрый и адекватный отклик на новую ситуацию (разнообразные реакции, гибкость);
- использование рассуждений при решении проблем и эффективном направлении поведения;
- использование относительной важности различных элементов в ситуации;
- мышление и рассуждение.

Как искусственный интеллект и интеллектуальные технологии отличаются от традиционных вычислений?

*Традиционные компьютерные программы* основываются на алгоритме, который ясно определяет последовательную процедуру для решения проблемы. Это может быть математическая формула или последовательная процедура, которая ведет к решению. Алгоритм преобразуется в компьютерную программу, которая только указывает компьютеру какие операции выполнять. Кроме того, для решения проблемы алгоритм использует данные, такие как числа, буквы или слова.

*Интеллектуальная технология* и программное обеспечение ИИ основывается на символическом представлении и манипуляции. В ИИ символ – это буква, слово или число, которые используются для представления объектов процессов и их отношений. Объектами могут быть люди, идеи, понятия, события или утверждения о фактах.

Таблица 1.1.1

Искусственный интеллект по сравнению с традиционным программированием Характеристики	Искусственный интеллект	Традиционное программирование
Обработка	В основном символьная	В основном алгоритмическая
Характер входной	Может быть неполной	Должна быть полной

информации		
Поиск	Эвристический (в большинстве)	Алгоритмический
Объяснение	Обеспечивается	Обычно не обеспечивается
Главный интерес	Знания	Данные, информация
Структура	Управление отделено от знаний	Управление интегрировано с информацией (данными)
Характер выходной информации	Может быть неполной	Должна быть правильной
Сопровождение и модернизация	Легко осуществимы	Обычно затруднительно
Техническое обеспечение	Главным образом рабочие станции и персональные компьютеры	Все типы
Способности к рассуждению	Ограничены, но улучшаются	Нет

При использовании символов возможно создать базу знаний (БЗ), которая содержит факты, понятия и отношения между ними. Используются различные процессы при манипулировании символами для генерации советов или рекомендаций при решении задач.

Каким образом программное обеспечение ИИ осуществляет рассуждения и делает заключения на основе этой БЗ? Базовой техникой являются поиск и сопоставление образцов. Получив первоначальную информацию, программа ИИ просматривает БЗ, осуществляя поиск специфичных условий или образцов. Она ищет подобию и соответствия, которые удовлетворили бы установленному критерию решения задачи.

Несмотря на то, что решение задач при помощи ИИ не реализуется напрямую алгоритмически, алгоритмы используются для осуществления процесса поиска.

Таким образом, ИИ – это в основном уникальный подход к программированию компьютеров, и только он должен рассматриваться. Можно использовать различные технологии, но система ИИ – это компьютерная информационная система, хотя и имеющая некоторые отличительные характеристики (табл. 1.1.1).

Методы искусственного интеллекта очень ценны и значимы. Они помогают показать, как мы думаем, и как лучше использовать наш интеллект. Интеллектуальные технологии и методы ИИ могут сделать компьютеры легче в использовании и сделать доступными большие объемы знаний. Возможно, это даже не имеет значения, что мы не можем полностью дублировать человеческий мозг. Даже когда мы имитируем его функциональные части, результаты могут быть очень полезными.

## 1.2. Искусственный интеллект в проблеме принятия решений

Человек в своей практической деятельности постоянно вынужден принимать решения, причем эти решения и соответствующие им действия должны быть правильными, т.е. приводить к желаемому результату. В основе решений, принимаемых человеком, всегда лежат его опыт и знание. С возникновением математики появилась возможность количественного обоснования решений. Компьютер значительно ускорил этот процесс, и в поддержке принятия решения большую роль начали играть математические

модели, что позволило принимать не просто правильные, а оптимальные решения, в частности, достигать желаемой цели с наименьшими затратами.

Знакомство с курсом математического моделирования в экономике может создать неверное впечатление, что математические методы (математическое программирование и др.) способны решать любые проблемы в принятии экономических решений. На самом деле в такой сфере деятельности, как экономика, где велика роль человека, это далеко не так. Поведение человека не поддается полностью количественному описанию, человек продолжает принимать решения в первую очередь на основе опыта, знаний. Более того, несмотря на существенное развитие экономико-математических методов, решения, принимаемые человеком, зачастую оказываются более правильными, чем рекомендации ЭВМ.

Это объясняется, в частности, тем, что количественные математические модели не в состоянии учесть многих качественных характеристик объекта моделирования, накопленного веками качественного опыта действий в той или иной ситуации. Поэтому решение остается прерогативой человека (и не только в экономике), а создаваемые формализованные методы и системы способны лишь обеспечить помощь, поддержку в принятии этих решений.

Осознание этого факта привело к имитации на математическом языке способности человека анализировать с точки зрения ситуации качества, что позволило добавить к количественным математическим методам и моделям их качественные аналоги и тем самым расширило возможности систем поддержки принятия решений, не выходя за их рамки.

Во второй половине XX в. были предприняты небезуспешные попытки создания систем искусственного интеллекта. Такие системы должны не только уметь обрабатывать комбинации количественной и качественной информации, но и каким-либо образом формализовать знания человека об окружающем мире либо свойственные человеку приемы мышления, посредством которых он изучает окружающий мир и подчиняет его своим интересам.

Интеллектуализация систем поддержки принятия решений началась с математической логики и ее приложений, которые позволили анализировать достаточно сложные ситуации при помощи простейших характеристик – «да» и «нет». Позднее появились нечеткие множества и нечеткая логика, позволившие существенно расширить семантику (смысл) качественных характеристик. Были созданы различные системы моделирования знаний, на их основе разработаны базы знаний и механизмы принятия решений, например, в виде правил «если ситуация такая – то действия должны быть такими». Это, в конечном счете, привело к созданию экспертных систем, имитирующих действия и решения наиболее квалифицированных специалистов – экспертов. В основе экспертных систем лежат знания, получаемые от экспертов, отсюда их синоним – когнитивные системы (лат. *cognitio* – познание) или системы, основанные на знаниях.

Существует другой подход к созданию систем искусственного интеллекта, называемый структурным, к которому относятся нейронные сети. Такие сети имитируют деятельность человеческого мозга, образуя совокупность заимствованных из нейрофизиологии моделей параллельных вычислительных структур.

Третье направление реализуется в рамках *эволюционного моделирования*, к которому можно отнести генетические алгоритмы. В основу эволюционных методов легло заложенное самой природой свойство живых организмов осуществлять правильный выбор, в частности, на основе эволюционного отбора. Генетические алгоритмы являются мощным универсальным средством решения задач глобальной оптимизации, с помощью которого выбираются решения, если не оптимальные, то достаточно близкие к ним.

Отдельное направление составляют системы, которые для решения задач используют сочетания различных методов искусственного интеллекта: экспертных систем, нейронных сетей, генетических алгоритмов и нечеткой логики. Такие сочетания получили название гибридных интеллектуальных систем.

Прежде чем приступить к изучению систем, имитирующих свойства живых организмов разумно рассуждать и делать правильный выбор (систем искусственного интеллекта), весьма полезно понять, зачем они нужны, какие задачи можно будет решать с их помощью и в чем состоит основная идея получения решения. Для этого вначале ознакомимся с идеологией и примерами использования этих систем.

Начнем с уточнения терминологии. Термин «artificial intelligence» (AI) впервые был предложен американскими учеными в 1956 г. В российской научной литературе термин переведен как «искусственный интеллект». В английском языке данное словосочетание не имеет той слегка фантастической окраски, которую оно приобрело в довольно неудачном русском переводе. Слово intelligence означает «умение рассуждать разумно», а вовсе не интеллект, для которого есть термин intellect. Искусственный интеллект обычно трактуется как свойство автоматических систем выполнять отдельные разумные действия, свойственные живым существам, прежде всего человеку. Например, выбирать и принимать правильные решения на основе ранее полученного опыта и (или) рационального анализа внешних воздействий.

### 1.3. Развитие исследований в области искусственного интеллекта

Искусственный интеллект как наука был основан тремя поколениями исследователей.

В таблице 1.1.2 представлены ключевые события в истории ИИ и инженерии знаний, начиная с первой работы Маккалока и Питса в 1943 г. и до современных тенденций в комбинированных усилиях экспертных систем, нечеткой логики и нейронных вычислений в современных системах, основанных на знаниях, способных осуществлять вычисления при помощи слов.

Таблица 1. 1.2

Краткий перечень главных событий в истории ИИ и инженерии знаний Период	События
Рождение ИИ (1943-1956)	<ul style="list-style-type: none"> <li>- Маккалок и Питс: Логическое исчисление идей, присущих нервной деятельности, 1943.</li> <li>- Тьюринг: <i>Вычислительная машина и интеллект</i>, 1950.</li> <li>- Шеннон: <i>Программирование компьютера для шахматной игры</i>, 1950.</li> </ul>

Подъем ИИ (1956- конец 1960-х)	<ul style="list-style-type: none"> <li>- Маккарти: <i>LISP – язык программирования искусственного интеллекта.</i></li> <li>- Куллиан: <i>Семантические сети для представления знаний, 1966.</i></li> <li>- Ньюэл и Саймон: <i>Универсальный решатель задач (GPS), 196</i></li> <li>- Минский: <i>Структуры для представления знаний (фреймы), 1975.</i></li> </ul>
Открытие и разработка экспертных систем (начало 1970-х – середина 1980-х).	<ul style="list-style-type: none"> <li>- Фейгенбаум, Буханан и др. (Стэндфордский университет): <i>Экспертная система DENDRAL</i></li> <li>- Фейгенбаум, Шортлиф: <i>Экспертная система MYCIN</i></li> <li>- Стэндфордский исследовательский центр: <i>Экспертная система PROSPECTOR</i></li> <li>- Колмероз, Ковальски и др. (Франция): <i>Язык логического программирования PROLOG.</i></li> </ul>
Возрождение искусственный нейронных сетей (1965 и далее)	<ul style="list-style-type: none"> <li>- Хопфилд: <i>Нейронные сети и физические с эмергентными коллективными вычислительными способностями, 1982.</i></li> <li>- Кохонен: <i>Самоорганизующиеся топологически правильные карты, 1982.</i></li> <li>- Румельхарт и Макклеланд: <i>Распределенная параллельная обработка данных, 1986.</i></li> </ul>
Эволюционное вычисление (начало 1970-х и далее)	<ul style="list-style-type: none"> <li>- Рехенберг: <i>Эволюционные стратегии – оптимизация технических систем по принципам биологической информации, 1973.</i></li> <li>- Холланд: <i>Адаптация в естественных и искусственных системах, 1975.</i></li> <li>- Коза: <i>Генетическое программирование: компьютерное программирование средствами естественного отбора, 1992.</i></li> <li>- Фогель: <i>Эволюционное вычисление – направление новой философии в машинном интеллекте, 1995.</i></li> </ul>
Нечеткие множества и нечеткая логика (середина 1960-х и далее)	<ul style="list-style-type: none"> <li>- Заде: <i>Нечеткие множества, 1965.</i></li> <li>- Заде: <i>Нечеткие алгоритмы, 1969.</i></li> <li>- Мамдани: <i>Применение нечеткой логики в приближенном рассуждении с использованием лингвистического синтеза, 1977.</i></li> </ul>
Вычисления при помощи слов (конец 1980-х и далее)	<ul style="list-style-type: none"> <li>- Нейгоца: <i>Экспертные системы и нечеткие системы, 1985.</i></li> <li>- Коско: <i>Нейронные сети и нечеткие системы, 1992.</i></li> </ul>

	<ul style="list-style-type: none"> <li>- Коско: <i>Нечеткое мышление</i>, 1993.</li> <li>- Ягер и Заде: <i>четкие множества, нейронные сети и мягкие вычисления</i>, 1994.</li> <li>- Коско: <i>Нечеткая инженерия</i>, 1996.</li> <li>- Заде: <i>Вычисления при помощи слов</i>, 1996.</li> </ul>
--	--

#### 1.4. Теория и практика искусственного интеллекта.

В процессе развития устройств и систем, которые проявляют интеллектуальные характеристики, вовлекаются различные науки и технологии, такие как лингвистика, психология, философия, техническое и программное обеспечение компьютеров, механика, гидравлика и оптика.

Пересечение интересов психологии и ИИ сосредоточено в областях когнитологии и психолингвистики. Философия и ИИ сотрудничают в областях логики, философии языка и философии разума. Взаимные пересечения между инженерией и ИИ включают обработку изображений, распознавание образов и робототехнику.

Позже свой вклад внесли менеджмент и теория организации (такие как принятие и реализация решений), химия, физика, статистика, математика, теория управления, эвристическое программирование, информационные системы менеджмента.

ИИ является наукой и технологией, а не коммерческой сферой. Это совокупность понятий и идей, которые предназначены для исследований. Однако, ИИ обеспечивает научены основы для нескольких развивающихся коммерческих технологий. Главными прикладными технологиями ИИ экспертные системы, интеллектуальные системы поддержки принятия решений, обработка естественного языка, понимание речи, нечеткая логика, робототехника и сенсорные системы, компьютерное зрение и распознавание образов. На рис. 1.1.1 представлены главные дисциплины и приложения ИИ.

Области применения существующих на сегодняшний день систем ИИ охватывает медицинскую диагностику, интерпретацию геологических данных, научные исследования в химии и биологии, военное дело, производство, финансы и другие сферы экономики. Однако, несмотря на значительные успехи в области ИИ, пока еще существует определенный разрыв между техническими разработками, программными средствами ИИ и возможностями их более широко практического применения в частности, в экономике.

Наиболее показательным сектором, аккумулирующим различные проблемные направления экономической области, является управление промышленным предприятием. На его примере особенно хорошо видны преимущества использования систем ИИ для решения как различных предметных задач, так и для управления интегрированной системой предприятия в целом.

Существует множество доводов в пользу того, что системы искусственного интеллекта могут и должны стать важнейшей составной частью в технологии современных систем ERP. Рассмотрим основные из них.

Главная проблема, стоящая перед предприятием (в смысле управления), - это проблема преодоления сложности. Как известно, сложности управления возникают тогда,



когда приходится делать выбор из множества возможных решений. Это может быть инженерный выбор решения (как проектировать данное изделие), выбор расписания (как это изделие производить) и т. д.

Данная проблема обостряется в случае гибких систем планирования материальных потребностей и систем планирования производственных ресурсов. Добавление гибкости приводит к увеличению числа альтернатив и, следовательно, возможных вариантов производства изделий. Уже сегодня составление расписания этапов производства изделий на «жестком» программном обеспечении представляет большие трудности. Усложнение самих изделий также ведет к усложнению проектирования.

Управление производством требует обработки большого объема информации. Проблема получения информации с объектов, функционирующих в реальном масштабе времени, в настоящее время решена. Но это породило другую проблему: как уменьшить долю информации до того уровня, который действительно необходим для принятия решения индивидуумом? В то же время следует отметить, что потеря информации, поступающей от объектов, работающих в реальном масштабе времени, может существенно сказаться на конечном результате.

Нехватка времени на принятие решения – еще одна проблема, которая проявляется по мере усложнения производства. Не менее важна и проблема координации. Известно, что проектирование неразрывно связано с производством, распределением и вспомогательными областями. Если проектирование не оптимально по отношению к стадиям производства, складирования, распределения или вспомогательному производству, то это может увеличить цену производства и снизить качество изделий.

И наконец, очень важный фактор – необходимость сохранения и распределения знаний отдельных опытных экспертов, полученных, ими в процессе многолетней работы и большого практического опыта. Проблема извлечения знаний и их распределения – сегодня одна из главных проблем производственных организаций.

Таким образом, происходит интеллектуализация информационных систем управления и трансформация их в интеллектуальные системы поддержки принятия решений (СППР), основной разновидностью которых являются ЭС. Это наиболее значимые и важные для экономики и бизнеса прикладные технологии ИИ.

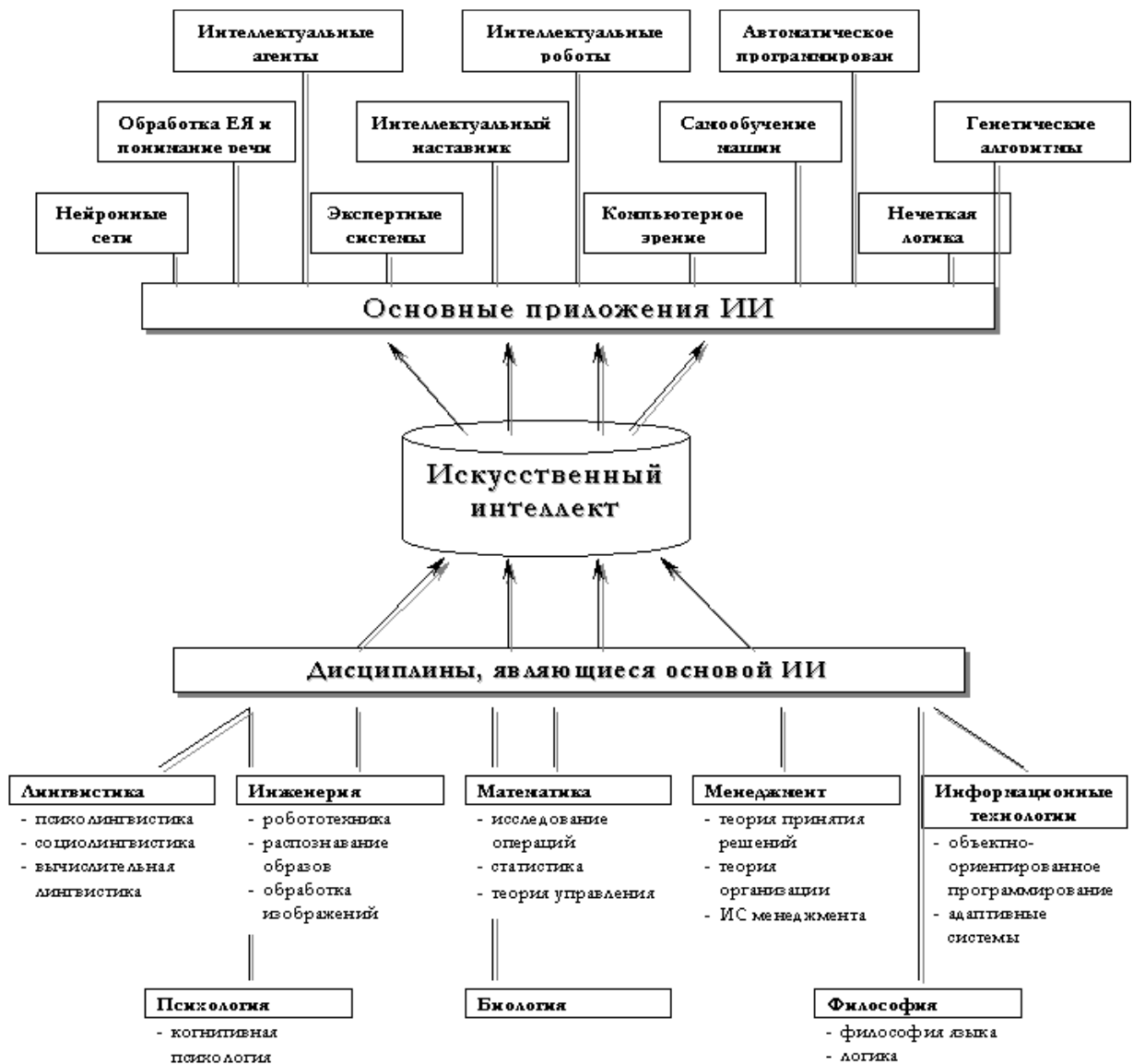


Рисунок 1.1.1 - Дисциплины, на которых базируется ИИ

Рисунок 1.1.1 - Дисциплины, на которых базируется ИИ и его приложения

## 2. ПОДХОДЫ К СОЗДАНИЮ СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

### 2.1. Экспертные системы

Рассмотрим наиболее распространенные в настоящий момент направления развития интеллектуальных систем и примеры их практического применения в экономике. 19

понятной специалистам предметной области, и могут быть изменены и дополнены также в понятной форме. Это и есть языки представления знаний (ЯПЗ).

В России в исследования и разработку ЭС большой вклад внесли работы Д. А. Поспелова (основателя Российской ассоциации искусственного интеллекта и его первого президента), Э. В. Попова, В. Ф. Хорошевского, В. Л. Стефанюка, Г. С. Осипова, В. К. Финна, В. Л. Вагина, В. И. Городецкого и многих других. Современное состояние разработок в области ЭС в России можно охарактеризовать как стадию все возрастающего интереса среди широких слоев специалистов - менеджеров, инженеров, программистов и других. Наибольшие трудности в разработке ЭС вызывает не процесс машинной реализации систем, а этап анализа знаний и проектирования базы знаний. Этим занимается специальная наука - инженерия знаний.

Экспертные системы имеют две категории пользователей и два отдельных входа, соответствующих различным целям взаимодействия пользователей с ЭС. К первой категории относятся обычные пользователи, которым требуется консультация ЭС. Вторую категорию представляют эксперты в предметной области и инженеры знаний. В их функции входит заполнение базы знаний с помощью специализированной диалоговой компоненты ЭС - подсистемы приобретения знаний. Подсистема приобретения знаний предназначена для добавления в базу знаний новых правил и модификации имеющихся. В ее задачу входит приведение правила к виду, позволяющему подсистеме вывода применять это правило в процессе работы. В более сложных системах предусмотрены еще и средства для проверки вводимых или модифицируемых правил на непротиворечивость с имеющимися правилами. Диалог с ЭС осуществляется через диалоговый процессор - специальную компоненту ЭС. Существуют две основные формы диалога с ЭС - это диалог на ограниченном подмножестве естественного языка с использованием словаря (меню) и диалог на основе из нескольких возможных действий. База знаний представляет наиболее важную компоненту экспертной системы. В отличие от всех остальных компонент ЭС, база знаний – есть «переменная» часть системы, которая может пополняться и модифицироваться инженерами знаний и опыта использования ЭС между консультациями, а в некоторых системах и в процессе консультации. Существует несколько способов представления знаний в ЭС. Общим для всех способов является то, что знания представлены в символьной форме (тексты, списки и другие символьные структуры). Тем самым, в ЭС реализуется принцип символьной природы рассуждений, который заключается в том, что процесс рассуждения представляется как последовательность символьных преобразований. Подсистема вывода - программная компонента экспертных систем, реализующая процесс ее рассуждений на основе базы знаний и рабочего множества. Она выполняет две функции: во-первых, просмотр существующих фактов из рабочего множества и правил из базы знаний и добавление (по мере возможности) в рабочее множество новых фактов и, во-вторых, определение порядка просмотра и применения правил. Эта подсистема управляет процессом консультации, сохраняет для пользователя информацию о полученных заключениях, и запрашивает у него информацию, когда для срабатывания очередного правила в рабочем множестве оказывается недостаточно данных. Цель ЭС - вывести некоторый заданный факт, который называется целевым утверждением. В результате применения правил добиться того, чтобы этот факт был включен в рабочее множество, либо опровергнуть этот факт, то есть убедиться, что его вывести невозможно. Целевое утверждение может быть либо «заложено» заранее в базу знаний системы, либо извлекается системой из диалога с пользователем. Работа системы представляет собой последовательность шагов, на каждом из которых из базы выбирается некоторое правило, которое применяется к текущему содержимому рабочего множества. Цикл заканчивается, когда выведено либо опровергнуто целевое утверждение. Цикл работы экспертной системы иначе называется логическим выводом.

Одной из первых попыток создания системы искусственного интеллекта было моделирование накопленного опыта в узкопрофессиональной предметной области. Опыт отображался в виде совокупности знаний, а вырабатываемое системой решение основывалось на методах анализа этих знаний. Такие системы

получили название экспертных, поскольку в их формировании принимают участие наиболее знающие и опытные в данной предметной области специалисты – эксперты. Экспертные системы, аккумулирующие знания множества специалистов – экспертов, в принципе способны вырабатывать наилучшие решения по сравнению с привлеченным к решению задачи отдельным экспертом. Проблема заключалась и продолжает заключаться лишь в одном: как наиболее адекватно отображать знания эксперта в памяти ЭВМ, правильно анализировать эти знания и получать новые знания на основе анализа предыдущих?

Сегодняшнее состояние развития экспертных систем показывает, что специалисты в этой области достаточно хорошо продвинулись в решении данной проблемы. Экспертные системы нашли применение в самых разнообразных областях деятельности: управлении, медицине, технике, химии, физике, экономике и многих других.

Принято выделять экспертные системы первого поколения, действующие на основе некоторых знаний о предметной области, представленных, например, правилами типа «если – то». Составлением таких правил для машины занимается человек, который, в свою очередь, в процессе этой работы также использует некоторые правила составления правил. Такие правила принято называть «метаправилами» или «метазнаниями», т.е. знаниями о знаниях. Формализация метазнаний приводит к появлению экспертных систем второго поколения. При этом надо понимать, что составлением метазнаний опять-таки занимается человек в соответствии с некоторыми правилами еще более высокого уровня. Таким образом, формализация правил более высокого уровня приводит к экспертным системам старшего поколения, но определяющая роль человека остается всегда, при любом поколении экспертных систем.

В экономике экспертные системы нашли применение при решении задач, прежде всего в области планирования и управления.

Экспертные системы – это быстро прогрессирующее направление в области искусственного интеллекта. Современные ЭС представляют собой сложные программные комплексы, аккумулирующие знания специалистов в конкретных предметных областях и распространяющие этот эмпирический опыт для консультирования менее квалифицированных пользователей. Парадигма ЭС предполагает следующие объекты, а также этапы разработки и функционирования информационных систем (ИС):

## **2.2. Этапы развития научного направления «мягкие вычисления»**

На рубеже тысячелетий для каждого научного направления важны результаты предшествующих исследований и перспективы предстоящих. В настоящее время согласно мнению широкой научной общественности искусственный интеллект не оправдал ожиданий – с помощью символической обработки информации невозможно решить прикладные задачи разнообразных предметных областей, так как приложения насыщены расчетами. Применение искусственного интеллекта было успешным в основном в тех приложениях, где сочетались символическая обработка информации и вычисления. С помощью вычислений определялись вспомогательные функции в интеллектуальных задачах, обеспечивалась эффективность приложений, но их использование

считалось «неправильным» для искусственного интеллекта, несмотря на то, что именно вычислительные модели были первыми интеллектуальными моделями, полученными в рамках бионического направления.

И все же традиционные компьютерные вычисления «слишком точны» для реального мира. Человечество столкнулось с проблемами, для решения которых невозможно получить полную информацию или определение которых недостаточно полно. Такая ситуация характерна для сложных технических систем, систем экономического планирования, систем жизни, социальных систем большой размерности и систем принятия решений. Казалось бы, положение безвыходное, но уже со второй половины XX в. благодаря развитию и совершенствованию так называемых нечетких и гибридных систем, теоретический фундамент которых и представлен в данном пособии, невероятное стало реальностью. В настоящее время уже довольно обыденно воспринимаются «сверхинтеллектуальные» стиральные машины и бытовые автоматы в повседневной жизни, гиперзвуковые самолеты и самонаводящиеся ракеты в оборонных подразделениях армий и многое, многое другое.

Математическую основу рассматриваемых систем составляют противоположные традиционным компьютерным вычислениям (hard computing) мягкие вычисления (soft computing) или сложная компьютерная методология, компонентами которой являются:

- нечеткая логика – приближенные вычисления, грануляция информации, вычисление на словах;
- нейронные сети – обучение, адаптация, классификация, системное моделирование и идентификация;
- генетические алгоритмы – синтез, настройка и оптимизация с помощью систематизированного случайного поиска и эволюции;
- вероятностные вычисления – управление неопределенностью, сети доверия, хаотические системы, предсказание.

Перечисленные составные части не конкурируют между собой, а создают эффект взаимного усиления для достижения низкой цены решения и большего соответствия реальности.

Для формализации предметной области нечетких и гибридных систем наряду с термином «мягкие вычисления» используют и другой интегрирующий термин – «вычислительный интеллект». В общем случае вычислительный интеллект – это целое научное направление, где решаются задачи искусственного интеллекта на основе теории нечетких систем, нейронных сетей и эволюционных вычислений.

Термин (от англ. umbrella term) «мягкие вычисления», введенный еще в 1994 г. американским ученым Л. Заде, интерпретируется следующей формулой:

**Мягкие вычисления = нечеткие системы + нейронные сети + генетические алгоритмы**

В обоснование формулы мягких вычислений приводятся свойства мягкой интеллектуальной системы, которые обеспечиваются ее компонентами:

**Мягкая интеллектуальная система = управление неопределенностью + обучаемость + самоадаптация**

Выбранная терминология объясняется тем, что мягкие системы, такие как нечеткие нейронные сети с генетической настройкой параметров,

демонстрируют взаимное усиление достоинств и нивелирование недостатков отдельных методов. Очевидно, что представление знаний в нейронных сетях в виде весовой матрицы не позволяет объяснить результаты проведенного распознавания или прогнозирования, в то время как в системах вывода на базе нечетких правил результаты воспринимаются как обратные протоколы вывода (ответы на вопрос: «Почему?»). Нейронные сети обучаются с помощью универсального алгоритма, т. е. трудоемкое извлечение знаний заменяется сбором достаточной по объему обучающей выборки.

Для нечетких систем вывода извлечение знаний включает в себя сложные процессы формализации понятий, определение функций принадлежности, формирование правил вывода. В то же время нечеткие нейронные сети обучаются как нейронные сети, но их результаты объясняются как в системах нечеткого вывода.

Увлечение символьной обработкой в течение 1970-х, 1980-х годов оставило почти незамеченным рождение и становление нового научного направления – вычислительного интеллекта, решающего задачи искусственного интеллекта на основе теории нечетких систем, нейронных сетей и эволюционного моделирования.

В 1990-е годы были также разработаны различные варианты гибридных интеллектуальных систем.

Мягкие вычисления позволили объединить в единую технологию управление неопределенностью, обучение и адаптацию. Сегодня созданы и используются в промышленности нейро-нечеткие контроллеры, разнообразные интеллектуальные системы. Пик теоретических изысканий в области теории нечетких систем приходится на создание вычислений на словах, основанных на перцептивных оценках, что обещает прояснить сложнейшие проблемы семантики естественного языка.

### **Развитие теории нечетких систем.**

Публикация статьи профессора Л. Заде «Нечеткие множества» в 1965 г. положила начало развитию теории нечетких систем в результате определения нового класса нечетких множеств, т. е. множеств с неточно определенными границами, описываемыми функциями принадлежности элементов. Несмотря на то что научная общественность встретила новую идею настороженно, за публикацией последовал поток научных исследований. В 1970-е годы прошлого века были развиты понятия лингвистической переменной, Е. Мамдани сформулировал основные идеи нечетких регуляторов. В 1978 г. Л. Заде предложил вариант исчисления неопределенностей, опирающийся на неаддитивную меру возможности, т. е. на интерпретацию нечеткого множества как функции распределения возможностей. В 1979 г. он же ввел теорию приближенных рассуждений.

Своевременно начатое практическое использование новой теории подтвердило правоту высказанных идей и породило поток прикладных исследований. В 1983 г. японская фирма «Фуджи Электрик» реализовала на основе нечетких алгоритмов установку для обработки питьевой воды. В 1987 г. была запущена в производство система управления новым метро в г. Сендай, близ Токио, предложенная фирмой «Хитачи». Сегодня ведущие компании США, Германии, Франции и других стран предлагают разнообразные товары и системы с использованием функций нечеткой логики (fuzzy logic). В

телевизионной рекламе бытовой техники наличие подобной функциональности отмечается в качестве признака самой современной на сегодняшний день системы управления устройством.

### **Развитие нейронных сетей.**

Исследования восприятия на искусственных моделях нейронных сетей успешно проводились уже с 1940-х годов прошлого века У. МакКаллоком, У. Питтсом (1943) и Д. Хеббом (1949).

Далее, на базе разработанных в 1982 г. Дж. Хопфилдом математических основ динамики нейронных сетей, созданной в 1984 г. Т. Кохоненом методике их обучения и предложенном в 1986 г. Д. Румельхартом алгоритме обратного распространения ошибки начались прикладные исследования нейронных сетей.

Искусственные нейронные сети можно рассматривать в качестве простых математических моделей мозгоподобных систем, функционирующих как параллельные распределенные вычислительные сети. Однако в противоположность компьютерам с архитектурой Дж. фон Неймана для решения специфических задач нейронные сети должны быть обучены или натренированы новым ассоциациям, функциональным зависимостям и образцам для распознавания. В отличие от компьютеров, основанных на точных и быстрых арифметических операциях, биологические и искусственные нейронные сети обещают решение сложных частично структурированных и неструктурированных проблем.

Изучение мозгоподобных структур велось в рамках создания теории интеллектуальных систем; начало его было ознаменовано более 50 лет назад публикацией все тех же работ У. Мак-Каллока и У. Питтса (1943) и продолжено позднее в знаменитой работе Д. Хеббса «Организация поведения» (1949). Ранние работы по искусственному интеллекту принадлежали перу тех, кто описывал структуру интеллектуальной системы подобной мозгу, и тех (М. Минский, С. Пайперт), кто считал интеллект фундаментальным свойством символьной обработки данных на компьютерах фон-неймановского типа. По многим причинам в 1970 г. второй подход стал доминирующим, но с 1980-х годов сменился так называемым нейронным бумом в результате резко возросшего интереса к нейронным вычислениям.

Применение алгоритма обучения линейного однослойного перцептрона, предложенного Ф. Розенблатом, показало, что нейроподобные вычисления (brain-style computation) эффективны для задач восприятия и распознавания, однако в результате убийственной критики М. Минского и С. Пайперта (1969) сложилось мнение, что искусственный интеллект следует создавать на основе символьной обработки на компьютерах архитектуры Дж. фон Неймана. (Перцептроном называют математическую модель фрагмента нейронной системы головного мозга, осуществляющего восприятие внешнего объекта. Первые перцептроны создавались как модель искусственного глаза. В настоящее время перцептрон представляет собой многослойную нейронную сеть прямого распространения сигнала, способную к обучению).

Однако уже в 1982 г. Дж. Хопфилд изложил математические основы динамики нейронных сетей с обратными связями – нейронная сеть Хопфилда. В 1984 г. Т. Кохонен предложил алгоритм обучения «без учителя» для самоорганизующихся карт. В 1986 г. Д. Румельхарт вывел алгоритм обратного распространения ошибки для нелинейных многослойных нейронных сетей.

## **Эволюционные вычисления.**

Эволюционные вычисления (ЭВ) – это обобщающий термин, используемый для описания компьютерных систем на основе вычислительных моделей эволюционных процессов в качестве базовых при разработке и эксплуатации. В основе моделирования лежит концепция подражания эволюционным процессам отбора (селекции), мутации и размножения (репродукции), которые определяются поведением индивидуумов в изменяющихся условиях окружающей среды. Эволюционные вычисления создают популяции структур, которые развиваются в соответствии с поисковыми операторами, называемыми правилами отбора, и в соответствии с генетическими операторами, представленными операциями рекомбинации и мутации.

Каждый индивидуум в популяции занимает свое место согласно мере его приспособленности условиям среды. Посредством рекомбинации и мутации обеспечивается основной набор для дальнейшего воспроизводства индивидуумов. С точки зрения биологии указанные алгоритмы упрощены, но достаточно сложны для реализации и представляют собой поисковые механизмы адаптации.

История эволюционных вычислений началась с разработки ряда независимых моделей, в частности генетических алгоритмов и классификационных систем Я. Холланда, представленных в начале 1960-х годов и получивших всеобщее признание после выхода ставшей впоследствии классической в своей области книги «Адаптация в естественных и искусственных системах» («Adaptation in Natural and Artificial Systems», 1975). В 1970-х годах в рамках теории случайного поиска Л. А. Растргиным был предложен ряд алгоритмов, использующих идеи бионического поведения особей. Развитие этих идей нашло отражение в работах по эволюционному моделированию И. Л. Букатовой. Развивая идеи М. Л. Цетлина о целесообразном и оптимальном поведении стохастических автоматов, Ю. И. Неймарк предложил осуществлять поиск глобального экстремума на основе коллектива независимых автоматов, моделирующих процессы развития и элиминации особей. Несмотря на разницу в подходах, каждая из одноименных «школ» взяла за основу ряд существующих в природе принципов и упростила до такой степени, чтобы их можно было реализовать на компьютере.

В 1975 г. книга Я. Холланда «Адаптация в природе и искусственных системах» заложила основы эволюционного моделирования, в результате которого генетические алгоритмы стали методом структурного синтеза интеллектуальных систем.

Усилия, направленные на моделирование эволюции по аналогии с природными системами, в настоящее время можно разбить на две категории. К первой относятся алгоритмы моделирования биологических систем, которые успешно используются для задач структурной и функциональной оптимизации. Ко второй категории относятся более реалистичные биологически, но не имеющие пока прикладного значения алгоритмы. Конечно, на практике нет такого строгого деления по категориям, они лишь играют роль своеобразных полюсов, между которыми лежат различные вычислительные системы. Ближе к первому – эволюционные алгоритмы, такие как эволюционное программирование (Evolutionary Programming), генетические алгоритмы (Genetic



Algorithms) и эволюционные стратегии (Evolution Strategies). Второй полюс представляют системы, которые классифицируются как искусственная жизнь (Artificial Life).

### **Парадигма вычислительного интеллекта.**

Большинство интеллектуальных технологий являются анализирующими или диагностирующими. Интеллектуальные методы, синтезирующие новые изделия, редки, поэтому для решения задач структурного и функционального синтеза столь велико значение генетических алгоритмов, с развитием которых может быть создана действительно интеллектуальная система автоматизации проектирования вместо многочисленных САПР, использующих экспертную систему на ранних стадиях разработки. С помощью генетических алгоритмов системы стохастической оптимизации успешно ищут профиль детали, размещают элементы на плоскости и в пространстве, компонуют и трассируют реальные технические объекты. В гибридных системах благодаря генетическим алгоритмам (ГА) производят оптимизационную настройку функций принадлежности нечетких множеств, которые задаются параметризованной функцией формы. Применяя ГА, оптимизируют и состав больших баз нечетких продукций, и структуру нейронной сети для генетических нечетких нейронных сетей.

Объединение методологий экспертных систем, теории нечетких множеств, нейронных сетей, генетических алгоритмов и других методов привело к созданию перспективного научного направления – *гибридных интеллектуальных систем*. В качестве предпосылки такого объединения выступает, прежде всего, тот факт, что множество существующих проблем интеллектуальных систем нельзя решить в рамках лишь одного подхода, наилучшие результаты могут быть достигнуты только путем комбинации вышеперечисленных составляющих.

Одним из наиболее распространенных примеров гибридных систем является обучение нейронных сетей при помощи генетических алгоритмов. При этом обеспечивается высокая скорость подстройки весов сети, достаточно малые ошибки обучения. Применение экспертных систем также позволяет значительно ускорить обучение нейронной сети. При разработке нечетких экспертных систем нейронные сети могут быть использованы для настройки параметров функции принадлежности. При этом нейронные сети обеспечивают настройку параметров на основе автоматического извлечения закономерностей из обучающей выборки.

Таким образом, сочетание в гибридных технологиях элементов различных подходов позволяет решить множество проблем, ограничивающих использование интеллектуальных систем в экономических и других приложениях. Гибридные методы на сегодня считаются самым востребованным инструментом искусственного интеллекта.

Причины формирования гибридных систем, которые составляют основное содержание вычислительного интеллекта, носят технологический прикладной характер. Более фундаментальной причиной развития вычислительного интеллекта, на наш взгляд, является необходимость интеграции в единой системе восприятия и логической обработки. Исторически исследования восприятия (нейронные сети, распознавание образов) ведутся отдельно от изучения классического искусственного интеллекта, моделирующего логическое мышление. Существует разрыв между бионическим интеллектом искусственных нейронных сетей и интеллектом систем логического вывода, в то время как

естественный интеллект не имеет такой резкой границы. Грубую схему «слоев естественного интеллекта» можно описать следующей формулой:

**Сенсорика + моторика + безусловные и условные рефлексy + врожденные программы (инстинкты) + мышление = интеллект**

Возможно, эффективность человеческого интеллекта состоит в том, что указанные слои работают согласованно, помогая тем самым решать многие задачи, недоступные искусственным системам. Вычислительный интеллект объединяет искусственные модели перечисленных компонент-слоев в гибридные системы, что позволяет рассматривать его не как технологическое достижение, а как парадигму развития искусственного интеллекта XXI в.

### **3. ОСНОВНЫЕ НАПРАВЛЕНИЯ, ФУНКЦИИ И КЛАССИФИКАЦИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ**

#### **3.1. Области применения ИИС**

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

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

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

3. Генерация и распознавание речи. Системы речевого общения создаются в целях повышения скорости ввода информации в ЭВМ, разгрузки зрения и рук, а также для реализации речевого общения на значительном расстоянии.

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

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

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

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

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

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

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

10. Интеллектуальные роботы. Создание интеллектуальных роботов составляет конечную цель робототехники. В настоящее время в основном

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

### **3.2. Признаки классификации интеллектуальных информационных систем**

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

Для ИИС характерны следующие признаки:

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

Каждому из перечисленных признаков условно соответствует свой класс интеллектуальных информационных систем (ИИС). Различные системы могут обладать одним или несколькими признаками интеллектуальности с различной степенью проявления. Среди них можно выделить следующие типы:

1. Интеллектуальные базы данных позволяют в отличие от традиционных баз данных (БД) обеспечивать выборку необходимой информации, не присутствующей в явном виде, а выводимой из совокупности хранимых данных.

2. Естественнo-языковой (ЕЯ) интерфейс применяется для доступа к интеллектуальным базам данных, контекстного поиска документальной текстовой информации, голосового ввода команд в системах управления, машинного перевода с иностранных языков. Для реализации ЕЯ - интерфейса необходимо решить проблемы морфологического, синтаксического и семантического анализа, а также задачу синтеза высказываний на естественном языке. При морфологическом анализе осуществляются распознавание и проверка правильности написания слов в словаре. Синтаксический контроль предполагает разложение входных сообщений на отдельные компоненты, проверку соответствия грамматическим правилам внутреннего представления знаний и выявление недостающих частей. Семантический анализ обеспечивает установление смысловой правильности синтаксических конструкций. В отличие от анализа синтез высказываний заключается в преобразовании цифрового представления информации в представление на естественном языке.

3. Гипертекстовые системы используются для реализации поиска по ключевым словам в базах данных с текстовой информацией. Для более полного отражения различных смысловых отношений терминов требуется сложная семантическая организация ключевых слов. Решение этих задач осуществляется с помощью интеллектуальных гипертекстовых систем, в которых механизм поиска сначала работает с базой знаний ключевых слов, а затем - с самим текстом. Аналогичным образом проводится поиск мультимедийной информации, включающей кроме текста графическую информацию, аудио - и видео образы.

4. Системы контекстной помощи относятся к классу систем распространения знаний. Такие системы являются, как правило, приложениями к документации. Системы контекстной помощи - частный случай гипертекстовых и ЕЯ-систем. В них пользователь описывает проблему, а система на основе дополнительного диалога конкретизирует ее и выполняет поиск относящихся к ситуации рекомендаций. В обычных гипертекстовых системах, наоборот, компьютерные приложения навязывают пользователю схему поиска требуемой информации.

5. Системы когнитивной графики ориентированы на общение с пользователем ИИС посредством графических образов, которые генерируются в соответствии с изменениями параметров моделируемых или наблюдаемых процессов. Когнитивная графика позволяет в наглядном и выразительном виде представить множество параметров, характеризующих изучаемое явление, освобождает пользователя от анализа тривиальных ситуаций, способствует быстрому освоению программных средств и повышению конкурентоспособности разрабатываемых ИИС. Применение когнитивной графики особенно актуально в системах мониторинга и оперативного управления, в обучающих и тренажерных системах, в оперативных системах поддержки принятия решений (СППР), работающих в режиме реального времени.

### **3.3. Понятие экспертной системы**

Экспертные системы как самостоятельное направление в искусственном интеллекте сформировалось в конце 1970-х гг. Группа по экспертным системам при Комитете British Computer Society определила ЭС как «воплощение в ЭВМ компоненты опыта эксперта, основанной на знаниях, в такой форме, что машина может дать интеллектуальный совет или принять решение относительно обрабатываемой функции». Одним из важных свойств ЭС является способность объяснить ход своих рассуждений понятным для пользователя образом.

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

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

Главное отличие ЭС и систем искусственного интеллекта от систем обработки данных состоит в том, что в них используется символьный, а не числовой способ представления данных, а в качестве методов обработки

информации применяются процедуры логического вывода и эвристического поиска решений.

ЭС охватывают самые разные предметные области, среди которых лидируют бизнес, производство, медицина, проектирование и системы управления. Во многих случаях ЭС являются инструментом, усиливающим интеллектуальные способности эксперта. Кроме того, ЭС может выступать в роли:

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

Для классификации ЭС используются следующие признаки:

- способ формирования решения;
- способ учета временного признака;
- вид используемых данных и знаний;
- число используемых источников знаний.
- партнера эксперта в процессе решения задач, требующих привлечения знаний из разных предметных областей.

Для классификации ЭС используются следующие признаки:

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

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

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

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

ЭС могут создаваться с использованием одного или нескольких источников знаний.

В соответствии с перечисленными признаками можно выделить четыре основных класса ЭС: классифицирующие, доопределяющие, трансформирующие и мультиагентные.

Классифицирующие ЭС решают задачи распознавания ситуаций. Основным методом формирования решений в таких системах является дедуктивный логический вывод.

Доопределяющие ЭС используются для решения задач с не полностью определенными данными и знаниями. В таких ЭС возникают задачи интерпретации нечетких знаний и выбора альтернативных направлений поиска в



пространстве возможных решений. В качестве методов обработки неопределенных знаний могут использоваться байесовский вероятностный подход, коэффициенты уверенности, нечеткая логика.

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

- генерация и проверка гипотез;
- логика предположений и умолчаний (когда по неполным данным формируются представления об объектах определенного класса, которые впоследствии адаптируются к конкретным условиям изменяющихся ситуаций);
- использование метазнаний, более общих закономерностей для устранения неопределенностей в ситуациях.

Мультиагентные системы - это динамические ЭС, основанные на интеграции нескольких разнородных источников знаний. Эти источники обмениваются между собой получаемыми результатами в ходе решения задач. Системы данного класса имеют следующие возможности:

- реализация альтернативных рассуждений на основе использования различных источников знаний и механизма устранения противоречий;
- распределенное решение проблем, декомпозируемых на параллельно решаемые подзадачи с самостоятельными источниками знаний;
- применение различных стратегий вывода заключений в зависимости от типа решаемой проблемы;
- обработка больших массивов информации из баз данных;
- использование математических моделей и внешних процедур для имитации развития ситуаций.

### **3.4. Самообучающиеся системы**

Самообучающиеся интеллектуальные системы основаны на методах автоматической классификации ситуаций из реальной практики, или на методах обучения на примерах. Примеры реальных ситуаций составляют так называемую обучающую выборку, которая формируется в течение определенного исторического периода. Элементы обучающей выборки описываются множеством классификационных признаков. Стратегия «обучения с учителем» предполагает задание специалистом для каждого примера значений признаков, показывающих его принадлежность к определенному классу ситуаций. При обучении «без учителя» система должна самостоятельно выделять классы ситуаций по степени близости значений классификационных признаков. В процессе обучения проводится автоматическое построение обобщающих правил или функций, описывающих принадлежность ситуаций к классам, которыми система впоследствии будет пользоваться при интерпретации незнакомых ситуаций. Из обобщающих правил, в свою очередь, автоматически формируется база знаний, которая периодически корректируется по мере накопления информации об анализируемых ситуациях.

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

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

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

1. Выбор классификационного признака из заданного множества.
2. Разбиение множества примеров на подмножества по значению выбранного признака.
3. Проверка принадлежности каждого подмножества примеров одному из классов.
4. Проверка окончания процесса классификации. Если какое-то подмножество примеров принадлежит одному подклассу, то есть у всех примеров этого подмножества совпадает значение классификационного признака, то процесс классификации заканчивается.
5. Для подмножеств примеров с несовпадающими значениями классификационных признаков процесс распознавания продолжается, начиная с первого шага. При этом каждое подмножество примеров становится классифицируемым множеством.

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

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

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

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



- занесение детальной информации о полученном решении в БЗ.

Прецеденты описываются множеством признаков, по которым строятся индексы быстрого поиска. Однако в системах, основанных на прецедентах, в отличие от индуктивных систем допускается нечеткий поиск с получением множества допустимых альтернатив, каждая из которых оценивается некоторым коэффициентом уверенности. Наиболее эффективные решения адаптируются к реальным ситуациям с помощью специальных алгоритмов. Системы, основанные на прецедентах, применяются для распространения знаний и в системах контекстной помощи.

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

Технологии извлечения знаний из хранилищ данных основаны на методах статистического анализа и моделирования, ориентированных на поиск моделей и отношений, скрытых в совокупности данных. Для извлечения значимой информации из хранилищ данных имеются специальные методы (OLAP-анализа, Data Mining или Knowledge Discovery), основанные на применении методов математической статистики, нейронных сетей, индуктивных методов построения деревьев решений и других. Технология OLAP (On-Line Analytical Processing - оперативный анализ данных) предоставляет пользователю средства для формирования и проверки гипотез о свойствах данных или отношениях между ними на основе разнообразных запросов к базе данных. Они применяются на ранних стадиях процесса извлечения знаний, помогая аналитику сфокусировать внимание на важных переменных. Средства Data Mining отличаются от OLAP тем, что кроме проверки предполагаемых зависимостей они способны самостоятельно (без участия пользователя) генерировать гипотезы о закономерностях, существующих в данных, и строить модели, позволяющие количественно оценить степень взаимного влияния исследуемых факторов на основе имеющейся информации.

### **3.5. Адаптивные информационные системы.**

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

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

Адаптивные свойства информационных систем обеспечиваются за счет интеллектуализации их архитектуры. Ядром таких систем является постоянно развиваемая модель проблемной области, поддерживаемая в специальной базе знаний - репозитории. Ядро системы управляет процессами генерации или переконфигурирования программного обеспечения. В процессе разработки адаптивных информационных систем применяется оригинальное или типовое проектирование. Оригинальное проектирование предполагает разработку информационной системы с «чистого листа» на основе сформулированных требований. Реализация этого подхода основана на использовании систем автоматизированного проектирования, или CASE-технологий (Designer2000, SilverRun, Natural Light Storm и др.). При типовом проектировании осуществляется адаптация типовых разработок к особенностям проблемной области. Для реализации этого подхода применяются инструментальные средства компонентного (сборочного) проектирования информационных систем (R/3, BAAN IV, Prodis и др.). Главное отличие подходов состоит в том, что при использовании CASE-технологии на основе репозитория при изменении проблемной области каждый раз выполняется генерация программного обеспечения, а при использовании сборочной технологии - конфигурирование программ и только в редких случаях - их переработка.

## **4. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ**

### **4.1. Хранилища данных**

Современным организациям присуще использование как старых централизованных систем, так и новых распределенных систем. Широкое разнообразие технологий обеспечено также большим числом продавцов программных продуктов. Сталкиваясь с таким технологическим и коммерческим окружением, менеджеры должны использовать новые понятия в управляющих информационных технологиях. Одним из таких понятий является складирование данных (или хранение данных)[1].

Определение понятия «хранилище данных» начинается с физического разделения оперативного окружения, поддерживающего решения. В сердцевине многих компаний используется хранилище оперативных данных, обычно извлекаемых из неавтономных систем обработки транзакций в режиме онлайн (OLTP – online transaction processing – оперативная обработка транзакций) и базирующихся на головных компьютерах (фейн – фрейм; mainframe).

OLTP – системы, например, для финансов, инвентаризации запасов или управления, также производят оперативные данные. В оперативном окружении доступ к данным, прикладные логические задачи и логика представления данных тесно взаимодействуют вместе, обычно в нереляционных БД. Эти нереляционные хранилища данных не очень способствуют эффективному поиску данных при поддержке решений.

Целью хранилища данных является установление такого репозитория данных, который делает оперативные данные доступными в форме, которая приемлема для приложений в информационных системах поддержки решений. Как часть этого нового уровня доступности, процесс должен преобразовать детализированные по уровням оперативные данные в реляционную форму, которая делает их более подходящими для обработки при поддержке решений.

Хранение данных (или хранение информации) – это понятие, предложенное и разработанное для обеспечения решения проблемы эффективного доступа к данным, описанным выше. Хранилище данных объединяет различные источники данных в простые источники для доступа конечного пользователя.

Существует несколько базовых структур для хранения данных. Основными являются двухрядные и трехрядные структуры. Вариант трехрядной архитектуры представлен на рис. 2.3.1.

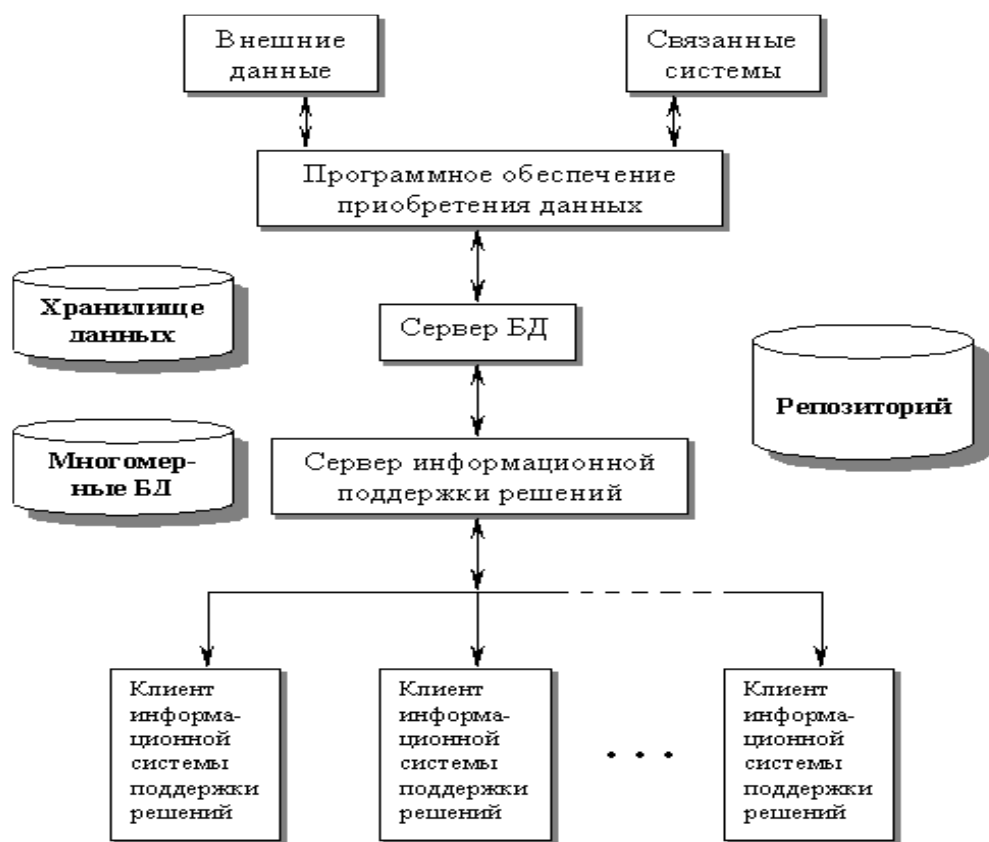


Рисунок 2.3.1 – Трехрядная архитектура хранилища данных 43

Перед размещением в хранилище данные, поступающие из внутренних (связанных) и внешних источников извлекаются, очищаются, фильтруются и суммируются посредством специального ПО. Далее данные снова обрабатываются и помещаются в дополнительную специальную многомерную БД (третий ряд в архитектуре), организованную для легкого многомерного представления. Пользователи информационной системы поддержки решений могут запрашивать сервер и осуществлять анализ.

В двухрядной архитектуре отсутствует многомерная БД или сервер. Подобное хранение данных наиболее подходит для организаций, где:

- данные хранятся в различных системах;
- используется информационно – аналитический подход к менеджменту;

- имеется большая и разнообразная покупательская и клиентская база;
- одни и те же данные представлены по – разному в различных системах;
- данные хранятся в высокотехнических, трудных для расшифровки форматах.

#### **4.2. Оперативная аналитическая обработка данных и интеллектуальный анализ данных**

В течение многих лет информационные технологии концентрировались на построении систем поддержки обработки корпоративных транзакций. Такие системы должны быть визуально отказоустойчивыми и обеспечивать быстрый отклик. Эффективное решение было обеспечено OLTP, которые сосредотачивались на распределенном реляционном окружении БД.

Более поздним достижением в этой области явилось добавление архитектуры клиент – сервер. Было издано много инструментов для развития OLTP приложений.

Доступ к данным часто требуется как OLTP приложениям, так и информационным системам поддержки решений. К сожалению, попытка обслужить оба типа запросов может быть проблематична. Поэтому некоторые компании избрали путь разделения БД на OLTP тип и OLAP тип.

*OLAP (Online Analytical Processing – оперативная аналитическая обработка)* – это информационный процесс, который дает возможность пользователю запрашивать систему, проводить анализ и т.д. в оперативном режиме (онлайн). Результаты генерируются в течении секунд.

С другой стороны, в OLTP системе огромные объемы данных обрабатываются так скоро, как они поступают на вход.

OLAP системы выполнены для конечных пользователей, в то время как OLTP системы делаются для профессиональных пользователей ИС. В OLAP предусмотрены такие действия, как генерация запросов, запросы нерегламентированных отчетов, проведение статистического анализа и построение мультимедийных приложений.

Для обеспечения OLAP необходимо работать с хранилищем данных (или многомерным хранилищем), а также с набором инструментальных средств, обычно с многомерными способностями. Этими средствами могут быть инструментарий запросов, электронные таблицы, средства добычи данных (Data Mining), средства визуализации данных и др.

В основе концепции OLAP лежит принцип многомерного представления данных. Э. Кодд рассмотрел недостатки реляционной модели, в первую очередь указав на невозможность объединять, просматривать и анализировать данные с точки зрения множественности измерений, то есть самым понятным для корпоративных аналитиков способом, и определил общие требования к системам OLAP, расширяющим функциональность реляционных СУБД и включающим многомерный анализ как одну из своих характеристик.

В большом числе публикаций аббревиатурой OLAP обозначается не только многомерный взгляд на данные, но и хранение самих данных в многомерной БД. Вообще говоря, это неверно, поскольку сам Кодд отмечает, что реляционные БД были, есть и будут наиболее подходящей технологией для хранения корпоративных данных. Необходимость существует не в новой

технологии БД, а скорее, в средствах анализа, дополняющих функции существующих СУБД и достаточно гибких, чтобы предусмотреть и автоматизировать разные виды интеллектуального анализа, присущие OLAP.

Кодд определил 12 правил, которым должен удовлетворять программный продукт класса OLAP. Эти правила:

1. Многомерное концептуальное представление данных.
2. Прозрачность.
3. Доступность.
4. Устойчивая производительность.
5. Клиент – серверная архитектура.
6. Равноправие измерений.
7. Динамическая обработка разреженных матриц.
8. Поддержка многопользовательского режима.
9. Неограниченная поддержка кроссмерных операций.
10. Интуитивное манипулирование данными.
11. Гибкий механизм генерации отчетов.
12. Неограниченное количество измерений и уровней агрегации.

Набор этих требований, послуживший фактическим определением OLAP, следует рассматривать как рекомендательный, а конкретные продукт оценивать по степени приближения к идеально полному соответствию всем требованиям.

#### **Интеллектуальный анализ данных.**

Интеллектуальный анализ данных (ИАД), или Data Mining, - термин, используемый для описания открытия знаний в базах данных, выделения знаний, изыскания данных, исследования данных, обработки образцов данных, очистки и сбора данных; здесь же подразумевается сопутствующее ПО. Все эти действия осуществляются автоматически и позволяют получать быстрые результаты даже непрограммистам.

Запрос производится конечным пользователем, возможно на естественном языке. Запрос преобразуется в SQL – формат. SQL запрос по сети поступает в СУБД, которая управляет БД или хранилищем данных. СУБД находит ответ на запрос и доставляет его назад. Пользователь может затем разрабатывать презентацию или отчет в соответствии со своими требованиями.

Многие важные решения в почти любой области бизнеса и социально сферы основываются на анализе больших и сложных БД. ИАД может быть очень полезным в этих случаях.

Методы интеллектуального анализа данных тесно связаны с технологиями OLAP и технологиями построения хранилищ данных. Поэтому наилучшим вариантом является комплексный подход к их внедрению.

Для того чтобы существующие хранилища данных способствовали принятию управленческих решений, информация должна быть представлена аналитику в нужной форме, то есть он должен иметь развитые инструменты доступа к данным хранилища и их обработки.

Очень часто информационно – аналитические системы, создаваемые в расчете на непосредственное использование лицами, принимающими решения, оказываются чрезвычайно просты в применении, но жестко ограничены в функциональности. Такие статические системы называются Информационными системами руководителя. Они содержат в себе предопределенные множества

запросов и, будучи достаточными для повседневного обзора, неспособны ответить на все вопросы к имеющимся данным, которые могут возникнуть при принятии решений. Результатов работы такой системы, как правило, являются многостраничные отчеты, после тщательного изучения которых у аналитика появляется новая серия вопросов. Однако каждый новый запрос, непредусмотренный при проектировании такой системы, должен быть сначала формально описан, закодирован программистом и только затем выполнен. Время ожидания в таком случае может составлять часы и дни, что не всегда приемлемо. Таким образом, внешняя простота статистических ИС поддержки решений, за которую активно борется большинство заказчиков информационно – аналитических систем, оборачивается потерей гибкости.

Динамические ИС поддержки решений, напротив, ориентированы на обработку нерегламентированных (*ad hoc*) запросов аналитиков к данным. Работа аналитиков с этими системами заключается в интерактивной последовательности формирования запросов и изучения их результатов.

Но динамические ИС поддержки решений могут действовать не только в области оперативной аналитической обработки (OLAP). Поддержка принятия управленческих решений на основе накопленных данных может выполняться в трех базовых сферах.

1. Сфера детализированных данных. Это область действия большинства систем, нацеленных на поиск информации. В большинстве случаев реляционные СУБД отлично справляются с возникающими здесь задачами. Общеизвестным стандартом языка манипулирования реляционными данными является SQL. Информационно – поисковые системы, обеспечивающие интерфейс конечного пользователя в задачах поиска детализированной информации, могут использоваться в качестве надстроек как над отдельными базами данных транзакционных систем, так и над общим хранилищем данных.

2. Сфера агрегированных показателей. Комплексный взгляд на собранную в хранилище данных информацию, ее обобщение и агрегация и многомерный анализ являются задачами систем OLAP. Здесь можно или ориентироваться на специальные многомерные СУБД, или оставаться в рамках реляционных технологий. Во втором случае заранее агрегированные данные могут собираться в БД звездообразного вида, либо агрегация информации может производиться в процессе сканирования детализированных таблиц реляционной БД.

3. Сфера закономерностей. Интеллектуальная обработка производится методами интеллектуального анализа данных главными задачами которых являются поиск функциональных и логических закономерностей в накопленной информации, построение моделей и правил, которые объясняют найденные аномалии и/или прогнозируют развитие некоторых процессов.

Полная структура информационно – аналитической системы построенной на основе хранилища данных, показана на рис. 2.3.2. В конкретных реализациях отдельные компоненты этой схемы часто отсутствуют.

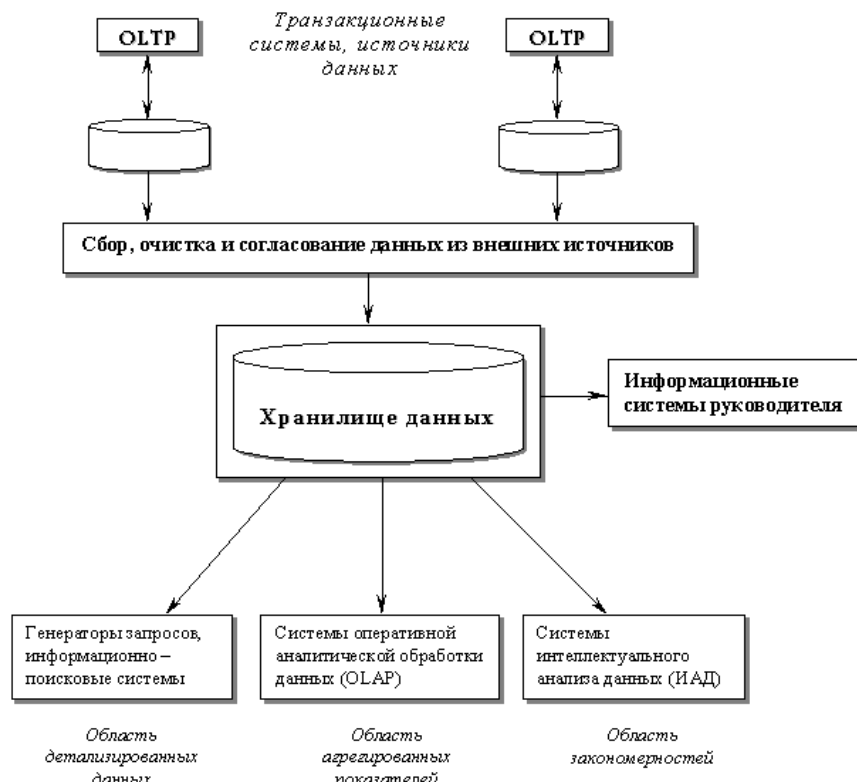


Рисунок 2.3.2 - Структура корпоративной информационно – аналитической системы

### 4.3. Интеллектуальные базы данных

Развитие приложений ИС требует реализации более легкого и удобного доступа к базам данных.

Технологии ИИ, особенно ЭС и искусственные нейронные сети (ИНС), могут сделать доступ и манипуляции в сложных БД проще. Одним из путей является усиление роли СУБД в обеспечении этого совместно со способностью вывода заключений, что в результате получило общее название *интеллектуальная БД*.

Один из вариантов интеграции ЭС и БД показан на рис. 2.2.3.

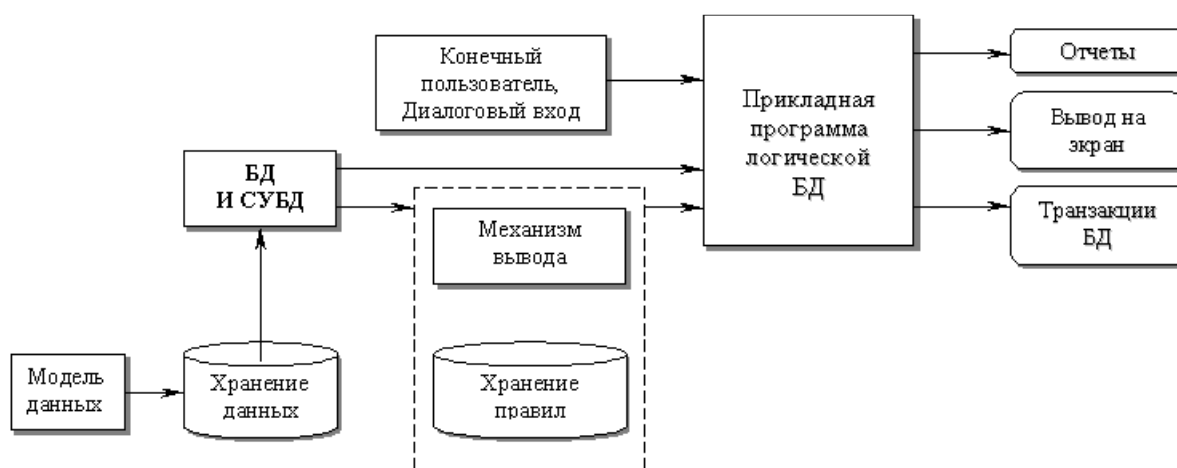


Рисунок 2.2.3 - Структура интеллектуальной базы данных, представляющая один из способов интеграции ЭС и БД

Трудности в соединении ЭС с большими БД являются главной проблемой даже для больших корпораций. Многие продавцы ПО, осознавая важность такой



интеграции, развивают свою программную продукцию для ее поддержки. Примером такого продукта является реляционная СУБД компании Oracle, которая объединяет функциональность ЭС с БД и представляет в форме оптимизатора запросов, которые отбирает наиболее эффективные пути следования запросов БД.

Оптимизация важна для пользователей, т.к. с такой способностью им нужно знать только несколько правил и команд для использования БД.

Некоторые программные инструменты для добычи данных включают интеллектуальные системы, которые поддерживают интеллектуальный поиск. Интеллектуальная добыча и анализ данных (ИАД) позволяет открыть информацию в хранилищах данных, когда запросы и отчеты не могут быть обнаружены.

Инструменты ИАД находят образцы в данных и выводят из них правила. Эти образцы и правила могут быть использованы для руководства при принятии решений и прогнозировании результатов этих решений. ИАД может ускорить анализ путем сосредоточения внимания на наиболее важных переменных.

Пять типов информации может быть применено при ИАД: ассоциации, последовательности, классификации, кластеры и прогнозирование.

Основными типами программных инструментариев, используемых в ИАД, являются:

- рассуждения на основе прецедентов;
- нейронные вычисления;
- интеллектуальные агенты;
- другие средства: деревья решений, ролевая индукция, визуализация данных.

## **5. ЭКСПЕРТНЫЕ СИСТЕМЫ - СИСТЕМЫ, БАЗИРУЮЩИЕСЯ НА ЗНАНИЯХ. ИНЖЕНЕРИЯ ЗНАНИЙ**

### **5.1. Экспертные системы – основная разновидность интеллектуальных систем**

Название «экспертная система» происходит из термина «экспертная система, базирующаяся на знаниях». *Экспертная система (ЭС)* – это система, которая использует человеческие знания, встраиваемые в компьютер, для решения задач, которые обычно требуют человеческой экспертизы. Хорошо разработанные системы имитируют процесс рассуждения экспертов, используя это для решения специфических задач.

Такие системы могут использоваться не экспертом для улучшения их способностей и возможностей в решении задач определенного класса в конкретной предметной области. ЭС могут быть также использованы для распространения источников редких знаний. В конечном счете, такие системы могут функционировать лучше, чем некоторые отдельные эксперты - люди при выработке решения или суждения в специфической, обычно узкой области экспертизы. Эта возможность может иметь значительное влияние как на деятельность таких профессиональных консультантов, как финансовые аналитики, юристы, аудиторы и др., так и на организации и их менеджмент.



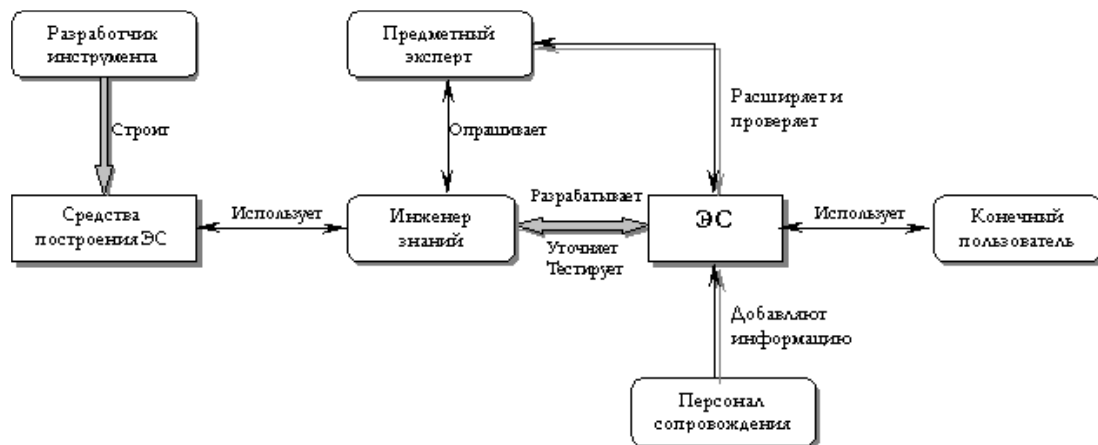


Рисунок 3.1.1 - Процесс построения ЭС

Технологию построения ЭС часто называют инженерией знаний. Этот процесс требует специфической формы взаимодействия создателя ЭС, которого называют инженером знаний, и одного или нескольких экспертов в некоторой предметной области. Инженер знаний «извлекает» из экспертов процедуры, стратегии, эмпирические правила, которые они используют при решении задач, и встраивает эти знания в ЭС (см. рис.3.1.1).

ЭС – это сложные программы, которые манипулируют знаниями в целях получения эффективного решения в узкой предметной области. Как и настоящий человек – эксперт, эти системы используют символическую логику и эвристики (эмпирические правила) чтобы найти решения. И, они могут ошибаться, но обладают способностью учиться на своих ошибках.

*Основные понятия ЭС.* Основными понятиями ЭС являются: экспертиза, эксперты, проведение экспертизы, вывод и объяснительные способности.

*Экспертиза* – это обширное, специфическое знание для решения задачи, извлеченное из обучения, чтения и опыта. Следующие типы знаний являются примерами того, что включает себя экспертиза:

- теории о проблемной области;
- правила и процедуры относительно проблемной области;
- правила (эвристики) о том, что делать в данной проблемной ситуации;
- глобальные стратегии для решения таких типов задач;
- метазнания (знания о знаниях);
- факты о проблемной области.

Эти типы знаний дают возможность ЛПР принимать решения лучше и быстрее при решении сложных задач.

*Эксперты.* Трудно дать определение понятию эксперт, т.к. мы в действительности говорим о разных степенях или уровнях экспертизы. Вопросом является также каким объемом экспертных знаний в данной области и какими навыками должен обладать, чтобы быть квалифицированным в качестве эксперта?

Обычно, человеческая экспертиза включает многогранное интеллектуальное поведение, которое вовлекает в процесс следующие виды деятельности:

- - выявление и формулировка проблемы и задачи;
- - решение задачи быстро и надлежащим образом;

- - объяснение решения;
- - обучение из опыта;
- - реструктуризация знаний;
- - при необходимости от устоявшихся правил и шаблонов;
- - определение уместности и соответствия;
- - осознание ограничений.

Для имитации эксперта – человека необходимо создать компьютерную систему, проявляющую все эти характеристики. Однако в современных ЭС прежде всего исследованы и разработаны вторая и третья из этих видов деятельности (решение задач и объяснение решений).

*Проведение экспертизы.* Целью ЭС является проведение экспертизы путем аккумуляции знаний от экспертов и предоставлению их другим людям (неэкспертам). В этот процесс вовлечены четыре вида деятельности: *извлечение знаний (из экспертов или других источников), представление знаний (в компьютере), вывод знаний и передача знаний пользователю. Знания хранятся в компьютере в базе знаний (БЗ).*

*Вывод.* Уникальной чертой ЭС является их способность рассуждать («думать»). Имеется в виду, что необходимые знания для экспертизы хранятся в БЗ, программа может иметь доступ к соответствующим данным в БД, а ЭС может делать логический вывод, получая нужное знание, зачастую не хранящееся в явном виде в БЗ. Процесс вывода осуществляется составляющей системы, которая называется *машина вывода*.

*Способность объяснять.* Другой уникальной чертой ЭС является ее способность объяснять свои советы или рекомендации. Объяснение и обоснование производятся *подсистемой объяснений*. Она дает возможность системе проверять свои рассуждения и объяснять их действия.

## 5.2. Функциональные возможности и характеристика ЭС

Основными характеристиками ЭС являются:

- - накопление и организация знаний – одна из самых важных характеристик ЭС;
- - знания – основа ЭС, они являются *явными и доступными*, что отличает эти системы от большинства традиционных программ;
- - ЭС применяет для решения проблем *высококачественный опыт* квалифицированных экспертов. Именно высококачественный опыт в сочетании с умением его применять делает системы рентабельной. Этому также способствует гибкость системы;
- - Наличие прогностических способностей. ЭС может объяснить каким образом новая ситуация привела к изменениям;
- - Ведущие специалисты уходят, но их опыт остается и используется в ЭС;
- - ЭС можно использовать для обучения и тренировки.

*Преимущества ЭС.*

Затем разрабатывать ЭС. Не лучше ли обратиться к человеческому опыту как это было в прошлом. Приведем доводы в пользу ЭС.

Таблица 3.1.1

Сравнение человеческой и искусственной компетентности <b>Человеческая компетентность</b>	<b>Искусственная компетентность</b>
Непрочная Трудно представляемая Трудно документируемая Непредсказуемая Дорогая	Постоянная Легко передаваемая Легко документируемая Устойчивая Приемлемая по затратам

Здесь очевидны преимущества искусственной компетентности. Кроме того, эксперт – человек может принимать различные решения в тождественных ситуациях из-за эмоциональных факторов (влияние дефицита времени, влияние стресса).

*Необходимость человека в контуре управления.* Почему необходимо оставить для человека место в системе?

Если искусственная компетентность настолько лучше человеческой, почему бы полностью не отказаться от экспертов – людей, заменив их ЭС? О несостоятельности подобных предложений и рассуждений говорят много доводов. Приведем некоторые из них:

- Хотя ЭС хорошо справляются со своей работой, но в некоторых областях деятельности человеческая компетентность превосходит любую искусственную. Это не есть отражение фундаментальных ограничений ИИ, но характерно для современного его состояния. Например, область творчества.
- Обучение: человеческая компетентность пока превосходит искусственную. Эксперты адаптируются к изменяющимся условиям, приспособливают свои стратегии к новым обстоятельствам. ЭС мало приспособлены к обучению новым концепциям и правилам. Обучающие программы разработаны для простых задач и мало пригодны, когда требуется учитывать всю сложность реальных задач.
- Эксперты могут непосредственно воспринимать комплекс входной сенсорной информации (визуальной, звуковой, осязательной, обонятельной и тактильной). ЭС – только символы. Хотя в отдельных направлениях разработки инженерных и производственных интеллектуальных систем получены реальные результаты определенной обработки сенсорной информации.
- Эксперты – люди могут охватить картину в целом, все аспекты проблемы и понять, как они соотносятся с основной задачей. ЭС стремится сосредоточить на самой задаче, хотя смежные задачи могут повлиять на решение основной.
- Люди, эксперты и не эксперты, имеют то, что мы называем здравым смыслом, или общедоступными знаниями. Это широкий спектр общих знаний о мире, о том, какие законы в нем действуют, т.е. знания, которыми каждый из нас обладает, приобретает из опыта и которыми постоянно пользуется. Из-за огромного объема знаний, образующих здравый смысл, не существует легкого способа

встроить их в интеллектуальную программу. Знания здравого смысла включают знания о том, что вы знаете и чего не знаете.

- Функциональные возможности ЭС определяются двумя ее главными системными частями: средой развития и средой рекомендаций (рис. 3.1.2). Среда развития используется разработчиком ЭС для построения компонентов и размещения знаний в БЗ. Среда рекомендаций используется неэкспертами для получения экспертных знаний и советов.

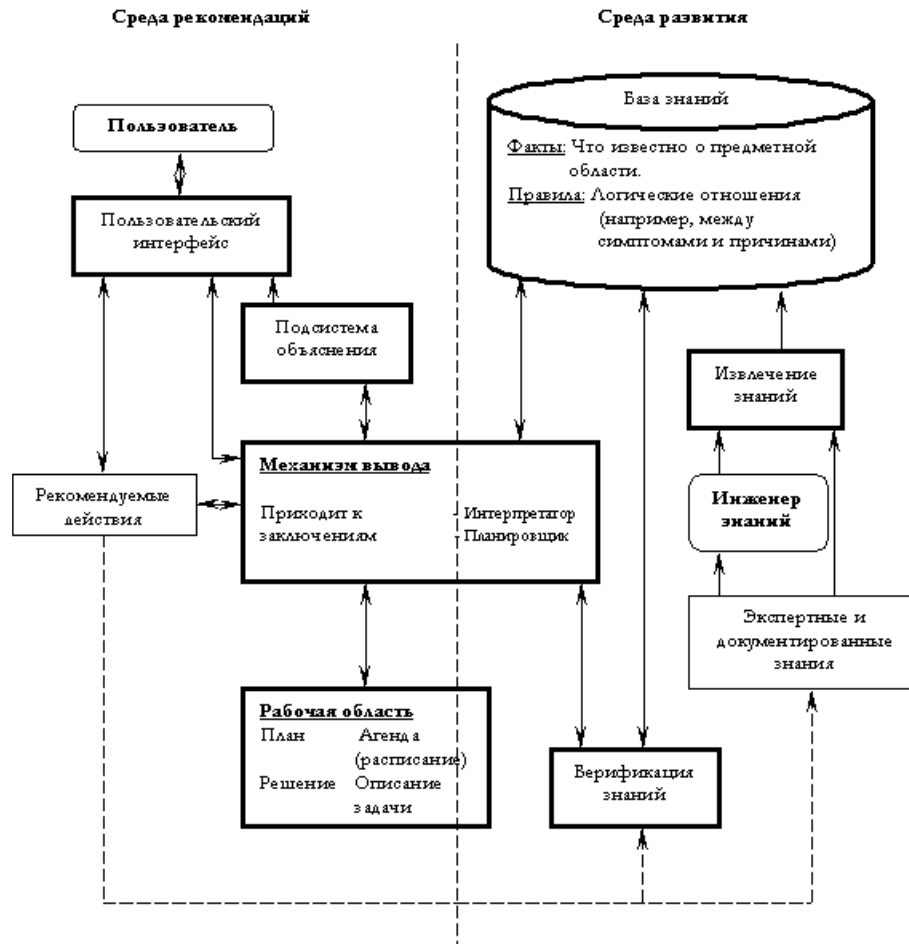


Рисунок 3.1.2 - Структура ЭС и ее окружение 54

Три главных компонента, которые проявляются в каждой ЭС - это БЗ, механизм вывода и пользовательский интерфейс. Хотя вообще ЭС могут содержать следующие компоненты:

- - подсистема извлечения знаний;
- - БЗ;
- - механизм вывода;
- - пользовательский интерфейс;
- - рабочая область;
- - подсистема объяснения;
- - подсистема верификации знаний.

Обычно, большинство ЭС не содержат подсистему верификации знаний. Существует также большие колебания в содержании и способностях каждой компоненты.

*Извлечение знаний* представляет собой накопление, передачу и преобразование экспертиз решения задачи от экспертов или документированных источников знаний компьютерной программой для конструирования или расширения БЗ. Потенциальные источники знаний включают экспертов, учебники, справочники, мультимедийные документы, базы данных (общественные или частные), специальные исследовательские отчеты и информацию, доступную через Интернет.

Извлечение знаний из экспертов является сложной задачей, которая часто создает узкое место при построении ЭС.

Современные условия требуют от знаний и способностей взаимодействовать с одним или более людьми – экспертами при построении БЗ. Инженер знаний помогает эксперту структурировать проблемную область путем интерпретации и объединения ответов человека на вопросы, проводя аналогии, предлагая контрпримеры и выявляя концептуальные трудности.

*База знаний* содержит знания, необходимые для понимания, формулирования и решения задач. Она включает два основных элемента: факты, такие как проблемная ситуация и теоретические знания о проблемной области; и специальные эвристики и правила, которые направляют использование знаний при решении специфических задач в отдельной области. Кроме того, механизм вывода, тесно связанный с БЗ, содержит стандартные правила решения задач и принятия решений. Эвристики выражают неформальные знания, мнения и суждения в прикладной области. Глобальные стратегии, которые могут быть как эвристиками, так и частью теории проблемной области, обычно включаются в БЗ. Знания, а не просто факты, являются первоначальным необработанным материалом экспертных систем. Информация и знания в БЗ представлены и включены в компьютерную программу путем реализации процесса, называемого представлением знаний.

*Механизм вывода* является мозгом ЭС, его также называют управляющая структура или *интерпретатор правил* (в ЭС, основанных на правилах).

Эта компонента является в основном компьютерной программой, которая обеспечивает методологию для рассуждения об информации в БЗ и в рабочей области, а также для формулирования заключений. Она обеспечивает указания о том, как использовать знания системы при реализации аренды (расписания запланированных действий в рабочей области), которая организует и управляет шагами, предпринимаемыми для решения задачи.

Механизм вывода имеет два главных элемента:

- интерпретатор, который выполняет выбранные позиции аренды, используя соответствующие правила БЗ.
- Планировщик, который поддерживает управление агендой. Он оценивает результаты используемых правил вывода в свете их приоритетов или других критериев в агенде.

*Пользовательский интерфейс.* ЭС содержат языковой процессор для дружественного, проблемно – ориентированного общения между пользователем и компьютером. Общение наилучшим образом выполняется на естественном языке. Иногда оно дополняется меню и графикой.

*Рабочая область* – это область, расположенная отдельно для описания текущей задачи, как определено входными данными. Она также используется

для запоминания промежуточных результатов. В рабочей области запоминаются промежуточные гипотезы и решения.

Могут быть запомнены три типа решений: план (как атаковать задачу), agenda (потенциальные действия, ожидающие выполнения) и решение (гипотезы – кандидаты и альтернативные направления действий, которые система сгенерировала до сих пор).

*Подсистема объяснения.* Способность отслеживать ответственность и соответствие заключений их источникам является решающей и при проведении экспертизы, и при решении задачи. Подсистема объяснения может отслеживать такую ответственность и объяснять поведение экспертной системы интерактивно отвечая на вопросы.

*Подсистема верификации и совершенствования знаний.* Эксперты обладают способностями верифицировать и совершенствовать знания. То есть, они могут анализировать свои собственные знания и их использование, обучаться от них и улучшать их для будущих консультаций. Аналогично, такая эволюция необходима в компьютеризованном обучении, так, чтобы программа могла анализировать рассуждения под углом их успеха или неудачи. Это может привести к улучшениям, и как результату, более точным БЗ и более эффективному рассуждению. Такой составляющей в настоящее время пока нет в коммерческих ЭС, но она разрабатывается в экспериментальных ЭС.

**5.3.      Области применения экспертных систем**

ЭС могут быть классифицированы несколькими путями. Одним из них является классификация по основным проблемным областям, на которые они ориентированы. При этом проблемные области определяются основными классами задач, эффективно решаемыми методами ЭС. Например, диагностика может быть определена как «выявление неисправностей системы через наблюдения». Диагностика является общей по своей сути деятельностью, совершаемой в медицине, организационных исследованиях, компьютерных операциях, контроле за оборудованием. Основные классы задач, для решения которых создаются экспертные системы перечисленные в таблице 3.1.2.

Таблица 3.1.2

Основные классы решения задач, решаемые ЭС <b>Класс</b>	<b>На решение какой задачи направлена</b>
Интерпретация	Выявление описаний ситуации из наблюдений
Предсказание	Выявление похожих последствий в данной ситуации.
Диагностика	Выявление неисправности системы через наблюдения.
Проектирование	Конфигурирование и разработка объектов, удовлетворяющих определенным требованиям.
Планирование	Разработка планов для достижения целей.
Мониторинг	Сравнение наблюдений с планами, сигнализируя об отклонениях и исключениях.
Отладка	Выявление и устранение неисправностей.
Управление	Интерпретирование, предсказывание восстановление и мониторинг поведения системы.

Некоторые ЭС принадлежат к двум или более из этих категорий. Дадим краткое описание каждой из этих категорий.

*Системы интерпретации* выявляют описания ситуации из наблюдений. Это категория включает наблюдения, понимание речи, анализ образов, интерпретацию сигналов и многие другие виды интеллектуального анализа. Система интерпретации объясняет наблюдаемые данные путем присвоения им символических значений, описывающих ситуацию.

*Системы предсказания* включают прогнозирование погоды, демографические предсказания, экономическое прогнозирование, оценки урожайности, а также военное, маркетинговое и финансовое прогнозирование.

*Системы диагностики* включают диагностику в медицине, электронике, механике и программном обеспечении. Диагностирующие системы обычно соотносят наблюдаемые поведенческие отклонения с причинами, лежащими в основе.

*Системы проектирования* разрабатывают конфигурации объектов, которые удовлетворяют определенным требованиям задачи проектирования. Такие задачи включают конструирование зданий, планировка расположения оборудования и др. Эти системы конструируют различные взаимосвязи описаний объектов друг с другом и проверяют, удовлетворяют ли эти конфигурации установленным ограничениям и требованиям.

*Системы планирования* специализируются на задачах планирования, например, такой как автоматическое программирование. Они также работают с кратко и долгосрочным планированием в управлении проектами, маршрутизация, коммуникация, разработка продукта, военные приложения, производственное и финансовое планирование.

*Системы мониторинга* сравнивают наблюдения поведения системы со стандартами, которые представляются определяющими для достижения цели. Эти решающие выявления соответствуют потенциальным недостаткам на предприятии. Существует много компьютерных систем мониторинга: от контроля движения воздушных потоков до задач управления сбором налогов.

*Системы управления и контроля* адаптивно управляют всеобщим поведением системы. Для осуществления этого система управления должна периодически интерпретировать текущую ситуацию, предсказывать будущее, диагностировать причины ожидаемых проблем, формулировать план устранения этих проблем и осуществлять мониторинг его выполнения для обеспечения успеха.

Не все задачи, которые обычно образуются в каждом из этих классов, подходят для ЭС. Однако есть тысячи задач, которые подходят к этим классам.

Рассмотренные классы задач ЭС, определяющие проблемные области, решаются в различных предметных областях. Области применения существующих на сегодняшний день ЭС охватывают: медицину, геологию, научные исследования в области химии и биологии, военное дело, инженерное дело, космическую технику, метеорологию, экологию, производство, управление процессами, юриспруденцию, маркетинг, финансы, банковское дело и др.

Сегодня ЭС используются многими большими и средними организациями как главный инструмент для улучшения производительности и качества. Они являются также важным инструментом для поддержания стратегических решений и реинжиниринга бизнес – процессов.

Важными работами, отражающими позитивные результаты использования ЭС в таких предметных областях как производство, финансы, бизнес и менеджмент (как проблемная область).

### **Статические и динамические ЭС.**

При классификации ЭС по проблемным областям на основе классов и типов задач, важно исследовать и оценивать характер проблемной и предметной областей с позиций динамики решаемых задач, важности временного фактора и темпоральной информации.

То есть, если исходная информация о предметной области или окружающем мире, на основе которой решается задача, не изменяется за время решения задачи, то такую предметную область можно условно назвать статической предметной областью, и ее представление в ЭС будет статическим. Если информация о предметной области изменяется за время решения задач, то такую предметную область можно назвать динамической предметной областью.

Если задачи, решаемые ЭС, явно не учитывают фактор времени и не изменяют в процессе решения данные о реальной действительности, то это статические задачи. Если задачи при решении требуют учета фактора времени или изменяют данные о реальных внешних процессах, то это динамические задачи.

То есть, ЭС работает в статической проблемной среде, если она использует статическое представление и решает статические задачи. Если ЭС использует динамическое представление или решает динамические задачи, то, соответственно она работает в динамической проблемной среде. Важность времени в динамических проблемных средах определила название таких ЭС, как систем, работающих в реальном времени.

Значимость инструментальных средств реального времени определяется не столько их бурным коммерческим успехом (хотя и это достойно тщательного анализа), но, в первую очередь, тем, что только с помощью подобных средств создаются стратегически значимые приложения в таких областях, как управление непрерывными производственными процессами, аэрокосмические исследования, транспортировка и переработка нефти и газа, управление атомными и тепловыми электростанциями, финансовые операции и многие другие.

Экспертные системы реального времени, решают следующие классы задач: мониторинг в реальном масштабе времени, обнаружения неисправностей, диагностика, оперативное планирование, системы – советчики оператора.

Статические экспертные системы не способны решать подобные задачи, так как они не выполняют требования, предъявляемые к системам, работающим в реальном времени:

1. Представлять изменяющиеся во времени данные, поступающие от внешних источников, обеспечивать хранение и анализ изменяющихся данных.
2. Выполнять временные рассуждения о нескольких различных асинхронных процессах одновременно (т.е. планировать в соответствии с приоритетами обработку поступивших в систему процессов).
3. Обеспечивать механизм рассуждения при ограниченных ресурсах (время, память). Реализация этого механизма предъявляет требования к высокой скорости работы системы, способности одновременно решать несколько задач (т.е. операционные системы UNIX, VMS, Windows NT).



4. Осуществлять постоянный мониторинг процесса, и при необходимости автоматически запускать механизм логического вывода решений по устранению критических ситуаций с одновременным информированием ЛПР.

5. Моделировать "окружающий мир", рассматриваемый в данном приложении, обеспечивать создание различных его состояний.

6. Протоколировать свои действия и действия персонала, обеспечивать восстановление после сбоя.

7. Обеспечивать наполнение базы знаний для приложений реальной степени сложности с минимальными затратами времени и труда (необходимо использование объектно-ориентированной технологии, общих правил, модульности и т.п.).

8. Обеспечивать настройку системы на решаемые задачи (проблемная/предметная ориентированность).

9. Обеспечивать создание и поддержку пользовательских интерфейсов для различных категорий пользователей.

10. Обеспечивать уровень защиты информации (по категориям пользователей) и предотвращать несанкционированный доступ.

Специфические требования, предъявляемые к экспертной системе реального времени, приводят к тому, что их архитектура отличается от архитектуры статических систем. Появляются две новые подсистемы: моделирования внешнего окружения и сопряжения с внешним миром (датчиками, контроллерами, СУБД и т.п.) – и значительные изменения, которым подвергаются оставшиеся подсистемы.

При создании ЭС реального времени, приобретают важное значение несколько новых по сравнению с обычными ЭС соображений. Главное из них – эффективность исполнения. В обычных ЭС факты и знания, на которых основываются рассуждения, носят статический характер. В производственных системах факты, или показания технологических датчиков, являются динамическими. В таких ЭС может существовать до нескольких тысяч показаний приборов и аварийных сигналов, заметно меняющих величину или состояние в течение нескольких минут.

## **6. СТРУКТУРА И ТЕХНОЛОГИЯ РАЗРАБОТКИ ЭКСПЕРТНОЙ СИСТЕМЫ**

### **6.1. Структура экспертных систем**

Типичная статическая ЭС состоит из следующих основных компонентов (рис. 3.2.1):

- решателя (интерпретатора);
- рабочей памяти (РП), называемой также базой данных (БД);
- базы знаний (БЗ);
- компонентов приобретения знаний;
- объяснительного компонента;
- диалогового компонента.

*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 (в первую очередь долгосрочных), хранимых в системе.

*База знаний (БЗ)* в ЭС предназначена для хранения долгосрочных данных, описывающих рассматриваемую область (а не текущих данных), и правил, описывающих целесообразные преобразования данных этой области.

*Решатель*, используя исходные данные из рабочей памяти и знания из БЗ, формирует такую последовательность правил, которые, будучи примененными к исходным данным, приводят к решению задачи.

*Компонент приобретения знаний* автоматизирует процесс наполнения ЭС знаниями, осуществляемый пользователем-экспертом. 61

*Объяснительный компонент* объясняет, как система получила решение задачи (или почему она не получила решение) и какие знания она при этом использовала, что облегчает эксперту тестирование системы и повышает доверие пользователя к полученному результату.

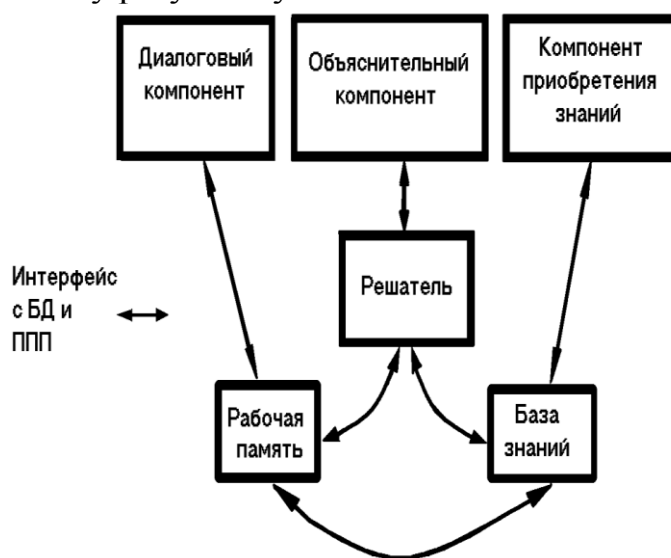


Рисунок 3.2.1 - Структура статистической ЭС

*Диалоговый компонент* ориентирован на организацию дружественного общения с пользователем как в ходе решения задач, так и в процессе приобретения знаний и объяснения результатов работы.

В разработке ЭС участвуют представители следующих специальностей:

- эксперт в проблемной области, задачи которой будет решать ЭС;
- инженер по знаниям - специалист по разработке ЭС (используемые им технологию, методы называют технологией (методами) инженерии знаний);
- программист по разработке инструментальных средств (ИС), предназначенных для ускорения разработки ЭС.

Необходимо отметить, что отсутствие среди участников разработки инженеров по знаниям (т. е. их замена программистами) либо приводит к неудаче процесс создания ЭС, либо значительно удлиняет его.

*Эксперт* определяет знания (данные и правила), 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 (если ИС разрабатывается заново), содержащее в пределе все основные компоненты ЭС, и осуществляет его сопряжение с той средой, в которой оно будет использовано.

Экспертная система работает в двух режимах: режиме приобретения знаний и в режиме решения задачи (называемом также режимом консультации или режимом использования ЭС).

*В режиме приобретения знаний* общение с ЭС осуществляет (через посредничество инженера по знаниям) эксперт. В этом режиме эксперт, используя компонент приобретения знаний, наполняет систему знаниями, которые позволяют ЭС в режиме решения самостоятельно (без эксперта) решать задачи из проблемной области. 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 с данными, характерные для рассматриваемой области.

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

*В режиме консультации* общение с ЭС осуществляет конечный пользователь, которого интересует результат и (или) способ его получения. Необходимо отметить, что в зависимости от назначения ЭС пользователь может не быть специалистом в данной проблемной области (в этом случае он обращается к ЭС за результатом, не умея получить его сам), или быть специалистом (в этом случае пользователь может сам получить результат, но он обращается к ЭС с целью либо ускорить процесс получения результата, либо возложить на ЭС рутинную работу). В режиме консультации данные о задаче пользователя после обработки их диалоговым компонентом поступают в рабочую память. Решатель на основе входных данных из рабочей памяти, общих данных о проблемной области и правил из БЗ формирует решение задачи. ЭС при решении задачи не только исполняет предписанную последовательность операции, но и предварительно формирует ее. Если реакция системы не понятна пользователю, то он может потребовать объяснения:

"Почему система задает тот или иной вопрос?", "как ответ, собираемый системой, получен?".

Структуру, приведенную на рис. 3.2.1, называют структурой статической ЭС. ЭС данного типа используются в тех приложениях, где можно не учитывать изменения окружающего мира, происходящие за время решения задачи. Первые ЭС, получившие практическое использование, были статическими.

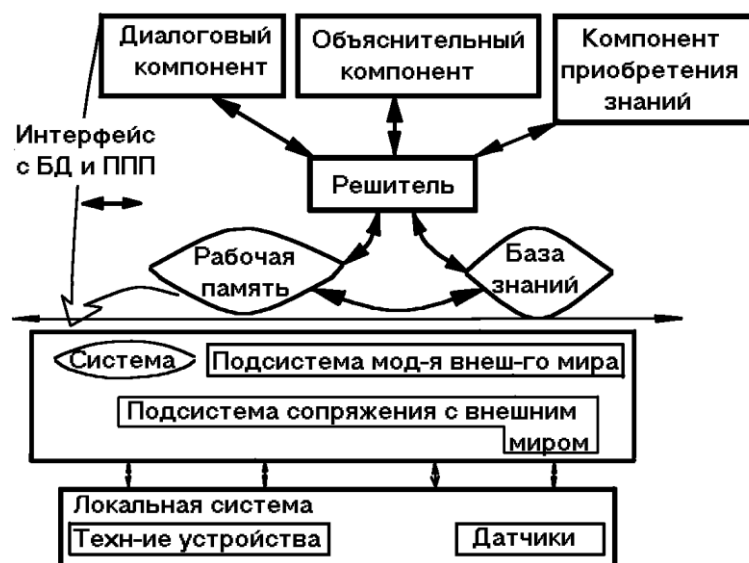


Рисунок 3.2.2 - Структура динамической ЭС

На рис. 3.2.2 показано, что в архитектуру динамической ЭС по сравнению со статической ЭС вводятся два компонента: подсистема моделирования внешнего мира и подсистема связи с внешним окружением. Последняя осуществляет связи с внешним миром через систему датчиков и контроллеров. Кроме того, традиционные компоненты статической ЭС (база знаний и машина вывода) претерпевают существенные изменения, чтобы отразить временную логику происходящих в реальном мире событий.

Подчеркнем, что структура ЭС, представленная на рис. 3.2.1 и 3.2.2, отражает только компоненты (функции), и многое остается "за кадром".

## 6.2. Этапы разработки экспертных систем

Разработка ЭС имеет существенные отличия от разработки обычного программного продукта. Опыт создания ЭС показал, что использование при их разработке методологии, принятой в традиционном программировании, либо чрезмерно затягивает процесс создания ЭС, либо вообще приводит к отрицательному результату.

Использовать ЭС следует только тогда, когда разработка ЭС возможна, оправдана и методы инженерии знаний соответствуют решаемой задаче. Чтобы разработка ЭС была возможной для данного приложения, необходимо одновременное выполнение по крайней мере следующих требований:

- 1) существуют эксперты в данной области, которые решают задачу значительно лучше, чем начинающие специалисты;
- 2) эксперты сходятся в оценке предлагаемого решения, иначе нельзя будет оценить качество разработанной ЭС;
- 3) эксперты способны вербализовать (выразить на естественном языке) и объяснить используемые ими методы, в противном случае трудно рассчитывать на то, что знания экспертов будут "извлечены" и вложены в ЭС;
- 4) решение задачи требует только рассуждений, а не действий;
- 5) задача не должна быть слишком трудной (т.е. ее решение должно занимать у эксперта несколько часов или дней, а не недель);

6) задача хотя и не должна быть выражена в формальном виде, но все же должна относиться к достаточно "понятной" и структурированной области, т.е. должны быть выделены основные понятия, отношения и известные (хотя бы эксперту) способы получения решения задачи;

7) решение задачи не должно в значительной степени использовать "здоровый смысл" (т.е. широкий спектр общих сведений о мире и о способе его функционирования, которые знает и умеет использовать любой нормальный человек), так как подобные знания пока не удастся (в достаточном количестве) вложить в системы искусственного интеллекта.

Использование ЭС в данном приложении может быть возможно, но не оправдано. Применение ЭС может быть *оправдано* одним из следующих факторов:

- решение задачи принесет значительный эффект, например экономический;
- использование человека-эксперта невозможно либо из-за недостаточного количества экспертов, либо из-за необходимости выполнять экспертизу одновременно в различных местах;
- использование ЭС целесообразно в тех случаях, когда при передаче информации эксперту происходит недопустимая потеря времени или информации;
- использование ЭС целесообразно при необходимости решать задачу в окружении, враждебном для человека.

Приложение *соответствует* методам ЭС, если решаемая задача обладает совокупностью следующих характеристик:

1) задача может быть естественным образом решена посредством манипуляции с символами (т.е. с помощью символических рассуждений), а не манипуляций с числами, как принято в математических методах и в традиционном программировании;

2) задача должна иметь эвристическую, а не алгоритмическую природу, т.е. ее решение должно требовать применения эвристических правил. Задачи, которые могут быть гарантированно решены (с соблюдением заданных ограничений) с помощью некоторых формальных процедур, не подходят для применения ЭС;

3) задача должна быть достаточно сложна, чтобы оправдать затраты на разработку ЭС. Однако она не должна быть чрезмерно сложной (решение занимает у эксперта часы, а не недели), чтобы ЭС могла ее решать;

4) задача должна быть достаточно узкой, чтобы решаться методами ЭС, и практически значимой.

При разработке ЭС, как правило, используется концепция "быстрого прототипа". Суть этой концепции состоит в том, что разработчики не пытаются сразу построить конечный продукт. На начальном этапе они создают прототип (прототипы) ЭС. Прототипы должны удовлетворять двум противоречивым требованиям: с одной стороны, они должны решать типичные задачи конкретного приложения, а с другой - время и трудоемкость их разработки должны быть весьма незначительны, чтобы можно было максимально запараллелить процесс накопления и отладки знаний (осуществляемый экспертом) с процессом выбора (разработки) программных средств (осуществляемым инженером по знаниям и программистом). Для

удовлетворения указанным требованиям, как правило, при создании прототипа используются разнообразные средства, ускоряющие процесс проектирования.

Прототип должен продемонстрировать пригодность методов инженерии знаний для данного приложения. В случае успеха эксперт с помощью инженера по знаниям расширяет знания прототипа о проблемной области. При неудаче может потребоваться разработка нового прототипа или разработчики могут прийти к выводу о непригодности методов ЭС для данного приложения. По мере увеличения знаний прототип может достигнуть такого состояния, когда он успешно решает все задачи данного приложения. Преобразование прототипа ЭС в конечный продукт обычно приводит к перепрограммированию ЭС на языках низкого уровня, обеспечивающих как увеличение быстродействия ЭС, так и уменьшение требуемой памяти. Трудоемкость и время создания ЭС в значительной степени зависят от типа используемого инструментария.

В ходе работ по созданию ЭС сложилась определенная технология их разработки, включающая шесть следующих этапов (рис. 3.2.3):

- идентификацию;
- концептуализацию;
- формализацию;
- выполнение;
- тестирование;
- опытную эксплуатацию.

На этапе *идентификации* определяются задачи, которые подлежат решению, выявляются цели разработки, определяются эксперты и типы пользователей.

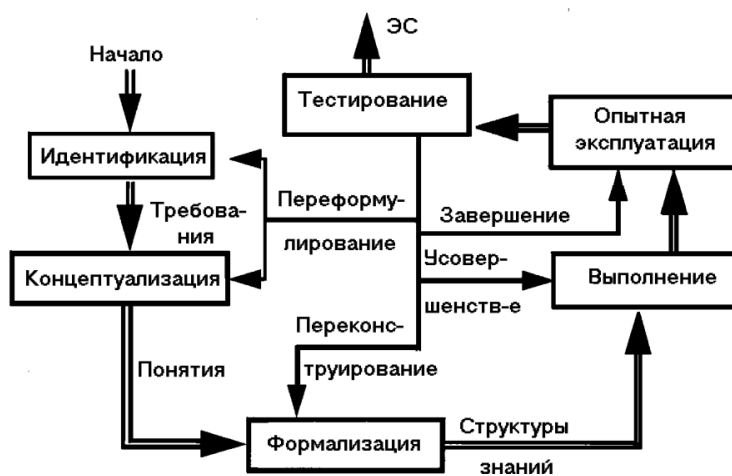


Рисунок 3.2.3 - Технология разработки ЭС

На этапе *концептуализации* проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач.

На этапе *формализации* выбираются ИС и определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, моделируется работа системы, оценивается адекватность целям системы зафиксированных понятий, методов решений, средств представления и манипулирования знаниями.

На этапе *выполнения* осуществляется наполнение экспертом базы знаний. В связи с тем, что основой ЭС являются знания, данный этап является наиболее важным и наиболее трудоемким этапом разработки ЭС. Процесс приобретения

знаний разделяют на извлечение знаний из эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном ЭС. Процесс приобретения знаний осуществляется инженером по знаниям на основе анализа деятельности эксперта по решению реальных задач.

## **7. ПРОДУКЦИОННЫЕ И ЛОГИЧЕСКИЕ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ**

### **7.1. Продукционные модели**

Продукционные модели в последнее время широко используются в системах представления знаний. Первоначально предложенные Постом в 1943 г., они были впервые применены в системах ИИ в 1972 г.

Продукционные модели могут быть реализованы как процедурно, так и декларативно. Их простота и строгая форма послужили основой ряда интересных свойств, что сделало их удобным средством представления знаний. Рядом исследователей отмечалось, что использование продукционных моделей имеет уже само по себе особую психологическую важность, хотя они могут быть с успехом использованы и вне рамок психологического моделирования.

Продукционные модели – это набор, правил вида «условия – действие», где условиями являются утверждения о содержимом некой базы данных, а действия представляют собой процедуры, которые могут изменять содержимое БД.

В продукционных системах можно выделить три основные компоненты:

1. Неструктурированная или структурированная БД.  
2. Некоторое число продукционных правил или просто продукций. Каждая продукция состоит из двух частей:

- условий (антецедент); в этой части определяются некоторые условия, которые должны выполняться в БД для того, чтобы были выполнены соответствующие действия;
- действий (консеквент); эта часть содержит описание действий, которые должны быть совершены над БД в случае выполнения соответствующих условий. В простейших продукционных системах они только определяют, какие элементы следует добавить (или иногда удалить) в БД.

3. Интерпретатор, который последовательно определяет, какие продукции могут быть активированы в зависимости от условий, в них содержащихся; выбирает одно из применимых в данной ситуации правил продукций; выполняет действие из выбранной процедуры.

Продукционные модели в основном находят применение в качестве решателей или механизмов выводов. В БД системы хранятся известные факты о некоторой предметной области. Продукции содержат специфические для данной области знания о том, какие дополнительные факты могут быть допущены, если специфические данные найдены в БД.

Действия продукций могут состоять из активных процедур, которые автоматически производят необходимые операции над содержимым БД (либо подобно «демонам» проверять самих себя на предмет того, выполняются ли их условия активации). В этом случае форма представления знаний является

процедурной, хотя и в весьма ограниченном виде. В последующих итерациях факты, добавленные в БД, могут подключать (активировать) другие продукции и т. д.

В классических продукционных системах БД представляют собой переменную часть системы, в то время как правила и интерпретатор чаще всего не меняются. Будучи реализованы процедурно, классические продукционные модели обладают весьма привлекательным свойством модульности. Поэтому правила могут быть добавлены или удалены без возникновения неожиданных побочных эффектов. Причина этого заключается также в том, что в классических системах вызов процедур осуществляется только в зависимости от состояния данных; процедуры, как правило, не активируются другими процедурами. Поэтому продукционные системы могут быть с большим успехом использованы для областей знаний, о которых располагаем только некоторым набором независимых правил (эвристик), а не четкой теорией, вполне завершенной и последовательной, и где поэтому нет алгоритмов, прямо приводящих к цели.

Продукционные системы все более широко используются для реализации продукционных ЭС.

Как правило, классические продукционные системы не содержат сведений о применении, т. е. знаний о том, например, какие продукции использовать для достижения цели. Это ведет к значительному снижению эффективности их работы: хотя в каждой итерации только одна продукция может быть активирована, должны быть проверены условия всех продукций. Для большого числа правил это может потребовать значительного расхода ресурсов. Более того, последовательность выполнения продукций зависит на каждой итерации от состояния всех переменных системы. В этом случае появляется проблема комбинаторного «взрыва».

Для решения этих проблем предлагались подходы, связанные с методами структурного совершенствования БД и условий в продукциях, что позволило бы повысить эффективность функционирования. Предпринимаются также попытки повлиять на ход управления.

В каждом цикле все правила, условия которых удовлетворены содержимым БД, считаются определенными. Если существует несколько таких правил, то вопрос о том, какое из правил выбрать, решается с помощью какой-либо приемлемой стратегии «разрешения конфликтов» (например, выбирается правило с наивысшим приоритетом из заранее определенного перечня приоритетов). Все действия, связанные с выбранным правилом, выполняются и вызывают соответствующие изменения в БД.

Другая возможность заключается в осуществлении точного контроля последовательности выполнения продукций. В простейшем случае продукции могли бы формировать специальные сигналы в БД, которые подключали бы соответствующие продукции в других циклах. В некоторых системах сами продукции могут активировать или дезактивировать другие продукции и даже влиять на работу интерпретатора.

Рассмотрим более подробно некоторые основные формы представления знаний.

В настоящее время разработано множество моделей представления знаний, используемых для реализации систем, основанных на знаниях, в которых знания представлены с помощью правил вида ЕСЛИ-ТОГДА (явление – реакция,



условие – действие). Систему продукций можно считать наиболее распространенной моделью представления знаний. Примерами реальных систем, основанных на знаниях, в которых в качестве основной модели представления знаний использовалась система продукций, являются EMYCIN, OPS-5, AGE.

Если систему продукций рассматривать как модель представления знаний, то правилам, рассматриваемым с точки зрения человека как средства прямого описания способа логического вывода для решения задач в предметной области, можно придать ясный смысл. При этом отличительной чертой представления знаний с высокой модульностью является простота дополнения, модификации и аннулирования.

Кроме того, со стороны компьютера имеется возможность определения простого и точного механизма использования знаний с высокой однородностью, описанных по одному синтаксису. Эти две отличительные черты, по видимому являются причинами столь широкого распространения метода представлений знаний правилами.

## 7.2. Логические модели представления знаний

### И с ч и с л е н и е п р е д и к а т о в .

Классическим механизмом представления знаний в системах является исчисление предикатов (использовалось уже в 50-х годах в исследованиях по ИИ). В системах, основанных на исчислении предикатов, знания представляются с помощью перевода утверждений об объектах некоторой предметной области в формулы логики предикатов и добавления их как аксиом в систему. Рассмотрим основные положения логики предикатов.

Пусть имеется некоторое множество объектов, н а з ы в а е м ы х п р е д м е т н о й о б л а с т ь ю  $M$ . Знаки, обозначающие элементы этого множества, называют *п р е д м е т н ы м и к о н с т а н т а м и*, а знак, обозначающий произвольный элемент этого множества, – *п р е д м е т н о й п е р е м е н н о й*. Терм – это всякая предметная область или предметная константа.

Если  $f$  – функциональная  $n$ -местная буква, и  $t_1, t_2, \dots, t_n$  – термы, то  $f(t_1, \dots, t_n)$  есть терм. Выражение  $P(x_1, x_2, \dots, x_n)$ , где  $x_i, i=1, n$  – предметные переменные, а  $P$  принимает значения 0 и 1, называется *л о г и ч е с к о й ф у н к ц и е й* или предикатом. Переменные принимают значения из произвольного конечного и бесконечного множества  $M$ .

*П р е д и к а т о м* или логической функцией называется функция от любого числа аргументов, принимающая истинные значения 1 и 0. Если в данном выражении заменить  $x_i$  на  $y_i$ , где  $y_i$  – предметные константы, то получим *э л е м е н т а р н у ю ф о р м у л у*, т.е. предикатные буквы применимы также и к предметным константам. Элементарные формулы иногда называют атомными. Из элементарных формул с помощью логических связок  $\vee$  (или),  $\wedge$  (и),  $\neg$  (отрицание),  $\rightarrow$  (импликация) строят предметные формулы (иногда их называют *п р а в и л ь н о п о с т р о е н н ы м и ф о р м у л а м и* – ППФ). ППФ – один из важных классов выражений в исчислении предикатов. Кроме логических связок в рассмотрение вводят кванторы общности  $\forall$  или существования  $\exists$ .

Если  $P$  – предметная формула, а  $x$  – предметная переменная, то выражения  $\forall x P(x)$  и  $\exists x P(x)$  также считаются предметными формулами. В логике предикатов для компактной записи высказываний типа: “для любого  $x$  истинно  $P(x)$ ” и “существует такое  $x$ , для которого истинно  $P(x)$ ” вводятся две новые дополнительные операции: квантор общности  $\forall$  и квантор существования  $\exists$ . Посредством этих операций приведенные выше

высказывания записываются в виде  $\forall x P(x)$  и  $\exists x P(x)$ . Выражение  $\forall x P(x)$  обозначает высказывание истинное, когда  $P(x)$  истинно при всех  $x \in M$  и ложно в противном случае. Если  $P(x)$  в действительности не зависит от  $x$ , то выражения  $\forall x P(x)$  и  $\exists x P(x)$  обозначают то же, что и  $P(x)$ .

Конкретное вхождение переменной  $x$  в формулу  $P$  называется связанным, если оно либо непосредственно следует за каким-либо квантором, либо содержится в области действия некоторого квантора  $\forall$  или  $\exists$ . Вхождение переменной является свободным, если оно не является связанным. В выражении  $\forall x P(x, y)$   $x$ - связанная,  $y$ - не связанная.

*Связанной переменной* называется переменная, если в  $P$  имеется вхождение этой переменной.

Использование обоих кванторов не является обязательным. Рассмотрим выражение  $\forall x P(x)$ , оно обозначает высказывание « $\forall x P(x)$  ложно», равносильное высказыванию «существует элемент  $x$ , для которого  $P(x)$ » ложно или, что тоже «существует элемент  $x$ , для которого  $P(x)$  истинно». Следовательно,  $\forall x P(x)$  равносильно выражению  $\exists x P(x)$ :

$$\forall x P(x) = \exists x P(x).$$

Под *интерпретацией* предикатных формул понимают конкретизацию предметной области, соответствующей данной предметной формуле, и установке соответствия между символами, входящими в предмет, и элементами (а также функциями и отношениями),

определяемыми в данной предметной области.

### **Вывод на предикатах.**

Выводом системы представления знаний на предикатах являются формулы, выводимые из аксиом с помощью правил вывода. Для организации логического вывода могут использоваться правила.

Для решения конкретной задачи начальное состояние и доступные операторы действий переводятся в формулы исчисления предикатов и добавляются к множеству аксиом. Целевое состояние также выражается формулой и рассматривается как теорема, которая должна быть выведена из аксиом с помощью активного *механизма выводов*.

Определим основные формы логического вывода.

*Индукция* (лат. наведение) – это форма мышления, посредством которой мысль наводится на какое-либо общее правило, общее положение, присущее всем единичным предметам какого-либо класса. Индуктивный логический вывод является перспективным направлением инженерии знаний, здесь не рассматривается.

*Дедукция* (лат. выведение) – такая форма мышления, когда новая мысль выводится чисто логическим путем (т.е. по законам логики) из предшествующих мыслей. Такая последовательность мыслей называется *выводом*, а каждый компонент этого вывода является либо ранее *доказанной мыслью*, либо *аксиомой*, либо *гипотезой*. Последняя

мысль данного вывода называется *заключением*.

Последовательность дедукции определяет “план” того, как достигнуть цели из начального состояния.

Дедукция обычно выполняется с помощью попытки вывести противоречие из получаемого в результате преобразований множества аксиом. Либо: для того, чтобы показать, что некоторое множество ППФ неудовлетворимо, надо доказать, что нет такой интерпретации, при которой каждая из ППФ в этом множестве имеет значение 1 (истинно). Хотя эта задача и кажется трудоемкой, существуют довольно эффективные процедуры ее решения. Для выполнения этих процедур требуется представить ППФ данного множества в специальном удобном виде – в виде *предложений*.

### *Процесс стандартизации.*

Любую ППФ исчисления предикатов можно представить в виде предложения, применяя к ней последовательность простых операций. Задача состоит в том, чтобы показать, как придать произвольной ППФ форму предложения. Этот процесс (преобразования ППФ в форму предложения) состоит из следующих этапов:

1. *Исключение знаков импликации.* В форме предложения в исчислении предикатов явно используются лишь связки  $\vee$  и  $\neg$ . Знак импликации можно исключить в исходном утверждении вместо  $A \rightarrow B$   $\neg A \vee B$ .
2. *Уменьшение области действия знаков отрицания.* Необходимо, чтобы знак отрицания  $\neg$  применялся не более чем к одной предикатной букве.
3. *Стандартизация переменных,* при которой осуществляется переименование переменных с тем, чтобы каждый квантор имел свою переменную. Так, вместо  $(\forall x)\{P(x) \rightarrow (\exists x)Q(x)\}$  следует написать  $(\forall x)\{P(x) \rightarrow (\exists y)Q(y)\}$ .  
В области действия любого квантора переменная, связываемая им, является “немой” переменной. По этому везде в области действия квантора ее можно заменить другой переменной, а это не приведет к изменению значения истинности ППФ. Стандартизация переменных в ППФ означает переименование “немых” переменных, с тем чтобы каждый квантор имел собственную немую переменную.
1. *Исключение кванторов существования.* Рассмотрим ППФ  $(\forall u \exists x)P(x, u)$ , которую можно интерпретировать, например, так: для всех  $u$  существует такой  $x$  (возможно, зависящий от  $u$ ), что  $x$  больше  $u$ . Заметим, что так как квантор существования  $\exists x$  находится внутри области действия квантора общности  $\forall u$ , допускается, что значение  $x$  может зависеть от значения  $u$  в  $x$ . Пусть эта зависимость определяется явно с помощью некоторой функции  $g(u)$ , отражающей каждое значение  $u$  в  $x$ , который «существует». Такая функция называется *функцией Сколема*. Если вместо  $x$ , который “существует”, взять функцию Сколема, то можно исключить квантор существования:  $(\forall u) P(g(u), u)$ . Общее

73

правило исключения из ППФ квантора существования состоит в замене в ней всюду переменной, относящейся к квантору существования, функцией Сколема, аргументами которой служат переменные, относящиеся к тем кванторам общности, области действия которых охватывают области действия исключаемого квантора существования.

Функциональные буквы для функций Сколема должны быть “новыми”, в том смысле, что они не должны совпадать с теми буквами, которые уже имеются в ППФ. Если исключаемый квантор существования не принадлежит области действия ни одного из кванторов всеобщности, то функция Сколема не содержит аргументов, т.е. является построй константой: так ППФ  $(\exists x)P(x)$  превращается в  $P(a)$ , где  $a$  - константа, про которую известно, что она существует. Чтобы исключить все переменные, относящиеся к кванторам существования, надо применить описанную ранее процедуру по очереди к каждой переменной.

2. *Приведение к предваренной нормальной форме* (ПНФ). На этом этапе уже не осталось кванторов существования, а каждый квантор общности имеет свою переменную. Теперь можно перенести все кванторы общности ( $\forall$ ) в начало ППФ и считать областью действия каждого квантора часть ППФ. Про полученную таким образом ППФ говорят, что она имеет *предваренную нормальную форму* (ПНФ). ППФ в ПНФ состоит из цепочки кванторов, называемой *префиксом*, и расположенной за ней формулы, не содержащей кванторов, называемой *матрицей*.

3. *Приведение матрицы к конъюнктивной нормальной форме* (КНФ). Любую матрицу можно представить в виде конъюнкций конечного множества дизъюнкций предикатов и (или) их отрицаний. Говорят, что такая матрица имеет КНФ. Заменить  $A \vee \{B \wedge C\}$  на  $\{A \vee B\} \wedge \{A \vee C\}$ .

4. *Исключение  $x, w, u, x, u$  кванторов общности.* Так как все переменные ППФ должны быть связанными, то все оставшиеся на этом этапе переменные относятся к  $\forall$  (общности). Так как порядок расположения кванторов общности несущественен, то эти кванторы можно явным образом не указывать, условившись, что все переменные в матрице относятся к кванторам общности. Таким образом остается лишь матрица в КНФ.

5. *Исключение связок  $\wedge$* , заменой  $A \wedge B$  двумя ППФ  $A$  и  $B$ . Результатом многократной замены будет конечное множество ППФ, каждая из которых представляет дизъюнкцию атомных формул и (или) их отрицаний. Атомную формулу или ее отрицание называют *литерой* (литералом), а ППФ, состоящую лишь из дизъюнкций литер, *предложением*. Этот процесс называется стандартизацией (т.е. преобразование формул в предложения).

Процесс, демонстрирующий, что некоторое множество  $F$  ППФ неудовлетворимо (невыполнимо), начинается с превращения каждой ППФ из множества  $F$  в предложение. В

74

результате возникает некоторое множество  $S$  предложений. Можно показать, что если множество  $S$  неудовлетворимо, то неудовлетворимо и множество  $F$ .

Далее используется *метод  $\forall \rightarrow \exists$*  (кванторная редукция), т.е. к стандартной форме формулы применяют процедуры поиска опровержения, это делается для удобства реализации поиска, т.е. вместо доказательства общезначимости формулы доказывается, что отрицание формулы противоречиво. Возможно использование *метода резолюций Робинсона*.

Система представления знаний на основе исчисления предикатов имеет ряд достоинств:

1. Они достаточно хорошо исследованы как формальная система.
2. Существуют ясные правила, т.е. результаты операций над БЗ также достаточно ясно определены.

Недостатками являются ограниченная выразимость, так как существует большое число факторов, которые тяжело или даже невозможно выразить средствами исчисления предикатов. В настоящее время аппаратное исчисление предикатов в чистом виде используется редко, однако основой большинства языков логического представления знаний, в частности, языка Пролог.