



Пилипенко Елена Федоровна  
Хашимова Диёра Пахритдиновна

# ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

УЧЕБНОЕ ПОСОБИЕ

Для направления бакалавриата:

5521900 – Информатика и информационные технологии  
(по отраслям)

**МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО  
ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН**



**ТАШКЕНТСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ  
УНИВЕРСИТЕТ**

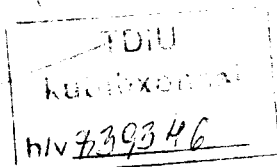
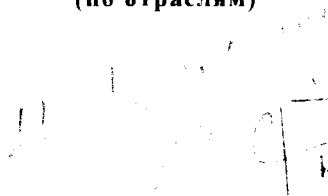
Пилипенко Елена Федоровна  
Хашимова Диёра Пахритдиновна

## **ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ**

**УЧЕБНОЕ ПОСОБИЕ**

Для направления бакалавриата:

**5521900 – Информатика и информационные технологии  
(по отраслям)**



ТАШКЕНТ – 2011

6112.13 (94)  
7-324

Интеллектуальные системы: Учебное пособие. Т.: ТГЭУ, 2011 г.  
- 158 с.

**Авторы:**

Е.Ф. Пилипенко, Д.П. Хашимова

Предлагаемое учебное пособие по дисциплине «Интеллектуальные системы», предназначенное для студентов, обучающихся по направлению образования «Информатика и информационные технологии», содержит теоретические основы проектирования и использования интеллектуальных информационных систем, экспертных систем и баз знаний, а также перспективы их развития.

Кроме того, рассматриваются вопросы организации знаний в базе знаний, а также модели представления знаний.

Печатается по решению Учебно-методического совета при Ташкентском государственном экономическом университете.

**Рецензенты:**

А.Т. Шермухамедов - профессор филиала Российского экономического университета имени Г.В. Плеханова в городе Ташкент, доктор физико-математических наук.

Г.Т. Юлчиева - доцент кафедры «Информационные технологии», кандидат экономических наук

© Е.Ф. Пилипенко, Д.П. Хашимова  
© ТГЭУ, 2011 г.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ В СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА.	9
ГЛАВА 1. ЧЕЛОВЕЧЕСКИЙ И МАШИННЫЙ ИНТЕЛЛЕКТ.	18
1.1. Понятие искусственного интеллекта	18
1.2. Механизмы упрощения и вывода.	19
1.3. Компоненты системы искусственного интеллекта.	21
1.4. Конфигурация системы искусственного интеллекта.	22
ГЛАВА 2. СИСТЕМЫ БАЗ ЗНАНИЙ.	26
2.1. Понятие базы знаний.	26
2.2. Функции баз знаний.	28
2.3. Структуры данных в системах баз знаний	30
2.4. Технология использования баз знаний.	32
ГЛАВА 3. СТРУКТУРИРОВАНИЕ ЗНАНИЙ.	37
3.1. Понятие предметной области и её задачи.	37
3.2. Состав представляемых знаний в базе знаний.	39
3.3. Организация знаний в базе знаний.	42
ГЛАВА 4. МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ.	48
4.1. Логические модели.	48
4.2. Сетевые семантические модели.	51
4.3. Фреймовые модели.	54
4.4. Продукционные модели.	56
ГЛАВА 5. ЭКСПЕРТНЫЕ СИСТЕМЫ.	61
5.1. Общие сведения об экспертных системах.	61
5.2. Организационная структура экспертных систем.	64
5.3. Основные характеристики экспертных систем.	65
ГЛАВА 6. МЕХАНИЗМ ВЫВОДА КАК СРЕДСТВО МАШИННОГО МЫШЛЕНИЯ	69
6.1. Обработка естественного языка в экспертных системах.	69
6.2. Прямая и обратная цепочки рассуждений.	72
6.3. Основные понятия теории вероятностей.	73
6.4. Нечеткая логика в экспертных системах.	76
ГЛАВА 7. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПОСТРОЕНИЯ ЭКСПЕРТНЫХ СИСТЕМ.	81
7.1. Общая характеристика языков программирования.	81
7.2. Процедурные языки программирования.	84
7.3. Функциональные языки программирования.	85
7.4. Логические языки программирования.	89
7.5. Языки интеллектуальных решателей.	92

7.6. Языки представления знаний.	93
<b>ГЛАВА 8. ТЕХНОЛОГИЯ РАЗРАБОТКИ ЭКСПЕРТНЫХ СИСТЕМ</b>	<b>101</b>
8.1. Этапы разработки экспертных систем.	101
8.2. Этап 1: выбор подходящей проблемы.	103
8.3. Этап 2: разработка прототипной системы.	107
8.4. Этап 3: развитие прототипа до промышленной экспертной системы.	112
8.5. Этап 4: оценка системы.	114
8.6. Этап 5: стыковка системы и этап 6: поддержка системы.	115
<b>ГЛАВА 9. КЛАССИФИКАЦИЯ И ИСПОЛЬЗОВАНИЕ ЭКСПЕРТНЫХ СИСТЕМ.</b>	<b>119</b>
9.1. Классификация экспертных систем.	119
9.2. Схема взаимодействия пользователей с экспертной системой	124
9.3. Технология использования экспертных систем.	126
9.4. Области применения экспертных систем.	128
<b>ТЕМА 10. ПЕРСПЕКТИВЫ РАЗВИТИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ЭКОНОМИЧЕСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ.</b>	<b>135</b>
10.1. Направления развития искусственного интеллекта.	135
10.2. Нейросетевые технологии.	141
<b>ГЛОССАРИЙ</b>	<b>148</b>
<b>СПИСОК РЕКОМЕНДОВАННОЙ ЛИТЕРАТУРЫ</b>	<b>152</b>

## МУНДАРИЖА

СУНЪИЙ ИНТЕЛЛЕКТ ТИЗИМЛАРИГА КИРИШ.	9
1-БОБ. ИНСОН ВА МАШИНА ИНТЕЛЛЕКТИ	18
1.1. Сунъий интеллект тушунчаси	18
1.2. Соддалаштириш ва чиқиш механизми.	19
1.3. Сунъий интеллект тизимлари компонентлари.	21
1.4. Сунъий интеллект тизимларининг конфигурацияси	22
2-БОБ. БИЛИМЛАР БАЗАЛАРИ ТИЗИМЛАРИ.	26
2.1. Билимлар базаси тушунчаси.	26
2.2. Билимлар базасининг вазифалари.	28
2.3. Билимлар базаси тизимда маълумотлар структураси.	30
2.4. Билимлар базасидан фойдаланиш технологияси.	32
3-БОБ. БИЛИМЛАРНИНГ ТУЗИЛМАСИНИ ТУЗИШ	37
3.1. Предмет соҳа ва унинг вазифалари	37
3.2. Тақдим этилаётган билимларнинг билимлар базасидаги таркиби.	39
3.3. Билимлар базасида билимларни ташкил этиш.	42
4-БОБ. БИЛИМЛАРНИ ТАҚДИМ ЭТИШ МОДЕЛЛАРИ	48
4.1. Логик моделлар.	48
4.2. Тармоқли семантик моделлар.	51
4.3. Фрейм моделлар.	54
4.4. Продукционные модели.	56
5-БОБ. ЭКСПЕРТ ТИЗИМЛАР	61
5.1. Эксперт тизимлари ҳақида умумий маълумот.	61
5.2. Эксперт тизимларининг ташкилий тузилмаси.	64
5.3. Эксперт тизимларининг асосий тавсифлари.	65
6-БОБ. ЧИҚАРИШ МЕХАНИЗМИ МАШИНА ФИКРЛАНИШИ СИФАТИДА	69
6.1. Эксперт тизимларида табиий тилни қайта ишлаш	69
6.2. Тўғри ва тескари мулоҳазалар занжири	72
6.3. Ихтимоллар назариясининг асосий тушунчалари	73
6.4. Эксперт тизимларида ноаниқ мантиқ.	76
7-БОБ. ЭКСПЕРТ ТИЗИМЛАРНИ ҚУРИШ АСЛАҲАВИЙ ВОСИТАЛАРИ	81
7.1. Дастурлаш тилларининг умумий таърифи.	81
7.2. Процедуралар дастурлаш тиллари.	84
7.3. Вазифавий дастурлаш тиллари.	85
7.4. Мантиқий дастурлаш тиллари.	89
7.5. Интеллектуал ечимларнинг тиллари.	92

7.6. Билимларни тақдим этиш тиллари.	93
<b>8 БОБ. ЭКСПЕРТ ТИЗИМЛАРНИНГ ИШЛАБ ЧИҚИШ ТЕХНОЛОГИЯСИ</b>	<b>101</b>
8.1. Эксперт тизимларнинг ишлаб чиқиш босқичлари	101
8.2. 1-босқич: тегишли муаммони танлаш	103
8.3. 2-босқич: прототип тизимни ишлаб чиқиш	107
8.4. 3-босқич: прототипни саноат эксперт тизимига чиривожланиши	112
8.5. 4-босқич: тизим баҳоланиши	114
8.6. 5-босқич: тизим стыковкаси ва 6-босқич: тизимни қўллаб куватланиши	115
<b>9-БОБ. ЭКСПЕРТ ТИЗИМЛАРИНИНГ ТАСНИФИ ВА ФЙДАЛАНИШИ</b>	<b>119</b>
9.1. Эксперт тизимларининг таснифи	119
9.2. Фойдаланувчининг эксперт тизими билан ўзаро алоқаси чизмаси	124
9.3. Эксперт тизимлардан фойдаланиш технологияси	126
9.4. Эксперт тизимлардан фойдаланиш соҳалари	128
<b>10-БОБ. ИНТЕЛЛЕКТУАЛ АХБОРОТ ТИЗИМЛАРИНИ РИ- ВОЖЛАНТИРИШ ИСТИҚБОЛЛАРИ</b>	<b>135</b>
10.1. Сунъий интеллектни ривожланишининг йўналишлари	135
10.2. Нейротармоқли технологиялар	141
<b>ГЛОССАРИЙ</b>	<b>148</b>
<b>ФЙДАЛАНГАН АДАБИЁТЛАР РЎЙХАТИ</b>	<b>152</b>

# CONTENS

INTRODUCTION TO ARTIFICIAL INTELLECT SYSTEMS	9
CHAPTER 1. HUMAN AND MACHINE INTELLECT	18
1.1. The concept of artificial intellect	18
1.2. The mechanism of simplification and conclusion	19
1.3. The components of the artificial intellect systems	21
1.4. Configuration of the artificial intellect systems	22
CHAPTER 2. SYSTEMS OF KNOWLEDGE BASES	26
2.1. The concept of knowledge bases	26
2.2. The function of knowledge bases	28
2.3. The structures of information in the systems of knowledge base	30
2.4. The operation technology of the knowledge bases	32
CHAPTER 3. KNOWLEDGE STRUCTURING	37
3.1. The concept of the subject sphere 2 its tasks	37
3.2. The structure of present knowledge in the base of knowledge	39
3.3. Knowledge arrangement in the base of knowledge	42
CHAPTER 4. MODELS OF KNOLEDGE REPRESENTATION	48
4.1. Logical models	48
4.2. Web semantic models	51
4.3. Frame models	54
4.4. Productive models	56
CHAPTER 5. EXPERT SYSTEMS	61
5.1. General information of expert systems	61
5.2. Organizational structure of expert systems	64
5.3. The main features of expert systems	65
CHAPTER 6. THE MECHANISM OF CONCLUSION AS THE MEANS OF MACHINE THINKING	69
6.1. The treatment of natural language in the expert systems	69
6.2. Direct and reverse chains of reasoning	72
6.3. General concept of the theory of possibilities	73
6.4. Unclear logics in the expert systems	76
CHAPTER 7. INSTRUMENTAL FACILITIES OF CONSTRUCTING EXPERT SYSTEMS	81
7.1. General features of programming languages	81
7.2. Process languages of programming	84
7.3. Functionial languages of programming	85
7.4. Logical languages of programming	89



7.5. The languages of intellectual determiners	92
7.6. The languages of knowledge representation expert systems	93
<b>CHAPTER 8. WORK OUT TECHNOLOGY OF EXPERT SYSTEMS</b>	<b>101</b>
8.1. Stage of working out the expert systems	101
8.2. Stage 1: choosing the right problems	103
8.3. Stage 2: working out the prototype systems	107
8.4. Stage 3: development of prototype up to industrial expert systems	112
8.5. Stage 4: evaluation of the systems	114
8.6. Stage 5: junction of the system and stage 6: system support	115
<b>CHAPTER 9. CLASSIFICATION AND USING THE EXPERT SYSTEMS</b>	<b>119</b>
9.1. Classification of expert systems	119
9.2. Scheme of users' interaction of with expert systems	124
9.3. The technology of using expert systems	126
9.4. Using spheres of expert systems	128
<b>CHAPTER 10. The perspectives of development of intellectual economic information systems</b>	<b>135</b>
10.1. Directions of development of artificial intellect	135
10.2. Nero-web technologies.	141
<b>GLOSSARY</b>	<b>148</b>
<b>THE LIST OF RECOMMENDED LITERATURES</b>	<b>152</b>

# **ВВЕДЕНИЕ В СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА.**

## **Предмет и содержание дисциплины**

Развитие информационно-коммуникационных технологий (ИКТ) является одним из основных факторов благосостояния и экономического роста страны. Сегодня Информационно-коммуникационные технологии становятся одним из основных приоритетов государственной политики Узбекистана.

В докладе Президента Республики Узбекистан Ислама Каримова на заседании Кабинета Министров, посвященном итогам социально-экономического развития страны в 2010 году и важнейшим приоритетам экономической программы на 2011г. отмечается: «Исключительно важное значение мы придаем дальнейшему развитию современных компьютерных и телекоммуникационных систем и технологий».<sup>1</sup>

В данном аспекте, одним из актуальных направлений выступает интеллектуализация информационных технологий. Это означает, что пользователь, применяя компьютерные технологии, сможет не только получать сведения на основе обработки данных, но и использовать по интересующей его проблеме накопленный опыт и знания профессионалов.

Целью дисциплины «Интеллектуальные информационные системы» является ознакомление студентов с ключевыми концепциями искусственного интеллекта и экспертных систем, формирование у студентов теоретических знаний и практических навыков по проектированию и использованию интеллектуальных информационных

---

<sup>1</sup> Каримов И.А. Все наши устремления и программы – во имя дальнейшего развития родины и повышения благосостояния народа // Народное слово. 22 января 2011 г. – с. 1-3.

систем, основанных на знаниях.

Центральной проблемой при создании интеллектуальных систем является адекватное отображение знаний специалистов в памяти компьютера. Важной целью данной дисциплины является изучение моделей представления знаний и методов организации знаний в базе знаний, формирование у студентов практических навыков по проектированию и использованию баз знаний.

В результате изучения дисциплины студенты должны получить теоретические знания и практические навыки по проектированию, разработке и использованию интеллектуальных информационных систем и баз знаний, уметь определять состав знаний и выбирать соответствующие методы организации и модели представления знаний в базе знаний.

Для успешного изучения дисциплины «Интеллектуальные информационные системы» студенты должны знать следующие дисциплины: «Информатика», «Теория вероятностей», «Информационные технологии и системы», «Экономико-математические методы и модели», «Проектирование баз данных».

Буквально на глазах искусственный интеллект вторгается во все направления компьютерной технологии. Системы искусственного интеллекта появились повсеместно - от большого бизнеса до армии. Искусственный интеллект делает профессиональные возможности человека безграничными. Если он и дальше будет завоевывать рынок с такой скоростью, не за горами то время, когда садиться за компьютер можно будет только имея необходимые познания в этой области.

### **История развития искусственного интеллекта.**

Идея создания искусственного подобия человеческого разума для решения сложных задач и моделирования мыслительной спо-

способности появилась еще в древнейшие времена. Впервые ее выразил Р.Луллий (1235 - 1315), который еще в XIV в. Пытался создать машину для решения различных задач на основе всеобщей классификации понятий.

В XVIII в. Г.Лейбниц (1646 - 1716) и Р.Декарт (1596 - 1650) независимо друг от друга развили эту идею, предложив универсальные языки классификации всех наук. Эти идеи легли в основу теоретических разработок в области создания искусственного интеллекта.

Развитие искусственного интеллекта как научного направления стало возможным только после создания ЭВМ. Это произошло в 40-х гг. XX в. В это же время Н.Винер (1894 - 1964) создал свои основополагающие работы по новой науке – кибернетике[15].

Термин **искусственный интеллект** был предложен в 1956 году на семинаре с аналогичным названием в Станфордском университете (США). Семинар был посвящен разработке логических, а не вычислительных задач. Вскоре после признания искусственного интеллекта самостоятельной отраслью науки произошло разделение на два основных направления: нейрокибернетику и кибернетику «черного ящика». И только в настоящее время стали заметны тенденции к объединению этих частей вновь в единое целое.

Основную идею нейрокибернетики можно сформулировать следующим образом. Единственный объект, способный мыслить, - это человеческий мозг. Поэтому любое «мыслящее» устройство должно каким-то образом воспроизводить его структуру. Таким образом, нейрокибернетика ориентирована на аппаратное моделирование структур, подобных структуре мозга.

В основу кибернетики «черного ящика» лег принцип, противоположный нейрокибернетике. Не имеет значения, как устроено «мыслящее» устройство. Главное, чтобы на заданные входные воз-

действия оно реагировало также, как человеческий мозг. Это направление искусственного интеллекта было ориентировано на поиски алгоритмов решения интеллектуальных задач на существующих моделях компьютеров. Начиная с 60-х гг. велись интенсивные поиски моделей и алгоритма человеческого мышления и разработка первых программ. Оказалось, что ни одна из существующих наук - философия, психология, лингвистика - не может предложить такого алгоритма. Тогда кибернетики предложили создать собственные модели.

Существенный прорыв в практических приложениях искусственного интеллекта произошел в середине 70-х годов, когда на смену поискам универсального алгоритма мышления человека пришла идея моделировать конкретные знания специалистов - экспертов и разрабатывать такие программные средства и системы, в которых важнейшей компонентой являются знания.

Рассмотрим этапы развития систем искусственного интеллекта:

1) 70-е годы характеризуются поиском общих методов решений задач и использованием их при построении универсальных программ.

2) 80-е годы характеризуются поиском общих методов представления информации, ее поиска для использования при создании специальных программ.

3) 90-е годы характеризуются использованием больших объемов высококачественных специальных знаний по некоторой предметной области для создания специальных программ.

В 70-х годах специалисты в области искусственного интеллекта пытались моделировать сложный процесс мышления человека, отыскивая общие методы решения задач и использования этих методов в универсальных программах. Однако разработка таких про-

грамм оказалось слишком трудным делом, так как чем шире класс задач, который может решить одна программа, тем беднее оказывается ее возможности при решении конкретной задачи.

80-е годы усилия программистов были сконцентрированы на разработке методов представления и поиска информации. Методы представления информации - это способы формулирования проблемы или задачи таким образом, чтобы ее можно было решить. А методы поиска - это оригинальные способы управления ходом решения таким образом, чтобы не требовалось слишком большого объема памяти и времени. В конце 80-х годов специалисты по искусственному интеллекту пришли к пониманию того, что эффективность программ при решении задач во многом зависит от знаний, которыми она обладает.

В начале 90-х годов была принята совершенно новая концепция. Суть ее заключается в том, чтобы сделать программу интеллектуально, ее нужно снабдить множеством высококачественных специальных знаний некоторой предметной области. Таким образом, разрабатываемые системы искусственного интеллекта должны иметь хорошо развитую базу знаний. В настоящее время наиболее полное развитие этой концепции получило проектирование экспертных систем.

### **Классификация интеллектуальных информационных систем.**

Интеллектуальные информационные системы базируются на знаниях и подразделяются на следующие типы:

1. интеллектуальные пакеты прикладных программ или решатели задачи (пакет EUREKA)
2. автоматизированные обучающие системы

### 3. экспертные системы[17].

**Интеллектуальные ППП** распространены в областях, связанных с научной и экономической деятельностью. В этих системах в значительной степени упрощены, т.е. интеллектуализированы процессы работы с данными, например, пакет EUREKA позволяет реализовать идею программирования без программиста. Запись условия задачи в этом пакете осуществляется средствами встроенного редактора по правилам математики. В эту группу также входят табличные процессоры, представляющие собой электронные таблицы. В них содержимое каждой «клеточки» может быть любым типом данных. Примером являются пакет Lotus, SuperCalc, Master, Framework. В зависимости выражаемые этими таблицами позволяют строить рабочую модель прикладной ситуации и проводить ее анализ типа, «что произойдет, если...» Например, какие значения примут некоторые переменные, если определенные исходные величины цена, % ссуды и т.д. изменяют свое значение.

**Автоматизированные обучающие системы (АОС)** - это специальный вид интеллектуальных систем, базирующихся на знаниях и получившей широкое распространение с развитием и появлением рынка ПЭВМ, АОС, как правило, один или несколько языков, на которых описываются действия и структуры, используемые для функционирования обучающей системы. АОС ориентированы на работу с двумя группами пользователей. Первая группа - сами авторы АОС. Вторая группа - обучаемые с помощью этой системы. В функции авторов входит разработка базы знаний, заданий, вида и структуры учебного материала, редактирование и его хранение в памяти компьютера.

**Экспертные системы** - это сложные программные комплексы, аккумулирующие знания специалистов в конкретных предметных областях и тиражирующие этот эмпирический опыт для консульта-

ций менее квалифицированных пользователей.

Современной состояние разработок в области экспертных систем можно охарактеризовать как стадию все возрастающего интереса среди широких слоев экономистов, финансистов, преподавателей, инженеров, медиков, психологов, программистов, лингвистов. К сожалению этот интерес имеет пока слабое материальное подкрепление - явная нехватка учебников и специальной литературы, отсутствие символьных процессоров и рабочих станций искусственного интеллекта, ограниченное финансирование исследований в этой области, слабый отечественный рынок программных продуктов для разработки экспертных систем.

Поэтому распространяются «поделки» под экспертные системы в виде многочисленных диалоговых систем и интерактивных пакетов прикладных программ, которые дискредитируют в глазах пользователей это чрезвычайно перспективное направление. Процесс создания экспертной системы требует участия высококвалифицированных специалистов в области искусственного интеллекта.

Современные экспертные системы широко используются для тиражирования опыта и знаний ведущих специалистов практически во всех сферах экономики. Традиционно знания существуют в двух видах - коллективный опыт и личный опыт[26].

Если большая часть в предметной области представлена в виде коллективного опыта (например, высшая математика), это предметная область не нуждается в экспертных системах.

Если в предметной области большая часть знаний является личным опытом специалистов высокого уровня (экспертов), если эти знания по каким либо причинам слабо структурированы, такая предметная область скорее всего нуждается в экспертной системе.

#### **Краткие выводы.**



В данной теме определены цели и задачи дисциплины «Интеллектуальные информационные системы», рассмотрены предмет и содержание дисциплины, дана история развития искусственного интеллекта и представлена классификация интеллектуальных информационных систем с описанием их основных типов.

#### **Ключевые слова:**

Информатика, интеллектуализация, знания, опыт, интеллект, модель, метод, база знаний, технология, мозг, искусственный, устройство, «черный ящик», алгоритм, предметная область, поиск, концепция, пакет прикладных программ, АОС, экспертные системы, электронная таблица.

#### **Вопросы для обсуждения и контроля.**

- 1) Что изучает дисциплина «Интеллектуальные информационные системы»?
- 2) Когда был предложен термин «искусственный интеллект»?
- 3) Что лежит в основе кибернетики «черного ящика»?
- 4) Что положено в основу нейрокибернетики?
- 5) Перечислите этапы развития систем искусственного интеллекта.
- 6) На какие типы подразделяются интеллектуальные системы?
- 7) Какие ППП называются интеллектуальными?
- 8) Что такое автоматизированные обучающие системы (АОС)?
- 9) Что собой представляет экспертная система?

#### **Литература.**

1. Закон Республики Узбекистан «Об информатизации»// «Народное слово». 2004 г., 11-февраля.
2. Каримов И.А. Все наши устремления и программы – во имя даль-

нейшего развития родины и повышения благосостояния народа// Народное слово, 22 января 2011 г. – с. 1-3.

3. Каримов И.А. Все наши устремления и программы – во имя дальнейшего развития родины и повышения благосостояния народа// Народное слово, 22 января 2011 г. – с. 1-3.

4. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления: Учебник. 2-е изд. (+CD).- СПб.: Питер, 2009.-120 с.

5. Информатика: Учебник/ Под ред. Проф. Н.В. Макаровой.3-е изд. -М.: Финансы и статистика, 2006. –768с.

6. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Учебник, М.: Финансы и статистика, 2004 г. - 424с.

7. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н.Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С.- Ташкент, 2003, 135с.

839346

# ГЛАВА 1. ЧЕЛОВЕЧЕСКИЙ И МАШИННЫЙ ИНТЕЛЛЕКТ

- 1.1. Понятие искусственного интеллекта.
- 1.2. Механизмы упрощения и вывода.
- 1.3. Компоненты системы искусственного интеллекта.
- 1.4. Конфигурация системы искусственного интеллекта.

## 1.1. Понятие искусственного интеллекта.

**Искусственный интеллект** – это некая система программных средств, имитирующая на компьютере процесс мышления человека. Для создания такой системы необходимо изучить сам процесс мышления человека, выделить основных шагов этого процесса и дальнейшее их воспроизведение на компьютере. При традиционном программировании программы предназначаются для решения строго определенного класса задач и любое изменение условий задачи или алгоритма влечет за собой необходимость пересмотра программ, что связано с дополнительными затратами времени, финансовых и материальных ресурсов. Искусственный интеллект предполагает «оснащение» компьютера чертами разума. Методы искусственного интеллекта упрощают объединение программ и дают возможность заложить в систему способность самообучению и накоплению новой информации[17].

Человек может накапливать знания, не изменяя способ мышления, не забывая уже известных фактов. Системы искусственного интеллекта разрабатываются аналогично. При этом достигаются высокая независимость отдельных частей программ, подобно блокам памяти человека. Выбирая нужную информацию, человеческий мозг автоматически подключает только те факты, которые относятся к данной проблеме, не используя при этом все доступные ему

знания. В основе человеческой деятельности лежит мышление и целью в данном случае является конечный результат мыслительного процесса. После достижения одной цели ставится и достигается новая цель. Цели можно разбить на локальные и главные. Исходя из этого можно дать определение интеллекту.

Интеллект представляет собой совокупность фактов и способов их применения для достижения определенной цели. А достижение цели – это применение необходимых правил, использование соответствующих фактов.

Рассмотрим пример:

Факт1 . Зажженная плита горячая

Правило1 . ЕСЛИ положить руку на зажженную плиту, ТО можно обжечься.

Факт2. В час пик на улице много машин

Правило2. ЕСЛИ попытаться в час пик перейти улицу, ТО можно попасть под машину.

## **1.2. Механизмы упрощения и вывода.**

Когда человеческий мозг приступает к решению даже самой простой задачи, для выбора нужных действий в его распоряжении имеется огромный объем информации. Например, направляясь на работу, человек выходит из дома и пересекает улицу, но пока он выбирает момент для перехода, его мозг анализирует и интенсивность движения, и скорость транспорта, и расстояние до противоположной стороны. Одновременно с этим мозг обрабатывает информацию, не имеющую прямого отношения к переходу улицы, т.е. погоду, цвет проезжающих машин и т.д. Кроме того, человек думает о том, куда он идет, как скоро он туда придет, с кем должен встретиться и т.п. Но если бы человек прежде чем перейти улицу анализировал все факты, имеющие прямое или косвенное отноше-

ние к цели перейти улицу, то он мог бы простоять несколько лет. Таким образом, в мозгу человека существует сложная система, руководящая выбором правильной реакции на конкретную ситуацию. Такой выбор называется **упрощением**. Механизм упрощения блокирует те факты, которые не имеют отношения к решаемой в данный момент задаче. Этот механизм положен в основу систем искусственного интеллекта[25]. На рисунке 1.1. Приведена схема работы механизма упрощения при выборе необходимых фактов и правил.

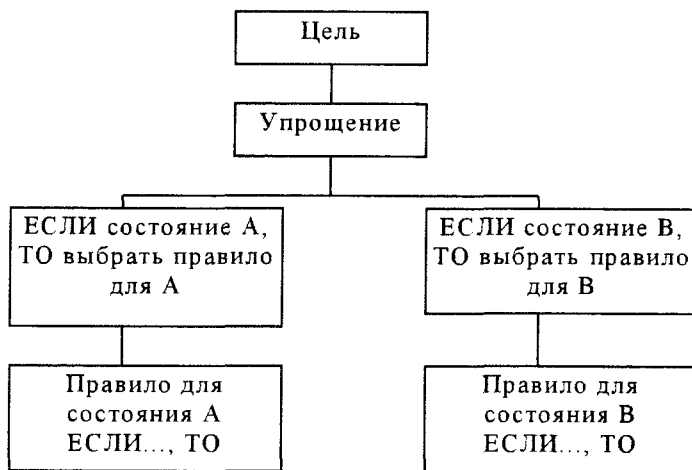


Рис. 1.1. Работа механизма упрощения.

Достигая цель человек не только приходит к решению поставленной перед ним задачи, но одновременно приобретает новые знания.

Например, Джон и Мери – родители Джима. Джон и Мери – родители Джейн. Цель – определить, кем приходится друг другу Джим и Джейн.

Механизм упрощения заставляет человека обратиться к хранящемуся в его мозгу правилу: ЕСЛИ у девочки и мальчика одни и те же родители, ТО мальчик и девочка брат и сестра. В процессе

достижения цели получен новый факт – Джим и Джейн – брат и сестра. Та часть интеллекта, которая помогает извлекать новые факты называется **механизмом вывода**. Именно механизм вывода позволяет человеку учиться на опыте и дает возможность генерировать новые факты из уже существующих, применяя имеющиеся знания к новой ситуации.

### 1.3. Компоненты системы искусственного интеллекта.

Для того, чтобы можно было говорить об искусственном интеллекте программная система должна иметь все элементы, составляющие процесс принятия решения человеком: цели, факты, правила, механизмы вывода и упрощения. Основные компоненты таких систем показаны на рисунке 1.2.

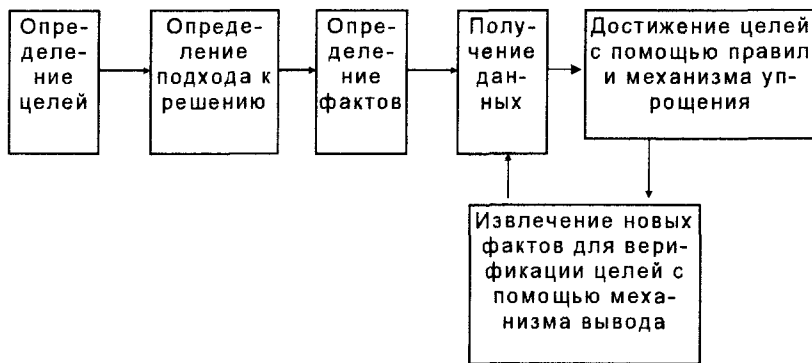


Рис. 1.2. Компоненты системы искусственного интеллекта.

При проектировании систем искусственного интеллекта на первом этапе определяются цели, для достижения которых она предназначена, определяется класс решаемых задач и необходимо уметь описывать их в крупных терминах. Факты являются важнейшей частью системы искусственного интеллекта, без них нельзя достичь цели. У каждой цели свои факты. Каждый факт имеет свой

вес, т.е. каждому факту присуща относительная важность. Чем большее значение имеет факт для решения данной задачи, тем больше его вес. После того, как определены общие факты, необходимые для достижения целей, надо получить конкретные данные. Для получения данных формулируются соответствующие вопросы, ответы на которые помогают системе прийти к окончательному решению. Факты и правила хранятся в компьютере, в так называемой базе знаний[17]. Разработка программы для системы искусственного интеллекта состоит из следующих этапов:

1. Определение целей
2. Определение фактов, имеющих отношение к этим целям.
3. Получение данных, соответствующих фактам, характерным для данной ситуации.
4. Оценка данных с использованием правил и механизма вывода. Известные факты применяются к заданной ситуации в соответствии с правилами. Правила помогают верно оценить данные и достичь цели.

#### **1.4. Конфигурация системы искусственного интеллекта.**

Процесс достижения целей с помощью описанных выше этапов называется **прямой цепочкой** рассуждения, т.е. цепочкой, идущей от данных к логическому заключению. А процесс, в котором заключение используется для поисков подтверждающих его данных, называется **обратной цепочкой**. Когда цель достигнута, необходимо проверить ее правильность, т.е. проанализировать задачу снова с данными и правилами. Этот процесс называется **верификацией целей** [15]. Также примером обратной цепочки может быть выбор механизмов вывода новых данных для подтверждения заключения, который изначально предполагается верным. Таким образом, обратная

цепочка рассуждений идет в сторону, противоположную прямой цепочке, т.е. от заключения к данным. Обратная цепочка возникает только после того, как цель уже достигнута. На рисунке 2.3 представлена конфигурация и работа системы искусственного интеллекта.

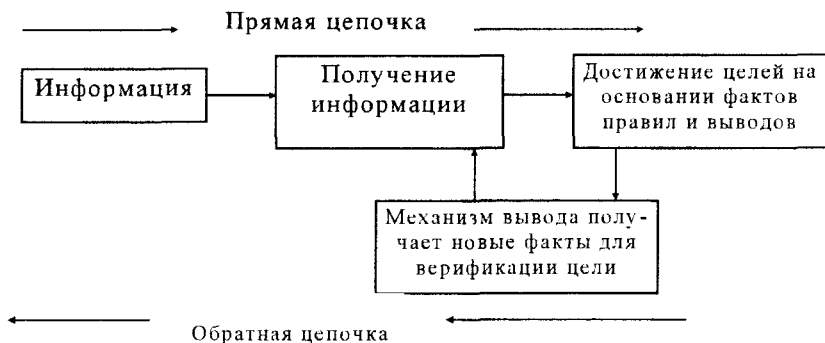


Рис. 1.3. Конфигурация и работа системы искусственного интеллекта.

В мозгу человека механизм упрощения руководит поиском дополнительных правил для верификации целей до тех пор, пока не будут проверены все возможные способы ее достижения. Механизм упрощения системы искусственного интеллекта позволяет компьютеру пропустить или обработать какую-то часть данных из базы знаний в зависимости от их важности для достижения определенной цели. Таким образом, и в мозгу человека, и в системах искусственного интеллекта механизм упрощения просто игнорирует ненужные и не относящиеся к делу рассуждения.



### **Краткие выводы.**

В данной теме даны понятия искусственного интеллекта, механизма упрощения и механизма вывода, определены компоненты систем искусственного интеллекта, рассмотрены процессы прямой и обратной цепочек рассуждений, а также процесса верификации целей. Все указанные понятия рассматриваются в сравнении с процессом мышления человека.

### **Ключевые слова:**

Искусственный интеллект, человеческий мозг, программные средства, разум, факты, правила, цели, механизмы упрощения, механизмы вывода, знания, ситуация, данные, верификация, структура, вес, прямая, обратная, цепочка, рассуждения, логический вывод, заключения, поиск, база знаний.

### **Вопросы для обсуждения и контроля.**

- 1) Что такое искусственный интеллект?
- 2) В чем заключается работа механизма упрощения?
- 3) В чем заключается работа механизма вывода?
- 4) Перечислите основные компоненты систем искусственного интеллекта?
- 5) В чем заключается прямая цепочка рассуждений?
- 6) В чем заключается обратная цепочка рассуждений?
- 7) Какой процесс называется верификацией целей?
- 8) Чем отличается традиционное программирование от создания систем искусственного интеллекта?

### **Литература**

1. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления: Учебник. 2-е изд. (+CD).- СПб.: Питер,

2009.-120 с.

2. Информационные технологии управления. Учебное пособие для вузов /Под ред. проф. ГА Титоренко. — 2-е изд., доп. — М: ЮНИТИ-ДАНА, 2007. - 439 с.

3. Информатика: Учебник/ Под ред. Проф. Н.В. Макаровой.3-е изд. -М.: Финансы и статистика, 2006. -768с.

4. Андрейчеков А.В., Андрейчева О.Н. Интеллектуальные информационные системы. Учебник, М.: Финансы и статистика, 2004 г. - 424с.

5. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н.Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С.- Ташкент, 2003. - 135с.

6. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем.- СПб.: Питер,2001. -384 с.

7. Джексон Питер. Введение в экспертные системы. Пер.с англ. Уч.пос.- М.: Издательский дом «Вильямс» 2001. – 624 с.

## ГЛАВА 2. СИСТЕМЫ БАЗ ЗНАНИЙ.

- 2.1. Понятие базы знаний.
- 2.2. Функции баз знаний.
- 2.3. Структуры данных в системах баз знаний
- 2.4. Технология использования баз знаний.

### 2.1. Понятие базы знаний.

В развитии информационного обеспечения автоматизированных информационных технологий управления экономической деятельностью наибольший интерес представляет применение в области искусственного интеллекта. Одной из форм реализации достижений в этой области является создание экспертных систем – специальных компьютерных систем, базирующихся на системном аккумулировании, обобщении, анализе и оценке знаний высококвалифицированных специалистов – экспертов. В экспертной системе используются базы знаний, в которой представляются знания о конкретной предметной области.

**База знаний** – это совокупность моделей, правил и факторов (данных), порождающих анализ и вывод для нахождения решений сложных задач в некоторой предметной области[19].

Выделенные и организованные в виде отдельных, целостных структур информационного обеспечения знания о предметной области становятся явными и отделяются от других типов знаний, например общих знаний. Базы знаний позволяют выполнять рассуждение не только и не столько на основе формальной (математической логики), но и на основе опыта, фактов, эвристик, то есть они приближены к человеческой логике.

Разработки в области искусственного интеллекта имеют целью

использование больших объемов высококачественных специальных знаний о некоторой предметной области для решения сложных неординарных задач.



Рис. 2.1. Основные свойства базы знаний.

База знаний является основой экспертной системы, она накапливается в процессе ее построения. Знания выражаются в явном виде, позволяющем сделать явным способ мышления и решения задач, и организованы так, чтобы упростить принятие решения. База знаний, обуславливающая компетентность экспертной системы, воплощает в себе знания специалистов учреждений, отдела, опыт группы специалистов и представляет собой институциональные знания (свод квалифицированных, обновляющихся стратегий, методов, решений) (рис.2.1).

Знания и правила работы можно рассматривать в различных аспектах:

- Глубинные и поверхностные;
- Качественные и количественные;
- Приближенные (неопределенные) и точные (определенные);
- Конкретные и общие;
- Описательные и предписывающие.

## 2.2. Функции баз знаний.

Основными компонентами системы БЗ являются:

- \* Сама БЗ и ее окружение;
- \* Механизм получения решений;
- \* Интерфейс;

В настоящее время существует два понятия термина БЗ:

1. БЗ – это пакет определенных специальных знаний, используемых в системе.
2. БЗ – это интегрированная система определенных специальных знаний[25].

Все знания подразделяются на **алгоритмические** (процедурные) и **неалгоритмические**. Неалгоритмические знания подразделяются на **концептуальные** (понятийные) и **фактуальные** (предметные)(рис 2.2.).

В системах БЗ важным является вопрос определения состава знаний, т.е. определение что представлять и как представлять. Вопрос что представлять определяется тем, чтобы решение конкретной задачи отображало моделируемую или представляемую сущность. А вопрос как представлять содержит в себе следующие два вопроса:

1. как организовать знания, т.е. структурировать их.
2. как представить знания в выбранном способе формализации.

Таким образом, круг вопросов, решаемых при представлении знаний в системе БЗ включает следующее:

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

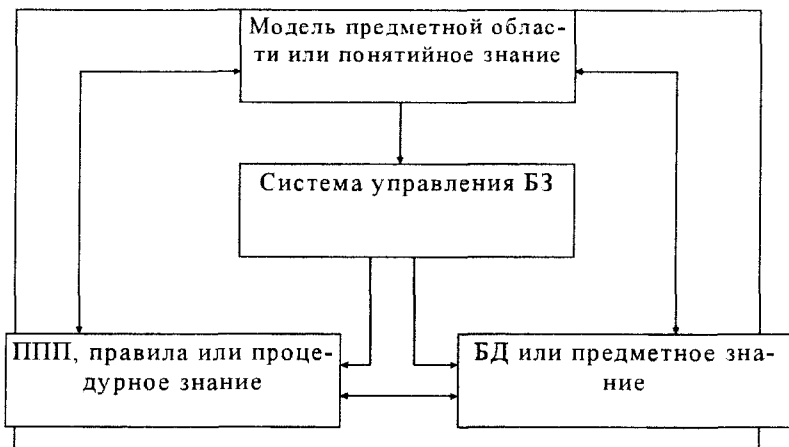


Рис.2.2. Классификация знаний.

При этом нами будут использоваться следующие понятия: предметная область, проблемная область, данные, знания.

**Предметная область** – это область знаний о задачах, а проблемная область включает в себя предметную область и сами задачи, решаемые в этой области. Данные могут быть исходными, промежуточными или окончательными о решаемой в текущий момент задаче и знания – это любая информация, в том числе и конкретные факты, которые хранятся в системе вне зависимости от того, решает система задачу в данный момент времени или нет. Непосредственное использование знаний из БЗ для решения задачи обеспечивается механизмом получения решений. Механизм получения решений дает возможность извлекать из БЗ ответы на вопросы, получать решения задач, формулируемых в терминах, хранящихся в БЗ. Принцип работы механизма получения решений тесно связан со способами представления знаний в БЗ. Для знаний, представленных в БЗ уровнями механизма получения решений является процедура решения уравнений. А для знаний, представленных логическими формулами

или правилами специального вида – этот определенный механизм вывода. Механизм получения решений содержит алгоритм решения, т.е. специальное алгоритмическое знание. С другой стороны механизм получения решений содержит определенную часть семантики знаний в процедурной форме. Это подтверждает относительность границ между механизмом получения решений и БЗ. **Интерфейс** – это часть системы БЗ, которая обеспечивает работу с БЗ, которая обеспечивает работу с БЗ и механизмом получения решений на языке достаточно высокого уровня, приближенного к профессиональному языку специалистов в той предметной области, к которой относится БЗ. В интерфейс включается соответствующий языковой процессор. Кроме того в функции интерфейса входит поддержка диалога с пользователем, что дает возможность пользователю получить объяснения действий системы, участвовать в поиске решений, выполнять и корректировать БЗ.

### **Структуры данных в системах баз знаний**

При создании конкретной экспертной системы для решения поставленных задач в систему необходимо включить ряд структур данных. Эти структуры данных необходимы для работы с базой знаний.

#### **Список логических выводов.**

Список логических выводов – это структура данных, содержащая упорядоченный список всех возможных логических выводов. Список состоит из номера правила и логического вывода, связанного с этим правилом. На каждое правило базы знаний в списке приходится одна запись.

Список логических выводов используется для поиска вывода по номеру правила. Когда все условия части ЕСЛИ истинны, вызывается часть ТО и переменной логического вывода присваивается

значение.

### **Список переменных.**

Список переменных содержит имена переменных для всех условных частей правил базы знаний и признаков их инициализации. Признак инициализации показывает, присвоено ли значение переменной или нет.

Независимо от того, в скольких условиях встречается переменная, в список переменных она включается один раз. Но в этот список нельзя включать переменные логического вывода, так как их значение определяется с помощью правил. Первоначально предполагается, что переменным значения не присвоены и признак инициализации для всех переменных равен NI. До того как правило включается в работу, все переменные, входящие в его условную часть, должны быть проинициализированы. Определить, присвоено ли переменной значение, можно, посмотрев список переменных. Если переменная отмечена NI, то, прежде чем начать работать с правилом, ей надо присвоить определенное значение, и она помечается как I. После этого значение переменной сравнивается с правой частью соответствующего условия, в которое она входит.

### **Список переменных условия.**

Условная часть правила может содержать несколько переменных, максимально до четырех. Если условная часть правила содержит несколько условий и они связаны логическими операторами И, то обратиться к части ГО можно только в том случае, если все переменные в условной части проинициализированы, поэтому необходимо сформировать для всех правил список переменных, входящих в их условные части, чтобы можно было проверить, проинициализирована переменная или нет. Структура данных, содержащая перечень всех переменных для всех условных частей правил, называется списком переменных условия.



Слева от имен переменных указывается индекс элемента массива, в который помещается имя соответствующей переменной. Каким образом 3 описанных структуры соотносятся с мыслительной деятельностью человека. Прежде всего, человек просматривает все возможные пути, способные привести к решению конкретной задачи (это список логических выводов). Затем он выделяет условия, определяющие эти пути. Это списки переменных и переменных условия.

Разработанные структуры данных позволяют быстро обрабатывать информацию, не повторяя одни и те же шаги по несколько раз, так как значения переменных можно использовать в определенной ситуации для различных логических выводов.

#### **Стек логических выводов.**

Стек логических выводов – это главная структура данных, так как в экспертной системе она связывает остальные структуры данных. В стеке логических выводов хранится информация о нужных пользователю логических выводах и о переменных условия, инициализацию которых необходимо проверить. Для того, чтобы выяснить, можно ли вызвать часть ТО, необходимо проверить все условия. Если все условия истинны, то в работу включается часть ТО. Если хотя бы одно из условий не выполняется, то правило отбрасывается. Чтобы сохранить последовательность правил и содержащихся в них условий и используется стек логических выводов.

#### **2.4. Технология использования баз знаний.**

Содержание базы знаний может быть применено пользователем для получения эффективных управленческих решений. На рисунке 2.3. показана структура базы знаний и ее функционирование[14].

Эксперт – это специалист, умеющий находить эффективные

решения в конкретной предметной области.

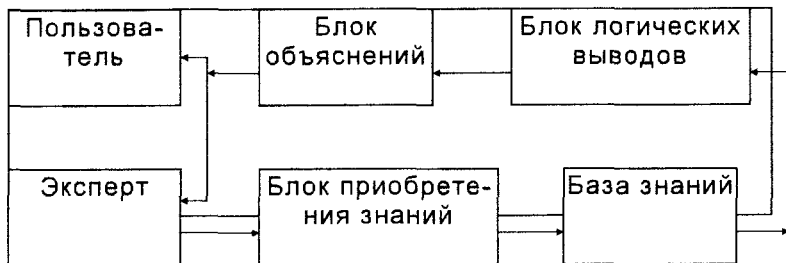


Рис. 2.3. Технология использования базы знаний.

Блок приобретения знаний отражает накопление базы знаний, этап модификаций знаний и данных. База знаний отражает возможность использования высококачественного опыта на уровне мышления квалифицированных специалистов, что делает экспертную систему рентабельной в соответствии с нуждами бизнеса и заказчика.

Блок логических выводов, осуществляя сопоставление правил с фактами, порождает цепочки выводов. При работе с ненадежными данными формируются нечеткая логика, слабые коэффициенты уверенности, низкая степень меры доверия и т. д.

Блок объяснений отражает в технологии использования базы знаний пользователя последовательность шагов, которые привели к тому или иному выводу с возможностью ответа на вопрос “Почему”.

К настоящему времени распространение баз знаний в значительной степени определяется темпами накопления профессиональных знаний.

Та область профессиональной человеческой деятельности, которая пока поддается формализации, а значит, и автоматизации на

базе ЭВМ, - это небольшая часть накопленных человеком знаний. В составе накопленных знаний огромный слой составляют индивидуально накапливаемые неотчуждаемые знания. Меньший объем составляют знания, которые доступны для традиционной передачи. И, наконец, едва различимые в общем объеме всех остальных знаний – это формализованные знания.

Структуризация или формализация знаний основана на различных способах представления знаний. В современных системах самый популярный способ использует факты и правила. Они обеспечивают естественный способ описания процессов в некоторой предметной области.

Правила обеспечивают формальный способ представления рекомендаций, указаний, стратегий. Они подходят в тех случаях, когда предметные знания возникают из опытных (эмпирических) ассоциаций, накопленных за годы работы по решению задач в данной области. Правила чаще всего выражаются в виде утверждений типа: Если... то... .

Описание предметной области в базе знаний предполагает разработку способов представления и организации знаний, методов формулирования, переформирования и решения задач. Понятия (объекты) предметной области представляются с помощью символов. Например, для банковской системы это могут быть: клиенты, фондовый инструмент, операция, задача и т.д.. Между символическими понятиями определяются соотношения, применяются различные стратегии (логические или полученные в результате опыта) для манипулирования понятиями. Представление знаний, их структуризация предполагает выбор понятий, сложных неординарных задач. Поэтому и правила в базе знаний бывают либо сложными, либо множественными и объемными.

Развитие концепции баз знаний связано с исследованиями и

достижениями в области систем искусственного интеллекта. Области применения баз знаний и систем на их основе расширяются. Создается целый спектр баз знаний – от небольших по объему для портативных систем до мощных, предназначенных для профессионалов, эксплуатирующих сложные и дорогие АРМ. Очень большие базы знаний хранятся в централизованных хранилищах, доступ к которым осуществляется через сети пользователями различных систем, уровней масштабов и т.д. Успехи в разработке баз знаний делают их доступными для массового пользователя, что будет способствовать их появлению как актуального коммерческого продукта.

### **Краткие выводы.**

В данной теме дано понятие базы знаний и описаны ее основные свойства, представлена структура базы знаний и ее функции. Даны классификация знаний, понятия предметной и проблемной областей, определено отличие данных от знаний и описана технология использования баз знаний.

### **Ключевые слова:**

База знаний, правила, факты, экспертная система, опыт, прогноз, обучение, институциональные знания, аспекты, механизм получения решений, интерфейс, алгоритмические знания, концептуальные знания, фактуальные знания, состав знаний, структуризация, модель, организация, представление, предметная и проблемная области, данные, механизм вывода, пользователь, эксперт, блок (объяснений, приобретения знаний, логических выводов), формализация.

### **Вопросы для обсуждения и контроля.**

- 1) В чем заключается понятие базы знаний?

- 2) Какими свойствами обладает база знаний?
- 3) Какие основные компоненты содержит система баз знаний?
- 4) Какие вопросы необходимо решить при представлении знаний в базе знаний?
- 5) Чем отличается предметная область от проблемной области?
- 6) Чем отличаются данные от знаний?
- 7) В чем заключается технология использования баз знаний?

### **Литература.**

1. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления: Учебник. 2-е изд. (+CD).- СПб.: Питер, 2009.-120 с.

2. Информационные технологии управления. Учебное пособие для вузов /Под ред. проф. ГА Титоренко. — 2-е изд., доп. — М: ЮНИТИ-ДАНА, 2007. - 439 с.

3. Ивасенко А.Г., А.Ю. Гридасов, В.А. Павленко. Информационные технологии в экономике и управлении. Учебное пособие. М. Кнорус, 2005г.— 160 с.

4. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Учебник, М.: Финансы и статистика, 2004 г. - 424с.

5. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н. Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С.- Ташкент, 2003. – 135с.

6. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем.- СПб.: Питер, 2001. -384 с.

## ГЛАВА 3. СТРУКТУРИРОВАНИЕ ЗНАНИЙ.

- 3.1. Понятие предметной области и её задачи.
- 3.2. Состав представляемых знаний в базе знаний.
- 3.3. Организация знаний в базе знаний.

### 3.1. Понятие предметной области и её задачи.

**Предметная область** характеризуется совокупностью объектов, их характеристик и отношениями между ними. В зависимости от особенности предметной области данные об образующих её сущностях могут быть точными, приблизительными, многозначными, полными или ошибочными.

Предметная область подразделяется на **статическую и динамическую**. Большинство существующих интеллектуальных систем рассматривают предметные области как статические, то есть как области, в которых динамикой можно пренебречь.

Примером статической области, является область, хранящая сведения о симптомах заболеваний и правилах постановки диагноза по этим симптомам. Это такие медицинские системы, как MYCIN, INTERNIST, GASNEN, PIP.

Динамические области подразделяются на следующие типы:

1. Представления, в которых учитываются только местоположение объектов в пространстве, но сами объекты неизменны во времени.
2. Представления, в которых учитываются изменения объектов во времени, но не рассматриваются изменения пространственных отношений между объектами.
3. Представления, в которых учитываются как изменения объектов во времени, так и изменения пространственных взаимоотношений между ними.

Представление первого типа обычно используется в робототехнике, где источником динамики являются только изменения в пространстве. Представления второго типа обычно используются в системах, где подлинные причины изменений неизвестны системе и за источник изменения принимается ход времени. Примером подобной системы является система, учитывающая изменение состояния больного во времени.

**Задачи предметной области** подразделяются на следующие типы:

1. Задача анализа предметной области
2. Задача преобразования предметной области.
3. Задача определения или выбора предметной области[19].

**Задача анализа** - это задача, которая осуществляет доопределение или переформулирование текущего состояния предметной области. В точной статической предметной области можно решать только задачи анализа. Задачи анализа являются простейшими из рассматриваемых задач. В ходе их решения данные только добавляются, но не изменяются и не устраняются.

**Задачи преобразования** - это задачи, при решении которых одно состояние предметной области преобразуется в другое. Правила вывода, используемые при решении этих задач преобразуют предметную область, но не выводят систему за рамки данной предметной области. При этом возникают следующие проблемы: можно ли преобразовать предметную область и как ее преобразовать, чтобы удовлетворить условиям решаемой задачи. При решении преобразования данные попадают и выбывают из рабочей области, так как сведения при переходе в новое состояние должны соответствовать этому новому состоянию предметной области.

**Задачи определения предметной области**- это задачи, в ходе решения которых система переходит из одной альтернативной об-

ласти в другую. При решении задач определения необходимо использовать предположения, устраняющие неполноту информации и позволяющие продолжить процесс поиска решения. Это предположение не должно противоречить точным известным фактам. Нарушения этого соответствия вызывает необходимость перехода в другую альтернативную область.

### **3.2. Состав представляемых знаний в базе знаний.**

Состав знаний в базе знаний (БЗ) определяется следующими вопросами: какие знания должны быть представлены и от чего зависит сам состав знаний.

Состав знаний в БЗ зависит от:

- Проблемной области
- Структуры системы БЗ
- Требований и целей пользователя
- Языка общения.

Рассмотрим обобщенную схему интеллектуальной информационной системы(рис.3.1.)

В соответствие с общей схемой для её функционирования требуются следующие знания:

1. Знания о процессе решения задачи(управляющие знания, используемые интерпретатором)
2. Знания о языке общения и способах организации диалога, используемые лингвистическим процессором
3. Знания о способах представления и модификации знаний, используемые компонентой приобретения знаний
4. Поддерживающие структурные знания, используемые объяснительной компонентой[17].



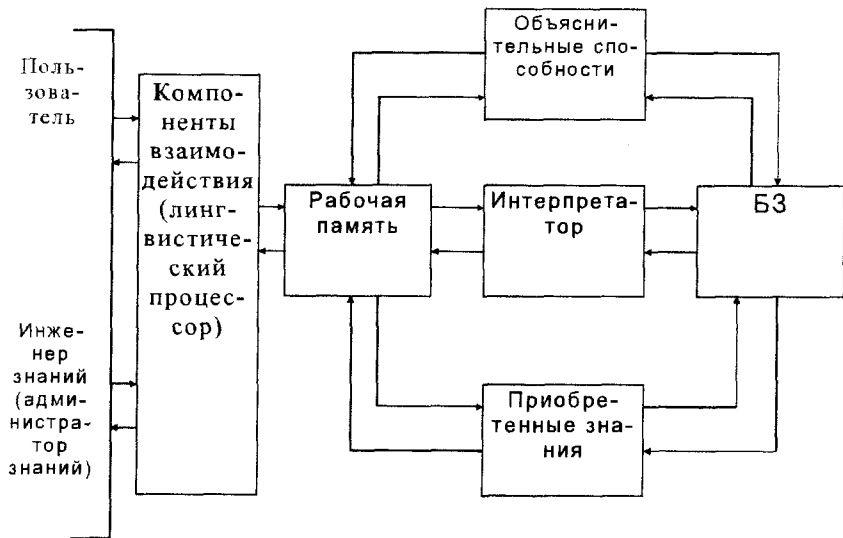


Рис. 3. 1. Обобщенная схема интеллектуальной информационной системы.

Зависимость состава знаний от требований пользователей проявляется в следующем:

1. Какие задачи от общего набора задач и с какими данными хочет решать пользователь?
2. Каковы предпочтительные способы и методы решений?
3. При каких ограничениях на количество результатов и способы их получения должна быть решена задача?
4. Каковы требования к языку общения и к организации диалога?
5. Какова степень общности или конкретности о проблемной области?
6. Каковы цели пользователей?

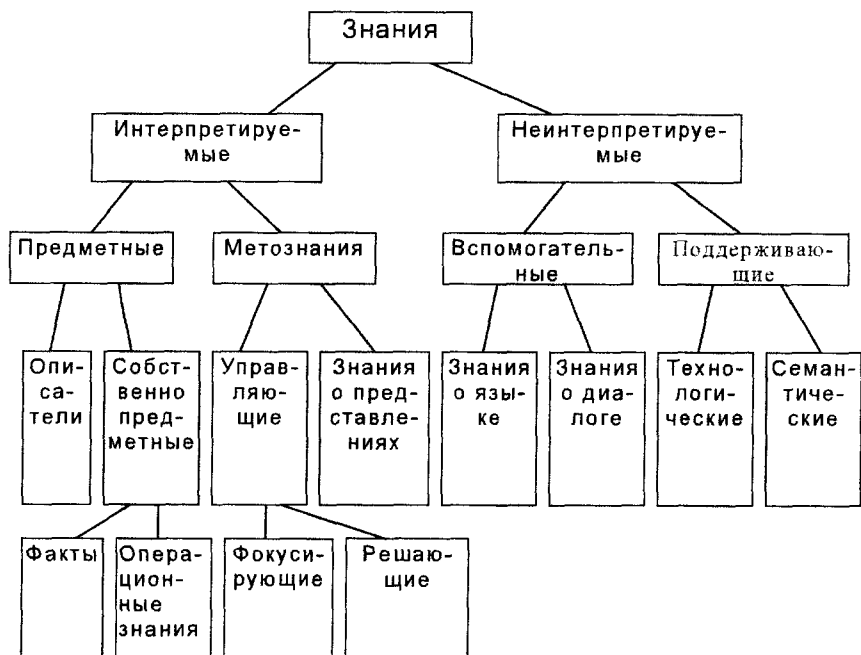


Рис 3.2 Структура знаний в БЗ.

На рисунке 3.2. представлена структура знаний в БЗ. Все знания подразделяются на **интерпретируемые** и **неинтерпретируемые**. Первый тип знаний – это тип, который способен определить или интерпретировать решатель, т. е. интерпретатор. Знания второго типа неизвестны решателю и делятся на **вспомогательные** и **поддерживающие** знания. Вспомогательные знания хранят информацию о лексике и грамматике языка и информацию о структуре диалога. Вспомогательные знания фактически подразделяются на **знания об языке** и **знания о диалоге**. Поддерживающие знания используются при создании системы и при выполнении объяснений. В свою очередь поддерживающие знания делятся на **технологические** и **семантические** знания. Технологические знания содержат сведе-

ния о времени создания знаний и об авторе знаний. Семантические знания – смысловое описание знаний. В их состав входит информация о причинах ввода знаний, о назначении знаний, о способах их использования и получаемом эффекте. Таким образом поддерживающие знания имеют описательный характер[19].

Интерпретируемые знания включают в свой состав **предметные знания, управляющие знания и знания о представлениях**. Последние два вида знаний объединяются под общим названием **метознания**. Знания о представлениях содержат информацию о том, каким образом в системе представлены интерпретируемые знания. Таким образом описатели предметной области содержат определенную информацию о предметных знаниях, например коэффициенты определенности правил, данных, мер важности или сложности. Предметные знания подразделяются на **описательные и собственно предметные знания**. Собственно предметные знания подразделяются на **факты и операционные знания**. Операционные знания содержат информацию о том как можно изменить описание предметной области в ходе решения задач. Это знания, задающие процедуры обработки. Управляющие знания подразделяются на **фокусирующие знания и решающие знания**. Фокусирующие знания описывают, какие знания следует использовать с той или иной ситуации. Обычно фокусирующие знания содержат сведения о наиболее перспективных гипотезах и о знаниях, которые наиболее целесообразно использовать при проверке соответствующих гипотез. Решающие знания используются для выбора стратегии, наиболее эффективной для решения данной задачи.

### **3.3. Организация знаний в базе знаний.**

Под организацией знаний понимается структурирование знаний. Знания могут быть организованы следующим образом:

1. Организация знаний по уровням представления и по уровням детальности

2. Организация знаний в рабочей памяти системы

3. Организация знаний в БЗ[25].

**Уровни представления** могут быть: нулевой уровень- он содержит знания системы о проблемной области; первый уровень содержит метознания, то есть знания о том, как представлены во внутреннем мире системы знания нулевого уровня; второй уровень содержит сведения о знаниях первого уровня, то есть знания о представлении базовых понятий первого уровня.

**Уровни детальности** позволяет рассматривать знания с различной степенью подробности, в основном выделяются три уровня детальности:

1. По общей организации знаний

2. По логической организации знаний

3. По физической организации знаний.

**В рабочей памяти** интеллектуальной системы хранятся определенные данные. В существующих интеллектуальных системах данные в рабочей памяти рассматриваются как изолированные или как связанные. В первом случае рабочая память состоит из множества простых элементов, а во втором из одного или нескольких сложных элементов. В простейшем случае это константы или переменные. Здесь переменными являются характеристики некоторого объекта. А константами – значения соответствующих характеристик в данный момент времени. Для рабочей памяти, состоящей из сложных элементов, связь между отдельными элементами указана явно в виде семантических отношений. Но наиболее важным является рассмотрение вопросов организации предметных и управляющих знаний в БЗ.

Этому способствует ряд причин:

1. Так как это основные знания в БЗ по значимости и объему
2. Эти знания оказывают наибольшее влияние на качество и эффективность работы интеллектуальной системы
3. Вопросы организации не интерпретируемых знаний не являются сложными в виду того, что способы обработки их просты.

Система БЗ должна быть организована так, чтобы в необходимый момент определить и использовать необходимые, то есть **релевантные знания**. В этой связи можно выделить три аспекта организации знаний:

1. Связность знаний
2. Механизм доступа к знаниям
3. Способ сопоставления

**Связность или агрегация знаний** является основным способом, обеспечивающим ускорение поиска релевантных знаний. Знания организуются вокруг наиболее важных объектов или сущностей предметной области, то есть связываются и представляются в виде отдельного блока, тогда при поиске знаний о некотором объекте ищется соответствующий блок и далее поиск осуществляется внутри блока. Связи в блоках бывают внешние и внутренние.

Внутренние связи выражают структуру блока, а внешние связи отражают взаимозависимости между ними(блоками). Внешние связи подразделяются на логические и ассоциативные. Логические связи выражают семантические отношения между элементами знаний.

Основной проблемой при работе с большой БЗ является проблема поиска знаний. Здесь более необходим общий **механизм доступа**. Задача механизма доступа состоит в том, что по некоторому описанию объекта, имеющегося в рабочей памяти, найти в БЗ блоки, удовлетворяющие этому описанию. В этом процессе могут быть выделены два этапа: на первом этапе по ассоциативным связкам осуществляется выбор из БЗ возможных кандидатов на роль нуж-

ных объектов. На втором этапе путем выполнения операций сопоставления возможных кандидатов с описаниями объекта осуществляется окончательный выбор нужных объектов. Организация подобного механизма доступа связана с решением вопросов выбора критерия пригодности определенного кандидата и осуществление работы в конфликтных ситуациях.

**Способ сопоставления** может быть использован для подтверждения и коррекции выбранного объекта. Для подтверждения неизвестного объекта он может быть сопоставлен с некоторыми известными образцами. Операции сопоставления выступают в следующих формах:

1. Синтаксическое сопоставление
2. Параметрическое сопоставление
3. Семантическое сопоставление
4. Принуждаемое сопоставление.

При синтаксическом сопоставлении соотносятся образцы или формы объекта, но не его содержание. При параметрическом сопоставлении вводится параметр, определяющий степень сопоставления. При семантическом сопоставлении соотносятся не формы объектов, а их функции, то есть содержание. Принуждаемое сопоставление характеризуется тем, что один сопоставляемый объект рассматривается с точки зрения другого. В отличие от других форм сопоставления здесь всегда может быть получен положительный результат. Вопрос лишь состоит в том, какова сила принуждения. Принуждение выполняют специальные процедуры, связанные с описаниями объектов. Если эти процедуры не в состоянии осуществить сопоставления, то система сообщает пользователю, что успех может быть достигнут только в том случае, если определенные характеристики рассматриваемых объектов можно считать сопоставляемыми.

### **Краткие выводы.**

В данной теме представлена обобщенная схема интеллектуальной информационной системы, в соответствии с этой схемой дана классификация знаний, определена зависимость состава знаний от требований пользователя, охарактеризована предметная область с точки зрения ее деления на статическую и динамическую, рассмотрены задачи анализа, преобразования, определения или выбора предметной области. Рассмотрены методы организации знаний по уровням представления и уровням детальности, в рабочей памяти системы и непосредственно в базе знаний, выделены основные аспекты организации знаний.

### **Ключевые слова:**

Знания, база знаний, проблемная область, пользователь, инженер знаний, рабочая память, объяснительные способности, интерпретатор, язык общения, приобретенные знания, требования, ограничения, задачи, предметная область, статическая, динамическая, время, пространство, анализ, преобразование, определение, выбор, интерпретируемые, неинтерпретируемые, предметные, метознания, вспомогательные, поддерживающие, диалог, симантические, описатели, факты, фокусирующие, уровни, релевантные, связность, доступ, сопоставление.

### **Вопросы для обсуждения и контроля.**

- 1) От чего зависит состав знаний в базе знаний?
- 2) Какие знания требуются для функционирования ИС?
- 3) В чем проявляется зависимость состава знаний от требований пользователя?
- 4) Что подразумевается под статической предметной областью?

- 5) На какие типы подразделяются динамические предметные области?
- 6) На какие типы подразделяются задачи предметной области?
- 7) Какова структура знаний в базе знаний?
- 8) Каким образом могут быть организованы знания в базе знаний?
- 9) Выделите основные аспекты организации знаний?

### **Литература.**

1. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления: Учебник. 2-е изд. (+CD).- СПб.: Питер, 2009.-120 с.
2. Информационные технологии управления. Учебное пособие для вузов /Под ред. проф. ГА Титоренко. — 2-е изд., доп. — М: ЮНИТИ-ДАНА, 2007. - 439 с.
3. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Учебник, М.: Финансы и статистика, 2004 г. - 424с.
4. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н. Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С.- Ташкент, 2003
5. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных : Учебник для высших учебных заведений /Под ред. проф. А.Д. Хомоненко. изд. втор., перераб. СПб.: КОРОНАПринт, 2002. – 672 с.
6. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем.- СПб.: Питер, 2001. -384 с.

### **ГЛАВА 4. МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ.**

#### 4.1. Логические модели.



4.2. Сетевые семантические модели.

4.3. Фреймовые модели.

4.4. Продукционные модели.

#### 4.1. Логические модели.

Многообразие моделей представления знаний может быть классифицировано на 2 типа:

1) логические модели;

2) эвристические модели[15,17].

В основе **логических моделей** лежит понятие формальной теории. В логических моделях отношения, существующие между отдельными единицами знаний ( фактами) выражается с помощью синтаксических знаний формальной теории (например, исчисление предикатов). В отличие от логических **эвристические модели** имеют разнообразный набор средств, передающих специфические особенности той или иной проблемной области. Вследствие этого эвристические модели превосходят логические и по возможности или способности адекватно отобразить, т.е. представить проблемную область и по эффективности используемого механизма вывода. Эвристические модели бывают сетевые, фреймовые или продукционные. Логические модели используют язык исчисления предикатов. Первому предикату соответствует имя отношения, а термину аргументы- объекты. Все логические выражения, используемые в логике предикатов имеют значение истинно или ложь. Например, рассмотрим выражение «Джон является специалистом по компьютеру». Это выражение может быть представлено следующим образом: является (Джон специалист по компьютеру). Это выражение может быть интерпретировано следующим образом: имеется некоторый объект X, который является специалистом по компьютеру. Тогда используется следующая формула записи: является (X, специалист по

компьютеру).

Выражение «Смит работает на фирме IBM в качестве специалиста» может быть представлено в виде предиката с 3 аргументами: работает (Смит, фирма IBM, специалист).

При работе с логическими моделями необходимо соблюдать следующие правила.

Порядок аргументов должен всегда задаваться в соответствии с интерпретацией предиката, принятой в данной предметной области. Программист принимает решение о фиксированном порядке аргументов и соблюдает его с начала до конца

Предикат может иметь произвольное число аргументов. Отдельные высказывания, состоящие из предиката и связанных с ним аргументов могут объединяться в сложные высказывания с помощью логических связок: И (and, &) ИЛИ (or, V) НЕ (not, ~),  $\rightarrow$  - импликация используется для формулирования правил по форме

ЕСЛИ ..., ТО ...

Рассмотрим несколько примеров:

1) название предиката – *является*

(Смит, специалист по компьютеру)&читает (Смит, литература)

Смит является специалистом по компьютеру И читает литературу.

2) название предиката – *отчитывается*

(Смит Джон)  $\rightarrow$  руководит (Джон, Смит)

ЕСЛИ Смит отчитывается перед Джоном, ТО Джон руководит Смитом.

3) название предиката - *написал*

(Смит, программа) & not работает (программа)  $\rightarrow$  отладить (Смит, программа, вечер) or передать ( программа, программист, следующий день)

ЕСЛИ Смит написал программу И она не работает, ТО Сми

следует отладить программу вечером ИЛИ передать программисту на следующий день.

В высказываниях в качестве аргументов могут быть использованы и переменные. В этом случае для работы с переменными вводится понятие **Квантор**. Квантор бывает двух видов:

- 1) квантор общности  $\forall$
- 2) Квантор существования  $\exists$

$\forall(X)$  означает, что все значения переменной в скобках, относящиеся к некоторой предметной области должны быть истинными.

$\exists(X)$  – только некоторые из значений  $X$  истинные  $\forall$ ,  $\exists$  могут входить один в состав другого. И от порядка вхождения меняется смысл высказывания. Рассмотрим несколько примеров:

$$1) \forall(X) (\text{специалист по ЭВМ } (X) \rightarrow \text{программист } (X))$$

Все специалисты по ЭВМ являются программистами.

$$2) \exists(X) (\text{специалисты по ЭВМ } (X) \rightarrow \text{хорошие программисты } (X))$$

Некоторые специалисты по ЭВМ являются хорошими программистами.

$$3) \forall(X) \exists(Y) (\text{служащий } (X) \rightarrow \text{руководитель } (Y, X))$$

У каждого служащего есть руководитель

$$4) \exists(Y) \forall(\tilde{O}) (\text{служащий } (X) \rightarrow \text{руководитель } (Y, X))$$

Существует некоторое лицо, которое руководит всеми.

Средства исчисления предикатов позволяют не изменяя смысла высказывания преобразовать их форму удобную для обработки на компьютере. В этом случае для преобразования одни логические связки выражают через другие.

Например, сложное высказывание

$N$  ( $N$  специалист по компьютеру (Смит)  $\vee N$  руководитель (Смит)) может быть выражено следующим образом: специалист по ЭВМ (Смит)  $\&$  руководитель (Смит) отчитывается (Джон, Смит)  $\vee$

руководит (Смит, Джон)

или же в следующем виде:

N отчитывается (Джон, Смит) V руководит (Смит, Джон). Таким образом,

$A \& B$  равносильно  $\sim (\sim A \vee \sim B)$

$A \rightarrow B$  равносильно  $\sim A \vee B$ .

#### 4.2. Сетевые семантические модели.

В основе этих моделей лежит понятие сети, вершины и дуги. Сети бывают простые и иерархические, где вершины – это некоторые понятия, сущности, объекты, события, процессы или явления. Отношения между этими сущностями выражаются дугами. Понятиями обычно выступают абстрактные или конкретные объекты, а отношения - это связи типа: «это («is»), «имеет частью», «принадлежит», «любит». Простые сети не имеют внутренней структуры, а в иерархических сетях некоторые вершины обладают внутренней структурой. Характерной особенностью семантических сетей является обязательное наличие трех типов отношений:

- \* класс - элемент класса;
- \* свойство - значение;
- \* пример элемента класса[15,17].

На рисунке 5.1. приведен фрагмент простой семантической сети: Икраму в течение некоторого времени с  $t_1$  по  $t_2$  принадлежит автомобиль «Лада». Он является «собственником» в событии  $V_1$ . Икрам является элементом множества «человек», а его автомобиль элементом множества «автомобиль-ВАЗ». Событие  $V_1$  является элементом множества событий «владеть», которое является подмножеством множества «ситуации». Дуга «Э» означает элемент множества, дуга «П» означает подмножество. Дуга «ПР» означает подмножество различное, непересекающееся. Дуга «ЭР» означает

элемент различный, непересекающийся.

В иерархических семантических сетях предусматривается разделение сети на подсети (пространства), и отношения устанавливаются не только между вершинами, но и между пространствами.

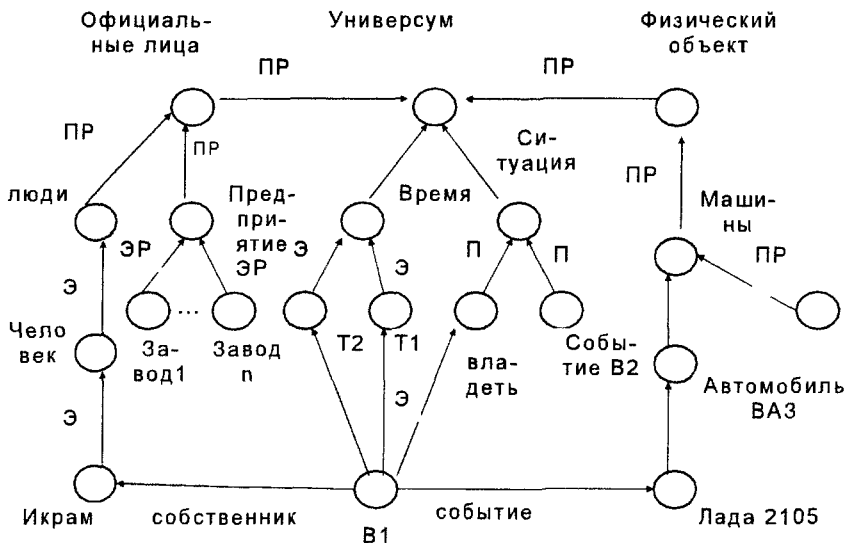


Рис.4.1. Фрагмент простой семантической сети.

На рисунке 4.2 изображено дерево пространств. Для пространства Р6 видимы все вершины пространства, лежащие в пространствах предков Р4, Р2, Р0, а остальные невидимы. Отношение «видимости» даёт возможность сгруппировать пространства в упорядочении множества «перспективы».

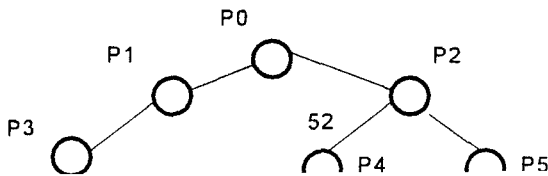


Рис. 4.2. Дерево пространств.

Рассмотрим правила или соглашения графического изображения иерархических сетей:

1) Вершины и дуги, лежащие в одном пространстве, ограничиваются прямоугольником или многоугольником.

2) Дуга принадлежит тому пространству, в котором находится её имя.

3) Пространство  $P_i$ , изображаемое внутри пространства  $P_j$ , считается потомком (внутренним уровнем), т.е. из  $P_i$  “видимо”  $P_j$ .  $P_i$  может быть рассмотрено как супервершина, которая лежит в  $P_j$ .

Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети соответствующего некоторой подсети, соответствующей поставленному вопросу.

Основное преимущество сетевых семантических моделей - в соответствии современным представлениям об организации долговременной памяти человека. Недостаток моделей - сложность поиска вывода на семантической сети.

#### 4.3. Фреймовые модели.

Стремление разработать представления, соединяющие в себе достоинство различных моделей, привело к возникновению фрейм-представлений. Фрейм (англ. Frame - каркас или рамка) - это структура знаний, предназначенная для представления некоторой стандартной ситуации или абстрактного образа [15,17]. С каждым фрейм-

мом связывается следующая информация:

О том, как пользоваться фреймом

Каковы ожидаемые результаты выполнения фрейма

Что делать, если ожидания не оправдались.

Верхние уровни фрейма фиксированы и представляют собой сущности или истинные ситуации, которые описываются данным фреймом. Нижние уровни представлены слотами, которые заполняются информацией при вызове фрейма. Слоты - это незаполненные значения некоторых атрибутов.

**Фреймом** называется также формализованная модель для отображения образа или ситуации.

Структуру фрейма можно представить так:

**ИМЯ ФРЕЙМА:**

(имя 1-го слота: значение 1-го слота),

(имя 2-го слота: значение 2-го слота),

.....

(имя N-го слота: значение N-го слота).

Системы фреймов обычно представляют в виде информационной поисковой сети, которые используются, когда предложенный фрейм не удаётся привести в соответствие с определённой ситуацией, т.е. когда слотам не могут быть присвоены значения, удовлетворяющие условиям, связанным с этими слотами. В подобных ситуациях сеть используется для поиска и предложения другого фрейма.

Различают фреймы-образцы, или прототипы, хранящиеся в базе знаний, и фреймы - экземпляры, которые создаются для отображения реальных ситуаций на основе поступающих данных.

Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире через:

- фреймы - структуры, для обозначения объектов и понятий (заем, залог, вексель);
- фреймы - роли (менеджер, кассир, клиент);
- фреймы - сценарии (банкротство, собрание акционеров);
- фреймы - ситуации (тревога, авария, рабочий режим устройства) и др.

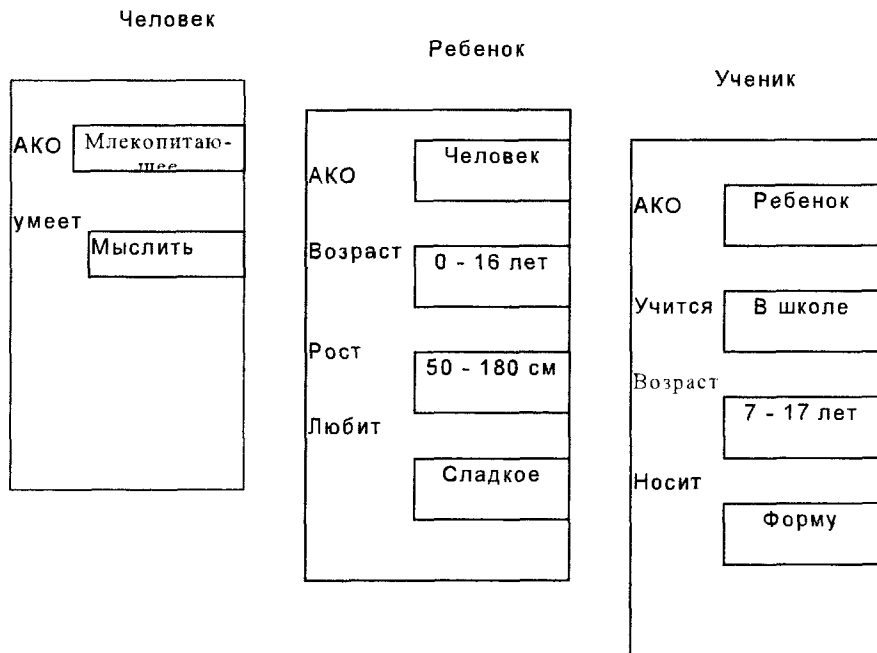


Рис. 5.3. Сеть фреймов.

Важнейшим свойством теории фреймов является заимствованное из теории семантических сетей наследование свойств. И во фреймах, и в семантических сетях наследование происходит по АКО - связям (А - Kind - Of = это). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда не явно наследуются, т.е. переносятся, значения аналогичных слотов.



Например, в сети фреймов, показанной на рисунке 4.3., понятие «ученик» наследует свойства фреймов «ребенок» и «человек», которые находятся на более высоком уровне иерархии.

Так, на вопрос: «Любят ли ученики сладкое?» следует ответ: «Да», так как этим свойствам обладают все дети, что указано во фрейме «ребенок». Наследование свойств может быть частичным, так, возраст для учеников не наследуется из фрейма «ребенок», поскольку указан явно в своем собственном фрейме.

Основным преимуществом фреймов как модели представления знаний является способность отражать концептуальную основу организации памяти человека, а также ее гибкость и наглядность.

#### **4.4. Продукционные модели.**

В традиционном программировании, если  $i$ -ая команда не является командой ветвления, то за ней следует  $i+1$  команда. Подобный способ программирования удобен в тех случаях, когда последовательность обработки мало зависит от обрабатываемых данных.

В противном случае программу лучше рассматривать как совокупность независимых модулей, управляемых образцами. Такая программа на каждом шаге при анализе образцов определяет, какой модуль подходит для обработки данной ситуации. Управляемый образцом модуль состоит из механизма исследования и модификации одной или нескольких структур. Каждый такой модуль реализует определенное продукционное правило. Функции управления при этом осуществляет интерпретатор. С точки зрения представления знаний подход, при котором используются управляемые образцами модули, характеризуется следующими особенностями:

1) разделение постоянных знаний, хранимых в базе знаний, и временных знаний из рабочей памяти.

2) Структурная независимость модулей

3) Отделение схемы управления от модулей, несущих знания о проблемной области.

Это позволяет рассматривать и реализовывать различные схемы управления, облегчает модификацию системы и знаний.

На рисунке 4.4. представлена классификация систем, управляемых образцами.

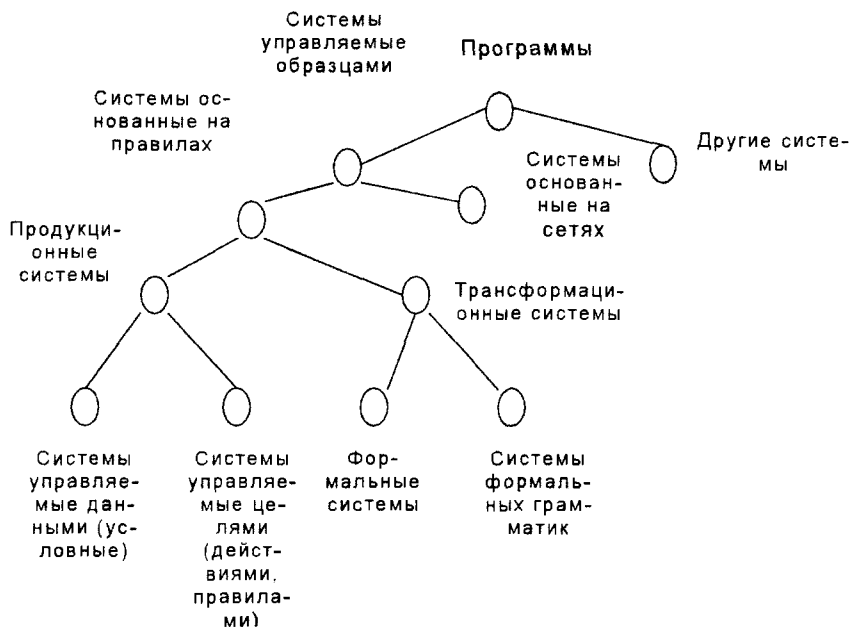


Рис.4.4. Классификация систем, управляемых образцами.

Основой продукционных систем являются правила, в которых сопоставление и управление являются явными функциями системы, зафиксированными в интерпретаторе. Обычно под продукционными системами понимают системы, использующие вывод, управляемый данными. В продукционных системах, управляемых целями, действия являются утверждениями о данных ( правилах), а вывод осуществляется в обратном направлении, т.е. от утверждений, которые должны быть доказаны[19,25].

Представление знаний в виде продукционных правил обладает следующими достоинствами:

- 1) модульность организации знаний
- 2) независимость правил
- 3) простота и естественность модификации знаний
- 4) отделение управляющих знаний от предметных
- 5) возможность создания для ряда приложений управляющих механизмов для автоматизированного решения задач.

### **Краткие выводы.**

В данной теме рассмотрены основные типы моделей представления знаний в базе знаний, а именно логические модели, сетевые, фреймовые и продукционные модели. Для каждого типа моделей представлены примеры с их объяснениями.

### **Ключевые слова:**

Модель, логические, эвристические, предикат, отношение, аргумент, объект, выражение, предметная область, квантор(общности, существования), сеть, вершина, дуга, иерархические, класс, свойство, значение, элемент, пространство, дерево, «видимость», «перспектива», фрейм, ситуация, образ, результат, слот, атрибут, прототип, экземпляр, память человека, продукционные, программирование, модуль, управляемые образцами, правило, постоянный, временный, знания, рабочая память, независимость, проблемная область, цели.

### **Вопросы для обсуждения и контроля.**

- 1) Что лежит в основе логических моделей?
- 2) Чем отличаются логические модели от эвристических моделей?

- 3) Какие правила необходимо соблюдать при работе с логическими моделями?
- 4) Что такое квантор общности?
- 5) Что такое квантор существования?
- 6) Что представляют собой сетевые модели?
- 7) Чем отличается простая сеть от иерархической?
- 8) Что такое фрейм и слот?
- 9) Как можно представить систему фреймов?
- 10) Что отображает фреймовая модель?
- 11) Что представляют собой производственные модели?
- 12) Чем характеризуются управляемые образцами модули?

### **Литература**

1. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления: Учебник. 2-е изд. (+CD).- СПб.: Питер, 2009.-120 с.

2. Информационные технологии управления. Учебное пособие для вузов /Под ред. проф. ГА Титоренко. — 2-е изд., доп. — М: ЮНИТИ-ДАНА, 2007. - 439 с.

3. Информатика: Учебник/ Под ред. Проф. Н.В. Макаровой. 3-е изд. -М.: Финансы и статистика, 2006. —768с.

4. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Учебник, М.: Финансы и статистика, 2004 г. - 424с.

5. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н. Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С.- Ташкент, 2003

6. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных : Учебник для высших учебных заведений /Под ред. проф. А.Д. Хомоненко. изд. втор., перераб. СПб.: КОРОНАПринт, 2002. — 672

с.

7. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем.- СПб.: Питер,2001. -384 с.

## ГЛАВА 5. ЭКСПЕРТНЫЕ СИСТЕМЫ.

- 5.1. Общие сведения об экспертных системах.
- 5.2. Организационная структура экспертных систем.
- 5.3. Основные характеристики экспертных систем.

### 5.1. Общие сведения об экспертных системах.

Система искусственного интеллекта, созданная для решения задач в конкретной предметной области называется **экспертной системой**. Источником знаний для экспертных систем служат эксперты в соответствующей предметной области. Основные свойства экспертных систем:

- Экспертная система для решения задач применяет высококачественные опыт и знания
- Знания в экспертных системах постоянно накапливаются и обновляются
- Экспертная система обладает прогностическими возможностями
- Экспертные системы могут быть использованы для обучения руководящих работников и специалистов[26].

В процессе проектирования и функционирования экспертных систем можно выделить следующих участников(рис.5.1):

1. Разработчик инструментальных средств проектирования экспертной системы
2. Инструментальные средства построения экспертной системы
3. Сама экспертная система
4. Эксперт

5. Инженер знаний или администратор БЗ

6. Пользователь

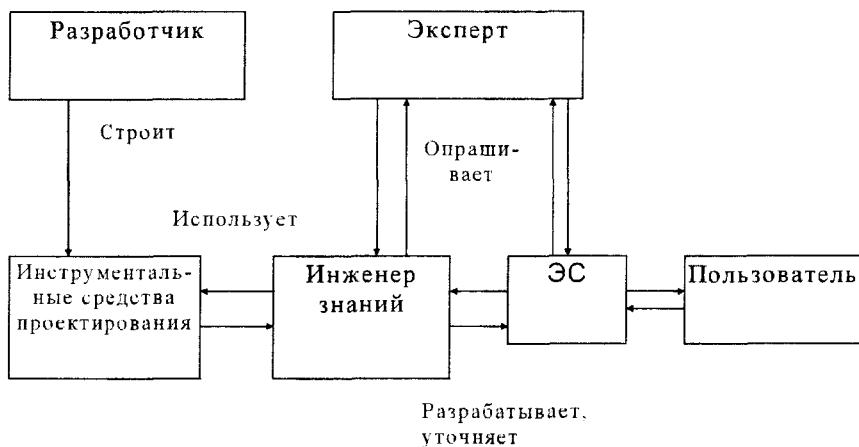


Рис.5.1. Взаимодействие участников экспертной системы.

**Инженер знаний** – это человек, имеющий навыки в разработке искусственного интеллекта и знающий как надо строить экспертную систему. Он опрашивает эксперта, организует знания в БЗ.

К **инструментальным средствам** проектирования относят язык построения экспертной системы и поддерживающие средства, через которые пользователь взаимодействует с экспертной системой. Взаимодействие этих участников можно представить в виде схемы

В таблице 5.1. рассмотрена компетентность экспертных систем. В ней сравнивается система человеческого интеллекта и система искусственного интеллекта.

Компетентность систем человеческого и искусственного  
интеллектов.

Система человеческого интеллекта	Система искусственного интеллекта
<p><b>Недостатки:</b></p> <ol style="list-style-type: none"> <li>1. Непрочная</li> <li>2. Труднопередаваемая</li> <li>3. Труднодокументируемая</li> <li>4. Непредсказуемая</li> <li>5. Дорогая</li> </ol>	<p><b>Преимущества:</b></p> <ol style="list-style-type: none"> <li>1. Постоянная</li> <li>2. Легкопередаваемая</li> <li>3. Легкодокументируемая</li> <li>4. Устойчивая</li> <li>5. Приемлемая по затратам</li> </ol>
<p><b>Преимущества:</b></p> <ol style="list-style-type: none"> <li>1. Творческая</li> <li>2. Приспосабливающаяся</li> <li>3. Использует чувственное восприятие</li> <li>4. Разносторонняя</li> <li>5. Использует широко доступные знания</li> </ol>	<p><b>Недостатки:</b></p> <ol style="list-style-type: none"> <li>1. Искусственно запрограммированная</li> <li>2. Нуждается в подсказке</li> <li>3. Использует символьное восприятие</li> <li>4. Узконаправленная</li> <li>5. Использует специальные знания</li> </ol>

Анализируя преимущества и недостатки этих систем, можно сделать вывод, что человек-эксперт все-таки необходим, так как во многих областях он превосходит искусственный интеллект, например, по творчеству, изобретательности, способности передавать информацию и вообще по здравому смыслу.

## 5.2. Организационная структура экспертных систем.



В основе экспертных систем лежат базы знаний. Знания здесь принимают форму взаимосвязанных фактов и правил, которые могут быть либо истинные, либо ложные, либо с некоторой степенью достоверности. Многие правила в экспертных системах являются **эвристиками**, т.е. эмпирическими, или опытными правилами, или упрощениями.

Экспертная система вынуждена использовать эвристики потому, что решаемые задачи здесь часто трудны и до конца непонятны, не поддаются строгому математическому описанию.

Выделенные знания о предметной области называются базами знаний, а общие знания о нахождении решения задачи называются механизмом вывода. Таким образом, БЗ в экспертной системе содержит факты и правила, а механизм вывода содержит интерпретатор, определяющий каким образом применять правила для вывода новых знаний и диспетчер, устанавливающий порядок применения этих правил[19,26]. Организационная структура экспертной системы представлена на рисунке 5.2.

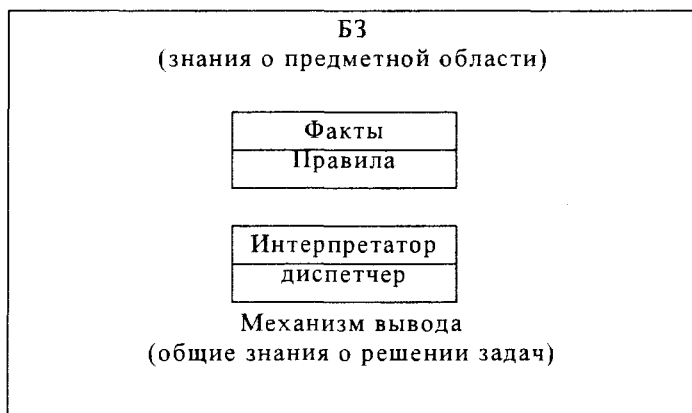


Рис.5.2. Организационная структура экспертной системы.

Многие языки высокого уровня (PROLOG, EMYCIN), предназначенные для построения экспертных систем, имеют механизм вы-

вода, встроенный в язык. А языки более низкого уровня (LISP) требуют специальной разработки механизма вывода. Разработчику экспертной системы легче работать, если механизм вывода встроен в язык. Но у него меньше возможности определить способы организации знаний и доступа к ним. Поэтому разработчик должен внимательно рассматривать вопрос о пригодности предлагаемой схемы управления процессом поиска знаний для данной области[24].

В языках более низкого уровня, у которых нет механизма вывода разработчику нужно затрачивать больше усилий для решения задачи, но при этом он строит схему управления процессом решения по своему усмотрению. В настоящее время появились такие инструментальные средства, в которые встроен механизм вывода, но имеется возможность его модификации (HEARSAY 3).

### 5.3. Основные характеристики экспертных систем.

Главным отличием экспертных систем от традиционных программ является то, что экспертная система манипулирует знаниями, а программа - данными. Экспертная система должна обладать следующими свойствами:

- \* компетентностью;
- \* символьным рассуждением;
- \* глубиной;
- \* Самосознание[25,26].

**Компетентность** характеризуется следующими параметрами:

1. Должно быть достигнуто экспертное управление системой;
2. Система должна быть умелой, т.е. получать результат;
3. Система должна иметь адекватную робастность, т.е. стремиться достигнуть профессионального уровня эксперта - человека;

Робастность подразумевает не только широкое, но и глубокое понимание предметной области. Для этого система должна уметь

рассуждать исходя из фундаментальных знаний, если наборы правил не полные или не корректные.

**Символьное рассуждение** характеризуется следующими параметрами:

Представление знаний в символьном виде. Символ - это строка знаков, соответствующая некоторому понятию реального мира. Символы могут быть объединены для того, чтобы выразить отношения между ними. Такое объединение символьной структурой.

Переформулирование символьных структур.

**Глубина** характеризуется следующими параметрами:

1. Система должна уметь работать в предметной области, содержащей сложные задачи.

2. Система должна использовать сложные правила и иметь глубокие знания о данной предметной области.

**Самосознание** характеризуется следующими параметрами:

1. Система должна уметь исследовать свои рассуждения.

2. Система должна уметь объяснять свои действия. Механизм объяснения должен включать демонстрацию цепочки выводов, т.е. объяснять почему были использованы те или иные правила.

Самосознание экспертной системы важно, так как в этом случае пользователь больше доверяет результатам. Развитие системы более ускоренное, так как ее легче модифицировать. Все предположения, положенные в основу системы, становятся явными и легче выявлять влияние изменений на работу системы.

### **Краткие выводы.**

В данной теме дано определение экспертной системы и рассмотрены ее основные свойства, определены участники проектирования и функционирования экспертных систем, приведена схема взаимодействия участников экспертной системы, дана сравнительная характеристика человеческого и искусственного интеллектов,

представлена организационная структура и основные характеристики экспертной системы.

### **Ключевые слова.**

Экспертная система, предметная область, знания, эксперты, свойства, опыт, прогноз, обучение, разработчик, инструментальные средства, инженер знаний, пользователь, компетентность, интеллект(искусственный, человеческий), база знаний, факты, правила, эвристики, интерпретатор, диспетчер, механизм вывода, языки программирования, высокий уровень, низкий уровень, символьное рассуждение, глубина, самосознание.

### **Вопросы для обсуждения и контроля.**

- 1) Дайте определение экспертной системы?
- 2) Какими основными свойствами обладают экспертные системы?
- 3) Каких участников можно выделить в процессе проектирования экспертных систем?
- 4) Дайте сравнительную характеристику человеческого и искусственного интеллектов?
- 5) Что лежит в основе экспертных систем?
- 6) Как можно представить организационную структуру экспертной системы?
- 7) Какими характеристиками должна обладать экспертная система?

### **Литература.**

1. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления: Учебник. 2-е изд. (+CD).- СПб.: Питер, 2009.-120 с.

2. Информационные технологии управления. Учебное пособие для вузов /Под ред. проф. ГА Титоренко. — 2-е изд., доп. — М: ЮНИТИ-ДАНА, 2007. - 439 с.
3. Балдин К.В., Уткин В.Б. Информационные системы в экономике. Учебник. – М.: Издательство-торговая корпорация «Дашков и К<sup>о</sup>», 2006. – 395 с.
4. Информатика: Учебник/ Под ред. Проф. Н.В. Макаровой.3-е изд. -М.: Финансы и статистика, 2006. –768с.
5. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Учебник, М.: Финансы и статистика, 2004 г. - 424с.
6. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н. Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С. - Ташкент, 2003. – 135 с.
7. Джексон Питер. Введение в экспертные системы. Пер.с англ. Уч. пос. - М.: Издательский дом «Вильямс» 2001. – 624 с.

## **ГЛАВА 6. МЕХАНИЗМ ВЫВОДА КАК СРЕДСТВО МАШИННОГО МЫШЛЕНИЯ.**

- 6.1. Обработка естественного языка в экспертных системах.
- 6.2. Прямая и обратная цепочки рассуждений.
- 6.3. Основные понятия теории вероятностей.
- 6.4. Нечеткая логика в экспертных системах.

### **6.1. Обработка естественного языка в экспертных системах.**

Общение человека и компьютера - это задача, над которой работает множество исследователей. Конечная цель заключается в том, чтобы пользователи могли разговаривать с компьютером на естественном языке, и компьютер отвечал им на том же языке. Компьютеры умны ровно на столько, насколько такими их сделали люди. Поскольку сами по себе они не умеют думать, необходимо дать им очень точные инструкции о том, какие действия они должны выполнять вплоть до мельчайших шагов. Чтобы компьютер понимал язык нужно не только разбить его на основные элементы и ввести эту информацию в компьютер. Сначала надо разработать такие программы, которые смогут воспринимать эту информацию. Таким образом, для того чтобы общение человека с компьютером стало возможным нужно создать систему обработки естественного языка. Например, представим себе, что есть робот (то есть экспертная система, обладающая искусственным интеллектом), которая умеет чинить автомобили, и ему была дана команда "Почини машину около дома со спущенной покрышкой". Человек здесь замечает неточность сразу и в уме ее исправляет. Так как очевидно, что спущенная покрышка у машины, а не у дома. Роботу же нужно знать нечто большее, чем только значения слов и их связь друг с другом, иначе ему придется искать дом со спущенной покрышкой. Поэтому важно

понимать, что правила человеческого языка имеют смысл только для людей, а экспертной системе, чтобы понимать, о чем мы говорим, необходимы специальные правила. Кроме того, искусственный интеллект должен уметь анализировать предложение в их связи с другими предложениями. Возьмем для примера такие два предложения:

1. Джон пьет молоко.
2. Затем он надевает пальто.

Слово «он» во втором предложении относится к слову «Джон» в первом предложении. Без первого предложения второе не имело бы смысла. Все естественные языки являются контекстуальными языками. Иначе говоря, чтобы полностью понять второе предложение, необходимо знать первое, а это и есть контекстуальная зависимость. Языки, в которых интерпретация предложения может быть выполнена без знания других предложений, называются контекстуально независимыми, т.е. безразличными к последовательности событий и действий. Для того чтобы заставить компьютер понимать человеческую речь необходимо построить анализатор естественного языка. Таким образом, экспертная система должна выполнять следующие функции анализа естественного языка:

1. Лексический анализ - то есть анализ слова.
2. Синтаксический анализ - то есть анализ порядка слов в предложении с учетом грамматики.
3. Семантический анализ – то есть анализ значения предложения самого по себе и в его связи с другими предложениями[25].

Для того чтобы научить компьютер воспринимать человеческий язык, необходимо, прежде всего, научить его выполнять разбор предложения. Надо перевести правила грамматики и синтаксиса в форму, которую компьютер мог бы понять. Обычно предложение П состоит из группы существительных и группы глаголов:

П → ГС, ГГ.

Группа существительного может быть разбита на определение и существительное:

ГС → О, С.

Группа глагола может быть разбита на глагол, за которым следует дополнение, выраженное существительным:

ГД → Г, ГС.

И группа существительного может быть представлена в виде существительного:

ГС → С.

Графически синтаксическая структура предложения может быть представлена в виде дерева (рис 6.1).

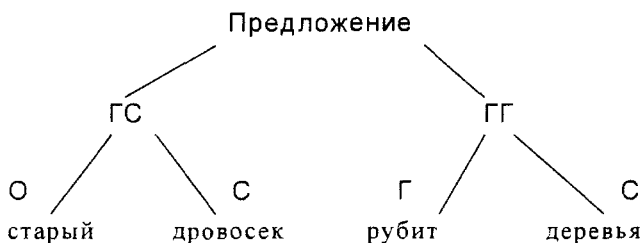


Рис.6.1. Синтаксическая структура предложения.

Рассмотрим предложение: "Старый дровосек рубит деревья".

Предложение разбивается на слова, а слова классифицируются по типу. Разбив предложение на составные части, компьютер проводит его семантический анализ, т. е. пытается понять его смысл. В системах искусственного интеллекта применяется ряд правил, позволяющих компьютеру понять смысл предложения:

1) ЕСЛИ определение стоит на первом месте и за ним идёт существительное, ТО существительное является подлежащим;

2) ЕСЛИ за подлежащим идёт глагол, ТО этот глагол является



сказуемым и поясняет, что делает подлежащее;

3) ЕСЛИ за подлежащим идёт сказуемое, а за ним следует существительное, ТО это существительное является дополнением;

4) ЕСЛИ предложение имеет специальный порядок слов: подлежащее, глагол, дополнение, ТО вся фраза говорит о том, что подлежащее делает по отношению к дополнению.

По существу, обработка естественного языка может освободить пользователя от необходимости изучать сложные языки программирования. Если удастся создать программу, которые позволят компьютеру и пользователю общаться на естественном языке, то будет сделан крупнейший шаг на пути создания подлинно «интеллектуального компьютера».

## 6.2. Прямая и обратная цепочки рассуждений.

В экспертных системах механизм вывода позволяет на основе имеющихся фактов и правил выводить новые факты. В работе механизма вывода в зависимости от направления хода рассуждений и предметной области различают:

- 1) прямую цепочку рассуждений;
- 2) обратную цепочку рассуждений;
- 3) нечёткую логику, построенную на основе вероятности [17].

Сформулируем задачу в общем виде:

Имеет место ситуация, требуется предсказать её последствия, например, во время движения у автомобиля перегрелся двигатель.

Можно разработать следующие правила:

ЕСЛИ двигатель перегрелся, ТО мотор заглохнет; ЕСЛИ мотор заглохнет, ТО это приведёт к определённым денежным затратам и позднему возвращению домой. В данном случае используется прямая цепочка рассуждений. Последовательность рассуждений называется **прямой цепочкой** потому, что отправной точкой рассужде-

ний служит возникшая ситуация ( перегрев двигателя), а часть правила выполняется только в том случае, если удовлетворяется условная часть правила ЕСЛИ. Т.е. сначала имеет место ситуация, а потом делаются выводы. В **обратной цепочке**, наоборот, выводы известны, а необходимо найти причины, например, автомобиль не трогается с места. В данном случае используются следующие правила:

1) **ЕСЛИ** автомобиль не заводится и сел аккумулятор, ТО не подаётся ток в стартер;

2) **ЕСЛИ** в стартер не подаётся ток, ТО автомобиль не тронется с места.

Известный результат ( автомобиль не трогается с места ) влечёт за собой цепочку рассуждений, которая приведёт нас к вызвавшим его причинам. Эта цепочка и называется обратной. Причины возникают раньше следствий, поэтому в процессе обратной цепочки просматриваются логические выводы, устанавливаются условия, которые к ним привели, и определяется, связаны ли эти условия с предыдущими логическими выводами.

Таким образом, обратная цепочка рассуждений всегда начинается со следствия, т.е. части ТО. Она начинается с уже происшедшего события и идёт к его истокам[26].

Система, реализующая прямую цепочку рассуждений на основании имеющихся условий, делает возможные логические выводы, а система, реализующая обратную цепочку рассуждений, по имеющимся выводам ищет необходимые для них условия.

### **6.3.Основные понятия теории вероятностей.**

Во многих эвристических правилах, т.е. правилах, сформулированных экспертом, лежит вероятность появления определенного

события, вычислить которую может только эксперт. Другими словами эксперт делает обоснованные предположения в своей проблемной области. В действительности это означает, что существуют статистические данные, позволяющие делать какие-либо предположения.

Это могут быть, например, медицинские диагнозы, которые врач ставит на основании своих наблюдений над пациентом. Опыт врача во многих случаях с большой точностью позволяет определить заболевание пациента. Конечно, есть вероятность, что врач ошибся, поэтому часто рассматриваются и другие диагнозы.

Компания, занимающаяся добычей нефти, может на основе данных, полученных в результате проведенных исследований, сделать вывод, что найдено место рождение нефти и вынести решение - бурить скважину. Но может оказаться, что в скважине нефти нет.

Байес разработал вероятностную методику, основанную на утверждении, что какое-то событие произойдет, потому что раньше уже произошло какое-то другое событие. Теория Байеса входит в теорию вероятностей в раздел, названный «Условная вероятность».

В экспертных системах широко применяются статистические решения, опирающиеся на теорию Байеса, использование которой описано ниже[24].

Теория вероятностей изучает случайные события. Очень часто человек, сам того не замечая, высказывает предположение или делает вывод пользуясь терминологией теории вероятностей. Например, «Я на 80% уверен, что все в порядке».

Вероятность можно определить следующим образом.

Число экспериментов, исходом которых является  
какое-то событие

$$P = \frac{\text{Число экспериментов, исходом которых является какое-то событие}}{\text{Общее число экспериментов}}$$

Байес занимался разработкой теории условной вероятности.

Условная вероятность – это вероятность наступления какого-то события  $S$  при условии, что уже наступило какое-то другое событие  $e$ . Условная вероятность обозначается  $P(S/e)$ . Вероятность наступления двух событий определяется следующим образом:

$$P(e \text{ и } S) = P(S/e) * P(e).$$

Уравнение читается так: вероятность того, что произойдет два события  $e$  и  $S$ , причем  $e$  произойдет первым равна вероятности наступления события  $S$ , если известно, что произошло событие  $e$ , умноженной на вероятность появления события  $e$ . Рассмотрим пример использования этого уравнения. Предположим, что из набора букв  $ИОО$  случайным образом выбирается либо буква  $I$ , либо  $O$ . Применяя уравнение условной вероятности можно вычислить вероятность того, что в двух попытках сначала попадается буква  $O$ , а потом  $I$ . Поставив в уравнение  $O$  и  $I$ , получим:

$$P(O \text{ и } I) = P(I/O) * P(O)$$

Вероятность выбора буквы  $O$ , т.е.  $P(O)$  равна  $2/4$ , так как в наборе букв две буквы  $I$  и две буквы  $O$ . Вероятность того, что после буквы  $O$  будет выбрана  $I$ ,  $P(I,O) = 2/3$ . Чтобы понять, почему это так, рассмотрим случай, когда буква  $O$  уже выбрана и остались три буквы  $ИО$ . Вероятность выбора буквы  $I$  составляет  $2/3$ , поскольку

из трех оставшихся букв две  $I$ . Вероятность  $P(I \text{ и } O)$  равна :

$$P(I O) * P(O) = 2/3 * 2/4 = 1/3$$

В экспертных системах используется еще одно уравнение условной вероятности:  $P(S) = P(S/e) * P(e) + P(S/\text{not } e) * P(\text{not } e)$ .

Уравнение читается так: вероятность появления события  $S$ ,  $P(S)$ , равняется вероятности появления события  $S$  при условии появления события  $e$ ,  $P(S/e)$ , умноженной на вероятность появления события  $e$ ,  $P(e)$ , плюс вероятность появления события  $S$  при условии, что событие  $e$  не произошло,  $P(S/\text{not } e)$ , умноженная на вероятность, что событие  $e$  не произошло,  $P(\text{not } e)$ .

#### 6.4. Нечеткая логика в экспертных системах.

Рассмотрим другой аспект теории вероятностей. Не всегда можно описать событие с помощью точно определенных правил. Например, можно сказать, что у человека легкое недомогание, если температура больше 37 градусов, но меньше 38. А при большей температуре заболевание может оказаться серьезным. Люди не всегда могут ответить на вопросы точно. Можно ли узнать какая у человека температура, если он говорит, что слегка заболел? Скорее всего нет.

Такие слова как высокий, горячий и легкий представляют собой лингвистические переменные, которые нельзя определить одним значением. Лингвистическая переменная может принимать различные значения из некоторого интервала, границы которого могут меняться в зависимости от обстоятельств. Например, границы интервала для лингвистической переменной «холодный» могут меняться в зависимости от того, идет ли речь о зиме или весне. Ис-

пользование этих понятий при формулировании правил называется **нечеткой логикой**.

Применяя лингвистические переменные, можно вычислить значения некоторых вероятностей, не обременяя пользователя лишними вопросами. Для этого необходимо несколько конкретизировать лингвистические переменные. Пользователю экспертной системы нужно позволить добавлять к этим переменным определения, например «маленький» или «средний», но экспертная система должна точно знать, что под этим подразумевается [24,25].

В нечеткой логике используются коэффициенты уверенности (КУ). Поскольку эвристические правила ЕСЛИ - ТО основываются исключительно на человеческом опыте, с полной определенностью никогда нельзя сказать, что они верны. Пользователь экспертной системы также не может полностью уверен, что значения, которые он присваивает переменным, абсолютно корректны. Например, правило:

ЕСЛИ процентные ставки = падают и налоги уменьшаются,

ТО уровень цен на бирже = растет

верно не всегда, поэтому можно приписать ему значение некоторого уверенности. КУ может иметь значение от -1 до 1. Отрицательное значение КУ показывает степень уверенности в том, что правило не верно, а положительное значение - что правило верно. Таким образом, КУ, равный +1, указывает на полную уверенность в том, что правило верно, а -1 - на полную уверенность в некорректности правила. Конечно, правила, для которых КУ равно -1, рассматривать нет смысла.

Пусть приведенное правило имеет КУ, равный 0,9, и нельзя утверждать, что процентные ставки падают, т.е. первому условию правила назначен КУ, равный 0,6. Кроме того, допустим, что налоги колеблются (то увеличиваются, то уменьшаются), поэтому пред-

положить уменьшение налогов можно, только если КУ равен 0,8.

Тогда правило можно записать так:

ЕСЛИ процентные ставки = падают (КУ =0,6) И

налоги уменьшаются (КУ =0,8),

ТО уровень цен на бирже = растет (КУ правила = 0,9)

Коэффициент уверенности, что уровень цен на бирже будет расти может быть подсчитан следующим образом: выбирается минимальный КУ для условий части ЕСЛИ правила, и умножается на КУ для всего правила для приведенного примера:

$$(\min(0.6, 0.8)) * 0.9 = 0.6 * 0.9 = 0.54$$

Следовательно, при КУ = 0.54 можно сказать, что уровень цен на бирже будет падать.

Во многих случаях изначально заданы граничные значения коэффициента уверенности. Логический вывод считается верным только в том случае, если его КУ превышает заранее заданные граничные значения. Работа с базой знаний продолжается до тех пор, пока значение коэффициента уверенности логического вывода больше граничного значения. В процессе работы выполняются определенные вычисления. Предположим, для частного логического вывода КУ равно 0,4. Это значение запоминается. Затем оно сравнивается с граничным значением КУ (допустим, что оно равно 0,8). Запомненное значение оказалось меньше граничного, и, значит, работа с базой знаний продолжается. Если при работе с базой знаний встретился тот же самый логический вывод, КУ для нового правила умножается на 1 минус значение запомненного ранее КУ и результат прибавляется к запомненному ранее КУ. Значение КУ, равное 1, свидетельствует об абсолютной уверенности в правильности вывода. Затем вновь запомненное значение КУ сравнивается с граничным и если оно больше, выполняется логический вывод, в противном случае, работа с базой знаний продолжается. Вышесказанное

можно записать следующим образом:

Запомненный КУ = Ранее запомненный КУ +  
(1 - Ранее запомненный КУ) \* КУ нового правила

Например:

Граничное значение КУ = 0,8

Правило: ЕСЛИ А, ТО В(КУ = 0,6)

Запомненный КУ: 0,6

Новое правило: ЕСЛИ С, ТО В(КУ = 0,7)

Запомненный КУ =  $0,6 + (1 - 0,6) * 0,7 = 0,88$  (граничные значения превышены, и выполняется вывод).

### **Краткие выводы.**

В данной теме рассмотрены проблемы обработки естественного языка в экспертных системах и функции, выполняемые экспертной системой при анализе естественного языка. Рассмотрен механизм вывода как средство машинного мышления в виде прямой цепочки рассуждения, обратной цепочки. Кроме того, рассмотрены основные понятия теории вероятностей, дано понятие условной вероятности и уравнение ее расчета. Рассмотрен такой аспект теории вероятностей, как нечеткая логика, применяемый в экспертных системах, дано понятие коэффициента уверенности и приведены формулы его расчета.

### **Ключевые слова:**

Естественный язык, компьютер, анализ (лексический, синтаксический, семантический), слово, предложение, цепочка(прямая, обратная), теория вероятностей, эвристические правила, условная вероятность, событие, уравнение, условие, нечеткая логика, лингвистические переменные, коэффициент уверенности, человеческий опыт, правила, граничные значения, логический вывод.



### **Вопросы для обсуждения и контроля.**

1. Какие проблемы возникают в процессе обработки естественного языка в экспертных системах?
2. Что является средством машинного мышления в экспертных системах?
3. Как осуществляется прямая и обратная цепочки рассуждений?
4. Как теория вероятностей применяется в экспертных системах?
5. Приведите уравнения вычисления условной вероятности.
6. Что такое нечеткая логика?
7. Какие переменные называются лингвистическими?
8. Что показывает коэффициент уверенности?
9. Как рассчитывается коэффициент уверенности?

### **Литература.**

1. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления: Учебник. 2-е изд. (+CD).- СПб.: Питер, 2009.-120 с.
2. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Учебник, М.: Финансы и статистика, 2004 г. - 424с.
3. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н. Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С.- Ташкент, 2003. – 135 с.
4. Джексон Питер. Введение в экспертные системы. Пер.с англ. Уч.пос.- М.: Издательский дом «Вильямс» 2001. – 624 с.
5. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем.- СПб.: Питер, 2001. -384 с.

## ГЛАВА 7. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПОСТРОЕНИЯ ЭКСПЕРТНЫХ СИСТЕМ.

- 7.1. Общая характеристика языков программирования.
- 7.2. Процедурные языки программирования.
- 7.3. Функциональные языки программирования.
- 7.4. Логические языки программирования.
- 7.5. Языки интеллектуальных решателей.
- 7.6. Языки представления знаний.

### 7.1. Общая характеристика языков программирования.

Все языки программирования можно разделить на процедурные и декларативные языки. Подавляющее большинство используемых в настоящее время языков программирования (Си, Паскаль, Бейсик и т. п.) относятся к процедурным языкам. Наиболее существенными классами декларативных языков являются функциональные и логические языки (рисунок 7.1.) [15,24].

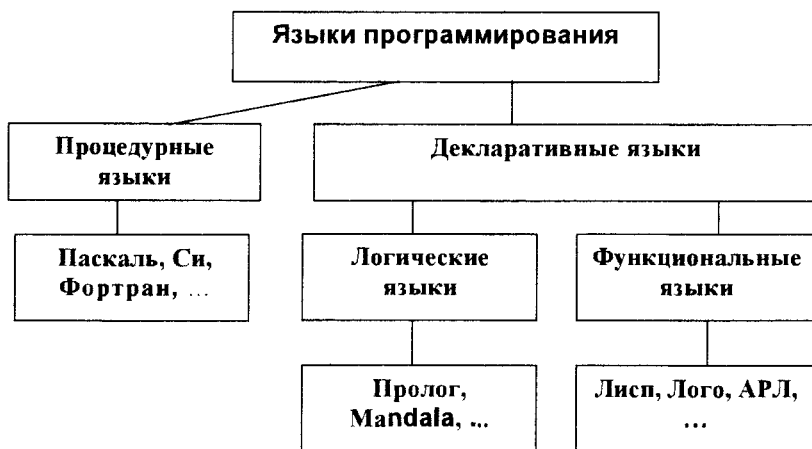


Рис. 7.1. Классификация языков программирования

На практике языки программирования не являются чисто процедурными, функциональными или логическими, а содержат в себе черты языков различных типов. На процедурном языке часто можно написать функциональную программу или ее часть и наоборот. Может точнее было бы вместо типа языка говорить о стиле или методе программирования. Естественно различные языки поддерживают разные стили в разной степени.

Процедурная программа состоит из последовательности операторов и предложений, управляющих последовательностью их выполнения. Типичными операторами являются операторы присваивания и передачи управления, операторы ввода-вывода и специальные предложения для организации циклов. Из них можно составлять фрагменты программ и подпрограммы. В основе процедурного программирования лежат взятие значения какой-то переменной, совершение над ним действия и сохранения нового значения с помощью оператора присваивания, и так до тех пор, пока не будет получено желаемое окончательное значение.

Логическое программирование - это один из подходов к информатике, при котором в качестве языка высокого уровня используется логика предикатов первого порядка в форме фраз Хорна. Логика предикатов первого порядка - это универсальный абстрактный язык предназначенный для представления знаний и для решения задач. Его можно рассматривать как общую теорию отношений. Логическое программирование базируется на подмножестве логики предикатов первого порядка, при этом оно одинаково широко с ней по сфере охвата. Логическое программирование дает возможность программисту описывать ситуацию при помощи формул логики предикатов, а затем, для выполнения выводов из этих формул, применить автоматический решатель задач (т. е. некоторую процедуру). При использовании языка логического программирования основное внимание уделяется описанию структуры прикладной задачи, а не выработке предписаний компьютеру о том, что ему следует делать. Другие

понятия информатики из таких областей, как теория реляционных баз данных, программная инженерия и представление знаний, также можно описать (и, следовательно, реализовать) с помощью логических программ.

Функциональная программа состоит из совокупности определений функций. Функции, в свою очередь, представляют собой вызовы других функций и предложений, управляющих последовательностью вызовов. Вычисления начинаются с вызова некоторой функции, которая в свою очередь вызывает функции, входящие в ее определение и т. д. в соответствии с иерархией определений и структурой условных предложений. Функции часто либо прямо, либо опосредованно вызывают сами себя.

Каждый вызов возвращает некоторое значение в вызывавшую его функцию, вычисление которой после этого продолжается; этот процесс повторяется до тех пор пока запустившая вычисления функция не вернет конечный результат пользователю.

«Чистое» функциональное программирование не признает присваиваний и передач управления. Разветвление вычислений основано на механизме обработки аргументов условного предложения. Повторные вычисления осуществляются через рекурсию, являющуюся основным средством функционального программирования.

На первом этапе развития искусственного интеллекта(ИИ) (в конце 50-х - начале 60-х годов) не существовало языков и систем, ориентированных специально на области знаний. Появившиеся к тому времени универсальные языки программирования казались подходящим инструментом для создания любых (в том числе и интеллектуальных) систем, поскольку в этих языках можно выделить декларативную и процедурную компоненты. Казалось, что на этой базе могут быть интерпретированы любые модели и системы представления знаний. Но сложность и трудоемкость таких интерпретаций оказались настолько велики, что прикладные системы для реализации были недоступны.

Исследования показали, что производительность труда программиста остается постоянной независимо от уровня инструментального языка, на котором он работает, а соотношение между длиной исходной и результирующей программ примерно 1:10. Таким образом, использование адекватного инструментального языка повышает производительность труда разработчика системы на порядок, и это при одноступенчатой трансляции. Языки предназначенные для программирования интеллектуальных систем содержат иерархические (многоуровневые) трансляторы и увеличивают производительность труда в 100-ни раз. Все это подтверждает важность использования адекватных инструментальных средств.

## **7.2. Процедурные языки программирования.**

В эту группу инструментальных средств входят традиционные языки программирования (Паскаль, С, С++, Basic, SmallTalk, Fortran и т.д.), ориентированные в основном на численные алгоритмы и слабо подходящие для работы с символьными и логическими данными. Поэтому создание систем искусственного интеллекта на основе этих языков требует большой работы программистов.

Однако большим достоинством этих языков является высокая эффективность, связанная с их близостью к традиционной машинной архитектуре. Кроме того, использование традиционных языков программирования позволяет включать интеллектуальные подсистемы (например, интегрированные экспертные системы) в крупные программные комплексы общего назначения.

Среди традиционных языков наиболее удобными считаются объектно-ориентированные (Delphi, SmallTalk, С++). Это связано с тем, что парадигма объектно-ориентированного программирования тесно связана с фреймовой моделью представления знаний. Кроме того, традиционные языки программирования используются для создания других классов инструментальных средств искусственного

интеллекта.

### 7.3. Функциональные языки программирования.

Рассмотрим краткую характеристику функциональных языков программирования. На рисунке 7.2. приведены примеры функциональных языков программирования.



Рис 7.2. Примеры функциональных языков программирования.

Язык Лисп был разработан в Стэнфорде под руководством Дж. Маккарти в начале 60-х годов. По первоначальным замыслам он должен был включать наряду со всеми возможностями Фортрана средства работы с матрицами, указателями и структурами из указателей и т. п. Но для такого проекта не хватило средств. Окончательно сформированные принципы положенные в основу языка Лисп: использование единого спискового представления для программ и данных; применение выражений для определения функций; скобочный синтаксис языка.

Лисп является языком низкого уровня, его можно рассматривать как ассемблер, ориентированный на работу со списковыми структурами. Поэтому на протяжении всего существования языка было много попыток его усовершенствования за счет введения дополнительных базисных примитивов и управляющих структур. Но все эти изменения, как правило, не становились самостоятельными языками. В новых своих редакциях Лисп быстро усваивал все ценные изобретения своих конкурентов.

После создания в начале 70-х годов мощных Лисп-систем Маклисп Интерлисп попытки создания языков ИИ, отличных от Лиспа, но на той же основе, сходят на нет. Дальнейшее развитие языка идет, с одной стороны, по пути его стандартизации (Стандарт-Лисп, Франц-Лисп, Коммон Лисп), а с другой - в направлении создания концептуально новых языков для представления и манипулирования знаниями в Лисп среде. В настоящее время Лисп реализован на всех классах ЭВМ, начиная с ПЭВМ и кончая высоко производительными вычислительными системами.

Современные диалекты языка Лисп можно рассматривать как мощные интерактивные системы программирования. Это объясняется двумя причинами. Во-первых, сам язык Лисп претерпевает серьезные изменения - развиваются средства языка, предназначенные для обработки нетрадиционных для Лиспа типов данных: массивов, векторов, матриц; появляются некоторые средства управления памятью (пакеты),

отсутствующие в Лиспе. Серьезные изменения претерпевают и управляющие структуры. Появились несвойственные природе языка Лисп функции, заимствованные из Фортрана, Алгола, Паскаля, Си: Do , Loop, Goto, Case и прочие, позволяющие пользователю, незнакомому с принципами функциональных языков, легко переходить на Лисп. Качество программ снижается, зато возрастает популярность языка. Во-вторых, если на первом этапе развития Лисп-системам была присуща небольшая скорость обработки данных и серьезные ограничения на емкость используемой оперативной памяти, то современные Лисп-системы уже могут соперничать по этим параметрам с такими языками, как Си, Паскаль и др. Использование Лисп - машин вообще практически снимает ограничения памяти и быстродействия[17,24].

Лисп - не единственный язык, используемый для задач ИИ. Уже в середине 60-х годов разрабатывались языки, предлагающие другие концептуальные основы. Наиболее важные из них в области обработки символической информации - СНОБОЛ и Рефал.

Это язык обработки строк, в рамках которого впервые появилась и была реализована в достаточно полной мере концепция поиска по образцу. Язык СНОБОЛ был одной из первых практических реализаций развитой продукционной системы. Наиболее известная и интересная версия этого языка Снобол-4 Здесь техника задания образцов и работа с ними существенно опередили потребности практики. По существу, он так и остался «фирменным» языком программирования, хотя концепции СНОБОЛа, безусловно, оказали влияние и на Лисп, и на другие языки программирования задач ИИ.

Язык Рефал - алгоритмический язык рекурсивных функций. Он был создан Турчиным в качестве метаязыка, предназначенного для описания различных, в том числе и алгоритмических, языков и различных видов обработки таких языков. При этом имелось в виду и использование Рефала в качестве метаязыка над самим собой. Для



пользователя это язык обработки символьной информации. Поэтому, помимо описания семантики алгоритмических языков, он нашел и другие применения. Это выполнение громоздких аналитических выкладок в теоретической физике и прикладной математике, интерпретация и компиляция языков программирования, доказательство теорем, моделирование целенаправленного поведения, а в последнее время и задачи ИИ. Общим для всех этих применений являются сложные преобразования над объектами, определенными в некоторых формализованных языках.

В основу языка Рефал положено понятие рекурсивной функции, определенной на множестве произвольных символьных выражений. Базовой структурой данных этого языка являются списки, но не односвязные, как в Лиспе, а двунаправленные. Обработка символов ближе к продукционной парадигме. При этом Активно используется концепция поиска по образцу, характерная для СНОБОЛа.

Программа написанная на Рефале, определяет некоторый набор функций, каждая из которых имеет один аргумент. Вызов функции заключается в функциональные скобки[19,24].

Во многих случаях возникает необходимость из программ, написанных на Рефале, вызывать программы, написанные на других языках. Это просто, так как с точки зрения Рефала первичные функции (Функции, описанные не на Рефале, но которые тем не менее можно вызывать из программ, написанных на этом языке.) - это просто некоторые функции, внешние по отношению к данной

программе, поэтому, вызывая какую-либо функцию, можно даже и не знать, что это - первичная функция.

Семантика Рефал-программы описывается в терминах абстрактной Рефал-машины. Рефал-машина имеет поле памяти и поле зрения. В поле памяти Рефал-машины помещается программа, а в поле зрения - данные, которые будут обрабатываться с ее помощью, т. е. перед началом работы в

поле памяти заносится описание набора функций, а в поле зрения - выражение, подлежащее обработке.

Часто бывает удобно разбить Рефал-программу на части, которые могут обрабатываться компилятором Рефала независимо друг от друга. Наименьшая часть Рефал-программы, которая может быть обработана компилятором независимо от других, называется модулем. Результат компиляции исходного модуля на Рефале представляет собой объектный модуль, который перед исполнением Рефал-программы должен быть объединен с другими модулями, полученными компиляцией с Рефала или других языков это объединение выполняется с помощью редактора связей и загрузчиков. Детали зависят от используемой ОС.

Таким образом, Рефал вобрал в себя лучшие черты наиболее интересных языков обработки символьной информации 60-х годов. В настоящее время язык Рефал используется для автоматизации построения трансляторов, систем аналитических преобразований, а также, подобно Лиспу, в качестве инструментальной среды для реализации языков представления знаний.

#### **7.4. Логические языки программирования.**

В начале 70-х годов появился новый язык составивший конкуренцию Лиспу при реализации систем, ориентированных на знания - Пролог. Этот язык не дает новых сверхмощных средств программирования по сравнению с Лиспом, но поддерживает другую модель организации вычислений. Его привлекательность с практической точки зрения состоит в том, что, подобно тому, как Лисп скрыл от программиста устройство памяти ЭВМ, Пролог позволил ему не заботиться о потоке управления в программе.

Пролог - европейский язык, был разработан в Марсельском университете в 1971 году. Но популярность он стал приобретать только в начале 80-х годов. Это связано с двумя обстоятельствами: во-первых, был

обоснован логический базис этого языка и, во-вторых, в японском проекте вычислительных систем пятого поколения он был выбран в качестве базового для одной из центральных компонент - машины вывода.

Язык Пролог базируется на ограниченном наборе механизмов, включающих в себя сопоставление образцов, древовидное представление структур данных и автоматический возврат. Пролог особенно хорошо приспособлен для решения задач, в которых фигурируют объекты и отношения между ними.

Пролог обладает мощными средствами, позволяющими извлекать информацию из баз данных, причем методы поиска данных, используемые в нем, принципиально отличаются от традиционных. Мощь и гибкость баз данных Пролога, легкость их расширения и модификации делают этот язык очень удобным для коммерческих приложений.

Пролог успешно применяется в таких областях как: реляционные базы данных (язык особенно полезен при создании интерфейсов реляционных баз данных с пользователем); автоматическое решение задач; понимание естественного языка; реализация языков программирования; представление знаний; экспертные системы и др. задачи ИИ.

Теоретической основой Пролога является исчисление предикатов. Прологу присущ ряд свойств, которыми не обладают традиционные языки программирования. К таким свойствам относятся механизм вывода с поиском и возвратом, встроенный механизм сопоставления с образцом. Пролог отличает единообразие программ и данных. Они являются лишь различными точками зрения на объекты Пролога. В языке отсутствуют указатели, операторы присваивания и безусловного перехода. Естественным методом программирования является рекурсия.

Пролог программа состоит из двух частей: базы данных (множество аксиом) и последовательности целевых утверждений, описывающих в совокупности отрицание доказываемой теоремы. Главное

принципиальное отличие интерпретации программы на Прологе от процедуры доказательства теоремы в исчислении предикатов первого порядка состоит в том, что аксиомы в базе данных упорядочены и порядок их следования весьма существенен, так как на этом основан сам алгоритм, реализуемый Пролог-программы. Другое существенное ограничение Пролога в том, что в качестве логических аксиом используются формулы ограниченного класса - так называемые дизъюнкты Хорна. Однако при решении многих практических задач этого достаточно для адекватного представления знаний. Во фразах Хорна после единственного заключения следует ноль и более условий [19,26].

Поиск «полезных» для доказательства формул - комбинаторная задача и при увеличении числа аксиом число шагов вывода катастрофически быстро растет. Поэтому в реальных системах применяют всевозможные стратегии, ограничивающие слепой перебор. В языке Пролог реализована стратегия линейной резолюции, предлагающая использование на каждом шаге в качестве одной из сравниваемых формул отрицание теоремы или ее «потомка», а в качестве другой - одну из аксиом. При этом выбор той или иной аксиомы для сравнения может сразу или через несколько шагов завести в «тупик». Это принуждает вернуться к точке, в которой производился выбор, чтобы испытать новую альтернативу, и т. д. Порядок просмотра альтернативных аксиом не произволен - его задает программист, располагая аксиомы в базе данных в определенном порядке. Кроме того в Прологе предусмотрены достаточно удобные «встроенные» средства для запрещения возврата в ту или иную точку в зависимости от выполнения определенных условий. Таким образом процесс доказательства в Прологе более прост и целенаправлен чем в классическом методе резолюций.

Смысл программы языка Пролог может быть понят либо с позиций декларативного подхода, либо с позиций процедурного подхода. [8]

Декларативный смысл программы определяет, является ли данная цель истинной (достижимой) и, если да, при каких значениях переменных

она достигается. Он подчеркивает статическое существование отношений. Порядок следования подцелей в правиле не влияет на декларативный смысл этого правила. Декларативная модель более близка к семантике логики предикатов, что делает Пролог эффективным языком для представления знаний. Однако в декларативной модели нельзя адекватно представить те фразы, в которых важен порядок следования подцелей. Для пояснения смысла фраз такого рода необходимо воспользоваться процедурной моделью.

### **7.5. Языки интеллектуальных решателей.**

Группа языков, которые можно назвать языками интеллектуальных решателей, в основном ориентирована на такую подобласть ИИ, как решение проблем, для которой характерны, с одной стороны, достаточно простые и хорошо формализуемые модели задач, а с другой - усложненные методы поиска их решения. Поэтому основное внимание в этих языках было уделено введению мощных структур управления, а не способам представления знаний. Это такие языки как Плэнер, Конивер, КюА-4, КюЛисп.

Плэнер. Этот язык дал толчок мощному языкотворчеству в области ИИ. Язык разработан в Массачусетском технологическом институте в 1967-1971гг. Вначале это была надстройка над Лиспом, в таком виде язык реализован на Маклиспе под названием Микро Плэнер. В дальнейшем Плэнер был существенно расширен и превращен в самостоятельный язык. В СССР он реализован под названием Плэнер-БЭСМ и Плэнер-Эльбрус. Этот язык ввел в языки программирования много новых идей: автоматический поиск с возвратами, поиск данных по образцу, вызов процедур по образцу, дедуктивный метод и т. д.

В качестве своего подмножества Плэнер содержит практически весь Лисп (с некоторыми модификациями) и во многом сохраняет его специфические особенности. Структура данных, синтаксис программы

правиль их вычисления в Плэпере аналогичны лисповским. Для обработки данных в Плэпере в основном используются те же средства, что и в Лиспе: рекурсивные и блочные функции[24].

Но между Лиспом и Плэпером существуют и различия. Отметим некоторые из них. В Лиспе при обращении к переменной указывается только её имя, например X, сам же атом как данное указывается как 'X. В Плэпере используется обратная нотация: атомы обозначают самих себя, а при обращении к переменным перед их именем ставится префикс. При этом префикс указывает как должна быть использована переменная. Отличается от лисповского и синтаксис обращения к функциям, которое в Плэпере записывается в виде списка не с круглыми, а с квадратными скобками.

Язык Конивер был разработан в 1972 году, реализован как надстройка над языком Маклисп. Авторы языка Конивер выступили с критикой некоторых идей языка Плэпер. В основном она относилась к автоматическому режиму возвратов, который в общем случае ведёт к неэффективным и неконтролируемым программам, особенно если они составляются неквалифицированными пользователями. Авторы Конивера отказались от автоматического режима возвратов, считая, что встраивать в язык какие-то фиксированные дисциплины управления (кроме простейших-циклов, рекурсий) не следует и что автор программы должен сам организовывать нужные ему дисциплины управления, а для этого язык должен открывать пользователю свою структуру управления и предоставлять средства работы с ней.

### **7.6. Языки представления знаний.**

В рамках каждого базового языка ИИ явным образом выделяются и прямое его использование, и расширение за счёт пакетов функций, и создания «автономного» языка представления знаний

(ЯПЗ) с последующей интерпретацией или компиляцией программ на созданном языке. Но в последнем случае базовый язык, как правило, становится инструментальным средством для реализации ЯПЗ.

Независимо от реализации ЯПЗ должен отвечать следующим требованиям:

- наличие простых и вместе с тем достаточно мощных средств представления сложно структурированных и взаимосвязанных объектов;
- возможность отображения описаний объектов на разные виды памяти ЭВМ;
- наличие гибких средств управления выводом, учитывающих необходимость структурирования правил работы решателя;
- «прозрачность» системных механизмов для программиста, т. е. возможность их до определения и переопределения на уровне входного языка;
- возможность эффективной реализации, как программной, так и аппаратной.

Конечно, перечисленные требования во многом противоречивы. Но лишь тогда, когда в рамках разумного компромисса учтены все эти требования, создаются удачные ЯПЗ.

Среди ЯПЗ первого поколения (70-е годы) лишь некоторые сыграли заметную роль в программной поддержке систем представления знаний (СПЗ). Выделим среди них KRL, FRL, KL-ONE. Характерными чертами этих языков были: двухуровневое представление данных, представление закономерностей предметной области и связей между понятиями в виде присоединенных процедур, семантический подход к сопоставлению образцов и поиску по образцу[19.24].

Один из самых интересных языков этой группы - KRL, в основу которого были положены следующие концепции:

- организация знаний в виде специально выделенных единиц с присоединенными описаниями и процедурами;

- наличие средств представления многоаспектного и/или неполного знания об объектах;
- возможность описания объектов через сопоставление их с другими объектами с учетом уточняющих описаний;
- наличие гибкого набора базовых средств описания стратегий вывода решений и возможность их динамического изменения.

Однако KRL широко не использовался в интеллектуальных системах из-за некоторой его громоздкости и отсутствия собственных средств описания процедур. Как следствие, в KRL активно использовался Лисп. Часто нельзя было понять, имеем мы дело с KRL-программой и присоединенными Лисп-функциями или с Лисп-программой, в которой применяется KRL как способ представления данных<sup>1</sup>

Язык FRL не самостоятельный язык, а хорошо продуманная библиотечная система над Лиспом. В FRL не предлагается принципиально новых по сравнению с KRL концепций представления знаний, но тем не менее он оказался более удобным благодаря тщательному и экономному отбору базовых алгоритмических средств, а также более простому их синтаксическому оформлению. Здесь имеются развитые средства манипулирования иерархическими списками свойств объектов, включая механизмы наследования свойств и набор присоединенных к описаниям процедур.[40]

Для всех ЯПЗ по сравнению с традиционными языками программирования характерна существенно большая «активность» данных, что приводит к стиранию граней между декларативной и процедурной компонентами. Кроме того, реальные объемы обрабатываемых данных требуют при реализации ЯПЗ использования концепции базы данных и методов, развитых при создании СУБД. И, наконец, ЯПЗ тяготеют больше к режиму интерпретации, чем к

---

<sup>1</sup> А.Н. Мелихов, Л.С. Бернштейн, С.Я. Коровин. Ситуационные советующие системы с нечеткой логикой. - М.: Наука, 2004г.



компиляции, характерной для реализации обычных языков программирования. В области разработки и реализации ЯПЗ можно выделить три круга проблем: определение входных языков СПЗ; выбор выходного языка соответствующего транслятора и собственно проблемы этапа трансляции.

Входной язык СПЗ должен быть близок к языку предметной области и по лексике, и по синтаксису, и по семантике.

От выбора выходного языка зависит не только эффективность, но и сама возможность реализации ЯПЗ. Выходной язык должен отвечать по крайней мере следующим требованиям: иметь достаточно большой набор примитивов работы с образцами; обладать встроенными средствами эффективной поддержки рекурсии; иметь гибкие средства описания потоков управления. Кроме того, в рамках выходного языка необходимы средства отображения данных на основную и внешнюю память и удобные средства работы с этими данными. И, наконец, желательно, чтобы в нем имелись достаточно развитые средства определения новых типов данных.

В настоящее время языков программирования, где имела бы место эффективная реализация всех указанных требований, пока нет. Поэтому выбор целевого языка ЯПЗ-транслятора всегда компромисс.

На данном этапе существует сотни языков и систем представления знаний. Поэтому рассмотрим лишь некоторые особенности нескольких ЯПЗ.

RLL. Это фреймовый язык представления знаний (представитель популярного в 70-х годах подхода «фреймы до конца»), является инструментальной средой для создания специализированных ЯПЗ.

Подобно другим инструментальным средствам, RLL содержит два слоя: базисные примитивы и средства их комбинирования на более высоком уровне, чем Лисп. При этом технология конструирования специальных ЯПЗ в рамках RLL-среды сводится, как правило, к редактированию уже существующих заготовок и последующему

конвертированию их в Лисп.

Учитывая последовательную ориентацию RLL на концепцию фреймов, все структуры (декларативные и процедурные), более сложные чем список значений, описываются здесь в виде фрейм-подобных RLL-элементов.

С помощью RLL-элементов описываются понятия не только предметной области, но и самой RLL-среды (например, слот, механизм наследования, структура управления и т. д.). Заранее на уровне RLL-интерпретатора или конвертора фиксируется семантика ограниченного числа системных понятий -это множества, списки, слоты и др. RLL-элементы имеют явно специфицированные родовидовые отношения, которые также являются системными понятиями, и встроенный механизм описания отношений с помощью многосвязных списков.

В RLL имеется и библиотека удачных управляющих структур, и определенные средства конструирования из них решателей, необходимых для конкретной эксперной системы (ЭС).

ART. Этот язык демонстрирует другую парадигму «фреймы плюс продукции», характерную для начала 80-х годов. Это не только язык представления знаний, но и определенное программное окружение, включающее редакторы, отладчики, трансляторы и модули управления.

Входной язык системы ART весьма гибкий и обеспечивает использование фактов, схем, комбинаций этих понятий и правил. Декларативную компоненту этого ЯПЗ составляют факты и схемы. По определению, факт включает три основных компонента: утверждение, значение истинности и точку зрения. С каждым утверждением может быть связано одно из трех значений истинности *Iше*, *iaize* или *шкпо* □□п, а также определенные сферы его справедливости, которое и называется точкой зрения. Факты описываются экземплярами фреймов. Фреймы-прототипы в ART представляются схемами, каждая из которых описывает объекты и/или классы объектов с фиксированными

свойствами. Механизмы наследования свойств при этом поддерживаются самой системой.

В целом язык ART погружен в Лисп-среду, так что синтаксически и фреймовые и продукционные структуры выражаются здесь как атомы, функции и списки языка Лисп. Такой подход в ART естественен, так как первоначально был реализован на Лисп-машинах. Средства описания фактов в языке ART почти полностью «отданы на откуп» Лиспу, что снижает концептуальную целостность языка, так как средства описания схем и правил здесь хотя и похожи на лисповские, но свои. В ART пользователю дается небольшой набор встроенных стратегий вывода решений и весьма ограниченный выбор из ART-действий, взаимодействующих с модулем вывода. Но в системе имеется возможность выхода в базовый язык Лисп, где программируются любые управляющие стратегии.

В полном объеме ART представляет разработчику ЭС достаточно мощные средства представления знаний, но эффективно в системе ART могут работать только квалифицированные Лисп-программисты, готовые реализовать в этом языке все процедуры поддержки ЯПЗ[19,24].

Дальнейшее развитие ЯПЗ смещается в сторону продукций. Вместе с тем в настоящее время уже редко удается классифицировать языки и системы представления знаний на шкале «фреймы - продукции - семантические сети -..» однозначно. И хотя тот или иной формализм представления знаний накладывает в большей или меньшей степени свой отпечаток на соответствующий ЯПЗ, современные языки и системы, как правило, поддерживают несколько формализмов одновременно.

### **Краткие выводы.**

В данной теме дана общая характеристика языков программирования используемых для построения экспертных систем, а также представлена их классификация. Дана сравнительная характеристика функциональных языков программирования: Лисп, Снобол, Рефал и дана подробная характеристика логического языка программирования Пролог. Кроме того, рассмотрены языки интеллектуальных решателей: Плэнер, Конивер и языки представления знаний: KRL, FRL, RLL и ART.

### **Ключевые слова:**

Языки программирования, функциональные языки программирования, Лисп, Снобол, Рефал, логические языки программирования, Пролог, Плэнер, Конивер, языки представления знаний, KRL, FRL, RLL, ART, языки интеллектуальных решателей, языки низкого уровня, поиск по образцам, рекурсия, исчисление предикатов, база данных, система представления знаний, входные языки, выходные языки, транслятор, фреймы.

### **Вопросы для обсуждения и контроля.**

- 1) Какие языки программирования, используемые для построения экспертных систем Вы знаете?
- 2) Какие языки программирования относятся к функциональным языкам?
- 3) Какие языки программирования относятся к логическим языкам?
- 4) В чем заключается сущность логического программирования?
- 5) Как можно представить структуру функциональной программы?

- 6) Дайте сравнительную характеристику функциональным языкам программирования?
- 7) Какими характерными особенностями обладают языки интеллектуальных решателей?
- 8) Каким требованиям должен отвечать?
- 9) Назовите наиболее распространенные языки представления знаний?

### **Литература.**

1. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления. Учебник. 2-е изд. (+CD).- СПб.: Питер, 2009.-120 с.
2. Информационные технологии управления. Учебное пособие для вузов /Под ред. проф. ГА Титоренко. — 2-е изд., доп. — М: ЮНИТИ-ДАНА, 2007. - 439 с.
3. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Учебник, М.: Финансы и статистика, 2004 г. - 424с.
4. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем.- СПб.: Питер,2001. -384 с.
5. А.Н. Мелихов, Л.С. Бернштейн, С.Я. Коровин. Ситуационные советующие системы с нечеткой логикой. - М.: Наука, 2004г.
6. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н.Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С.- Ташкент, 2003. – 135 с.

## ГЛАВА 8. ТЕХНОЛОГИЯ РАЗРАБОТКИ ЭКСПЕРТНЫХ СИСТЕМ

- 8.1. Этапы разработки экспертных систем.
- 8.2. Этап 1: выбор подходящей проблемы.
- 8.3. Этап 2: разработка прототипной системы.
- 8.4. Этап 3: развитие прототипа до промышленной ЭС.
- 8.5. Этап 4: оценка системы.
- 8.6. Этап 5: стыковка системы и этап 6: поддержка системы.

### 8.1. Этапы разработки экспертных систем.

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

Процесс разработки промышленной экспертной системы, опираясь на традиционные технологии, можно разделить на шесть более или менее независимых этапов (рис. 8.1), практически не зависящих от предметной области.

Последовательность этапов дана для общего представления о создании идеального проекта. Конечно, последовательность эта не вполне фиксированная. В действительности каждый последующий этап разработки ЭС приносит новые идеи, которые могут повлиять

на предыдущие решения и даже привести к их переработке. Именно поэтому многие специалисты по информатике весьма критично относятся к методологии экспертных систем. Они считают, что расходы на разработку таких систем очень большие, время разработки слишком длительное, а полученные в результате программы ложатся тяжелым бременем на вычислительные ресурсы.

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

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

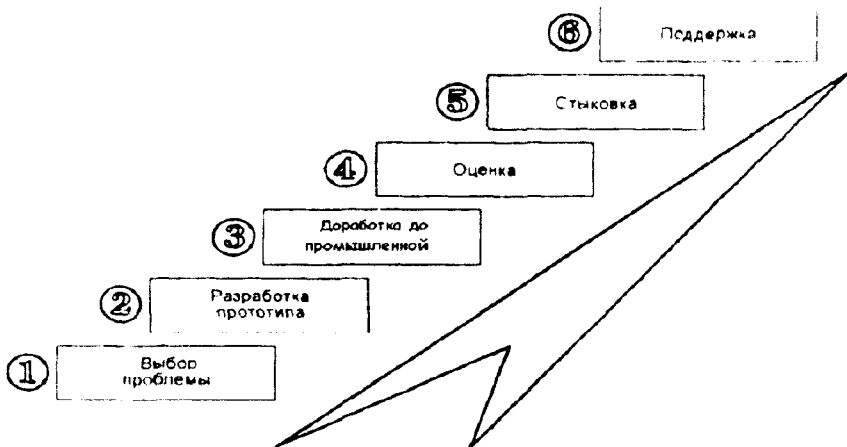


Рис 8.1. Этапы разработки экспертных систем

Решение таких задач, во-первых, подготавливает высококвал-

лифицированных специалистов по информатике, необходимых для создания интеллектуальных систем, во-вторых, позволяет отделить от экспертных систем неэкспертные задачи.

## **8.2. Этап 1: выбор подходящей проблемы**

Этот этап включает деятельность, предшествующую решению начать разрабатывать конкретную ЭС. Он включает(рис 8.2.):

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

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





Рис 8.2. Состав работ на этапе «выбор подходящей проблемы».

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

Экспертная система ни в коем случае не устранил потребность в реляционных базах данных, статистическом программном обеспечении, электронных таблицах и системах текстовой обработки. Но

если результативность задачи зависит от знания, которое является субъективным, изменяющимся, символьным или вытекающим частично из соображений здравого смысла, тогда область может обоснованно выступить претендентом на экспертную систему.

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

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

Подходящие задачи имеют следующие характеристики:

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

Обычно экспертные системы разрабатываются путем получения специфических знаний от эксперта и ввода их в систему. Некоторые системы могут содержать стратегии одного индивида. Следо-

вательно, найти подходящего эксперта — это ключевой шаг в создании экспертных систем.

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

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

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

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

Прибыль возможна за счет снижения цены продукции, повышения производительности труда, расширения номенклатуры продукции или услуг или даже разработки новых видов продукции или услуг в этой области. Соответствующие расходы и прибыли от системы определяются относительно времени, в течение которого возвращаются средства, вложенные в разработку. На современном эта-

не большая часть фирм, развивающих крупные экспертные системы, предпочли разрабатывать дорогостоящие проекты, приносящие значительные прибыли.

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

После того как инженер по знаниям убедился, что:

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

### **8.3. Этап 2: разработка прототипной системы.**

#### **Понятие прототипной системы**

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

Объем прототипа — несколько десятков правил, фреймов или

примеров. На рис. 8.3. изображены шесть стадий разработки прототипа и минимальный коллектив разработчиков, занятых на каждой из стадий. Приведем краткую характеристику каждой из стадий, хотя эта схема представляет грубое приближение к сложному итеративному процессу.

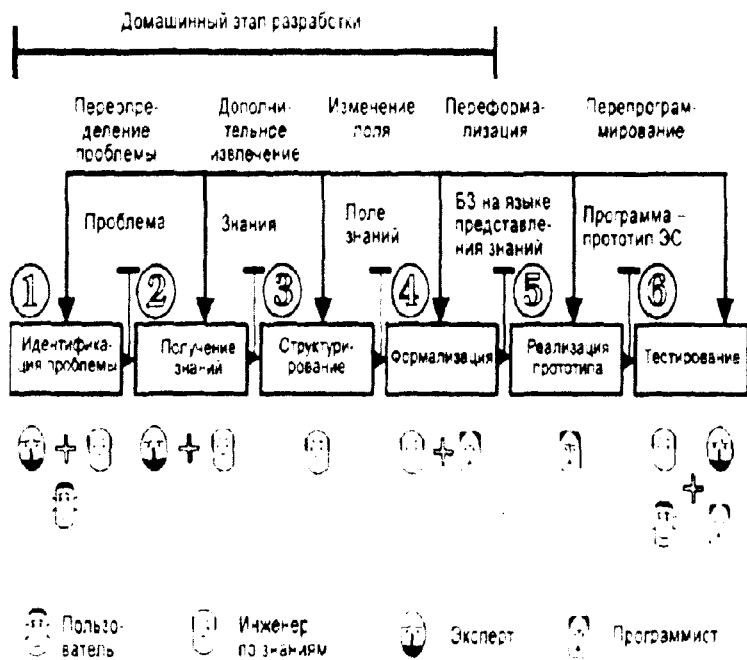


Рис. 8.3. Стадии и участники разработки прототипа экспертной системы

Хотя любое теоретическое разделение бывает часто условным, осознание коллективом разработчиков четких задач каждой стадии представляется целесообразным. Роли разработчиков (эксперт, про-

граммист, пользователь и аналитик) являются постоянными на протяжении всей разработки. Совмещение ролей нежелательно.

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

### **Идентификация проблемы**

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

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

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

Средняя продолжительность 1 - 2 недели.

### **Извлечение знаний**

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

- анализ текстов;
- диалоги;
- экспертные игры;
- лекции;
- дискуссии;
- интервью;
- наблюдение и другие.

**Извлечение знаний** — получение инженером по знаниям наиболее полного представления о предметной области и способах

принятия решения в ней.

Средняя продолжительность 1—3 месяца.

### **Структурирование или концептуализация знаний**

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

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

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

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

### **Формализация**

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

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

классов, объектов и др.

**Формализация знаний** — разработка базы знаний на языке, который, с одной стороны, соответствует структуре поля знаний, а с другой — позволяет реализовать прототип системы на следующей

стадии программной реализации.

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

### **Реализация**

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

- программирование на традиционных языках типа Паскаль, Си и др.;

- программирование на специализированных языках, применяемых в задачах искусственного интеллекта: LISP, FRL, SmallTalk и др.;

- использование инструментальных средств разработки ЭС типа СПЭИС, ПИЭС;

- использование "пустых" ЭС или "оболочек" типа ЭКСПЕРТ, ФИАКР и др.

**Реализация** — разработка программного комплекса, демонстрирующего жизнеспособность подхода в целом. Чаще всего первый прототип отбрасывается на этапе реализации действующей ЭС.

Средняя продолжительность 1 - 2 месяца.

### **Тестирование**

Оценивается и проверяется работа программ прототипа с целью приведения в соответствие с реальными запросами пользователей. Прототип проверяется на:

- удобство и адекватность интерфейсов ввода-вывода (характер вопросов в диалоге, связность выводимого текста результата и др.);

- эффективность стратегии управления (порядок перебора, ис-



пользование нечеткого вывода и др.);

- качество проверочных примеров;
- корректность базы знаний (полнота и непротиворечивость правил).

**Тестирование** — выявление ошибок в подходе и реализации прототипа и выработка рекомендаций по доводке системы до промышленного варианта.

Средняя продолжительность 1 -2 недели.

#### **8.4. Этап 3: развитие прототипа до промышленной экспертной системы**

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

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

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

Понятие же коммерческой системы в нашей стране входит в понятие промышленный программный продукт, или промышленной ЭС в этой работе (табл. 8.1).

Основное на третьем этапе заключается в добавлении большо-

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

**Таблица 8.1. Переход от прототипа к промышленной экспертной системе**

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

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

стности, могут оказаться полезными графические представления.

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

### **8.5. Этап 4: оценка системы**

После завершения этапа разработки промышленной экспертной системы необходимо провести ее тестирование в отношении критериев эффективности. К тестированию широко привлекаются другие эксперты с целью апробирования работоспособности системы на различных примерах. Экспертные системы оцениваются главным образом для того, чтобы проверить точность работы программы и ее полезность. Оценку можно проводить, исходя из различных критериев, которые сгруппируем следующим образом:

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

### **8.6. Этап 5: стыковка системы и этап 6: поддержка системы**

#### **Этап 5: стыковка системы**

На этом этапе осуществляется стыковка экспертной системы с другими

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

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

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

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

**Пример 1.** Успешно состыкована со своим окружением система PUFF — экспертная система для диагностики заболеваний легких.[15] После того, как PUFF была закончена и все были удовлетворены ее работой, систему перекодировали с LISPa на Бейсик. Затем систему перенесли на ПК, которая уже работала в больнице. В свою очередь, эта ПК была связана с измерительными приборами. Данные с измерительных приборов сразу поступают в ПК. PUFF обрабатывает эти данные и печатает рекомендации для врача. Врач в принципе не взаимодействует с PUFF. Система полностью

интегрирована со своим окружением — она представляет собой интеллектуальное расширение аппарата исследования легких, который врачи давно используют.

**Пример 2.** Другая система, которая хорошо функционирует в своем окружении, — CAT-1 [15] — экспертная система для диагностики неисправностей дизелей локомотивов.

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

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

### **Этап 6: поддержка системы**

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

**Пример 3.** Удачным примером ЭС, внедренной таким образом, является XCON (R1) — ЭС, которую фирма DEC использует для комплектации ЭВМ

семейства VAX. Одна из ключевых проблем, с которой столкнулась фирма DEC, — необходимость постоянного внесения изменений для новых версий оборудования, новых спецификаций и т.д. Для этой цели XCON поддерживается в программной среде OPS5. [15].

### **Краткие выводы.**

В данной теме представлены этапы разработки экспертных систем, рассмотрен перечень работ, выполняемых на каждом этапе. Кроме того, дана характеристика понятию прототипной системы. Приведены стадии разработки прототипа экспертной системы.

### **Ключевые слова:**

Интеллектуальный редактор БЗ, интерфейс пользователя, база знаний, данные, подсистема объяснений, поле знаний, представление знаний, инженер по знаниям, Пролог, проблема, прототипная система, оценка, стыковка, поддержка, идентификация проблемы, извлечение знаний, формализация, реализация, тестирование.

### **Вопросы для обсуждения и контроля**

1. Какие работы выполняются на этапе «выбор подходящей проблемы»?
2. Какая из выполняемых на первом этапе работ, на Ваш взгляд является наиболее важной?
3. Дайте характеристику следующим стадиям разработки прототипа ЭС: структурирование знаний и формализация знаний.
4. Какие способы используются на стадии реализации прототипа ЭС?
5. Какие дополнительные этапы для перехода от прототипа к промышленной ЭС Вы знаете?

6. Раскройте сущность каждого прототипа ЭС.
7. На каком этапе разработки ЭС происходит добавление эвристик?
8. Назовите критерии для оценки ЭС.
9. Какие работы выполняются на этапе «стыковка системы»?
10. Обоснуйте важность этапа «поддержка системы».

### **Литература.**

1. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления. Учебник. 2-е изд. (+CD). - СПб.: Питер, 2009. -120 с.
2. Информационные технологии управления. Учебное пособие для вузов /Под ред. проф. ГА Титоренко. — 2-е изд., доп. — М: ЮНИТИ-ДАНА, 2007. - 439 с.
3. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Учебник, М.: Финансы и статистика, 2004 г. - 424с.
4. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. - СПб.: Питер, 2001. -384 с.
5. А.Н. Мелихов, Л.С. Бернштейн, С.Я. Коровин. Ситуационные советующие системы с нечеткой логикой. - М.: Наука, 2004г.
6. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н. Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С.- Ташкент, 2003. – 135 с.
7. Информатика: Учебник/ Под ред. Проф. Н.В. Макаровой. 3-е изд. -М.: Финансы и статистика, 2006. –768с.

## ГЛАВА 9. КЛАССИФИКАЦИЯ И ИСПОЛЬЗОВАНИЕ ЭКСПЕРТНЫХ СИСТЕМ.

- 9.1. Классификация экспертных систем.
- 9.2. Схема взаимодействия пользователей с экспертной системой.
- 9.3. Технология использования экспертных систем.
- 9.4. Области применения экспертных систем.

### 9.1. Классификация экспертных систем.

Экспертные системы сегодня объединяет несколько тысяч различных программных комплексов, которые можно классифицировать по различным критериям (рис. 9.1.).

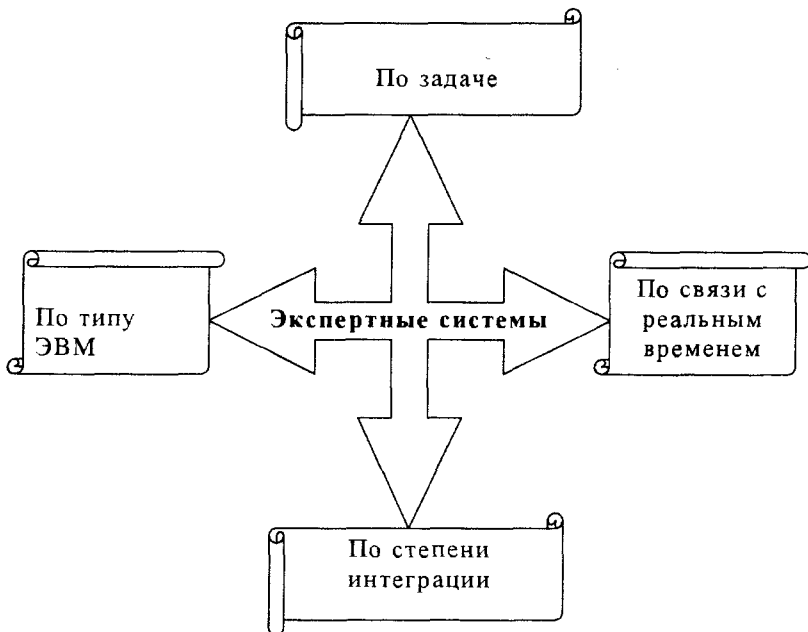


Рис. 9.1. Критерии классификации экспертных систем



Рассмотрим классификацию по каждому критерию.  
(рис. 9.2.)[14].



Рис. 9.2. Классификация экспертных систем.

Классификация по решаемой задаче.

**Интерпретация данных.** Это одна из традиционных задач для экспертных систем. Под интерпретацией понимается определение смысла данных, результаты которого должны быть согласованными и корректными. Обычно предусматривается многовариантный анализ данных.

**Диагностика.** Под диагностикой понимается обнаружение неисправности в некоторой системе. Неисправность - это отклонение от нормы. Такая трактовка позволяет с единых теоретических позиций рассматривать и неисправность оборудования в технических системах, и заболевания живых организмов, и всевозможные природные аномалии. Важной спецификой является необходимость понимания функциональной структуры («анатомии») диагностирующей системы.

**Мониторинг.** Основная задача мониторинга - непрерывная интерпретация данных в реальном масштабе времени и сигнализация о выходе тех или иных параметров за допустимые пределы. Главные проблемы - «пропуск» тревожной ситуации и инверсная задача «ложного» срабатывания. Сложность этих проблем в размытости симптомов тревожных ситуаций и необходимость учета временного контекста.

**Проектирование.** Проектирование состоит в подготовке спецификаций на создание «объектов» с заранее определенными свойствами. Под спецификацией понимается весь набор необходимых документов - чертеж, пояснительная записка и т.д. Основные проблемы здесь - получение четкого структурного описания знаний об объекте и проблема «следа». Для организации эффективного проектирования и, в еще большей степени, перепроектирования необходимо формировать не только сами проектные решения, но и мотивы их принятия. Таким образом, в задачах проектирования тесно свя-

зываются два основных процесса, выполняемых в рамках соответствующей ЭС: процесс вывода решения и процесс объяснения.

**Прогнозирование.** Прогнозирующие системы логически выводят вероятные следствия из заданных ситуаций. В прогнозирующей системе обычно используются параметрическая динамическая модель, в которой значения параметров «подгоняются» под заданную ситуацию. Выводимые из этой модели следствия составляют основу для прогнозов с вероятностными оценками.

**Планирование.** Под планированием понимается нахождение планов действий, относящихся к объектам, способным выполнять некоторые функции. В таких ЭС используются модели поведения реальных объектов с тем, чтобы логически вывести последствия планируемой деятельности.

**Обучение.** Система обучения диагностирует ошибки при изучении какой-либо дисциплины с помощью ЭВМ и подсказывают правильное решение. Они аккумулируют знания о гипотетическом «ученике» и его характерных ошибках, затем в работе способны диагностировать слабости в знаниях обучаемых и находить соответствующие средства для их ликвидации. Кроме того, они планируют акт общения с учеником в зависимости от успехов ученика с целью передачи знаний.

В общем случае все системы, основанные на знаниях, можно подразделить на системы, решающие задачи анализа, и на системы, решающие задачи синтеза. Основные отличия задач анализа от задач синтеза заключается в следующем: если в задачах анализа множество решений может быть перечислено и включено в систему, то в задачах синтеза множество решений потенциально строится из решений компонентов или под проблем. Задача анализа - это интерпретация данных, диагностика; к задачам синтеза относится проектирование, планирование. Комбинированные задачи: обучение, мо-

ниторинг, прогнозирование.

**Классификация по связи с реальным временем.**

**Статистические ЭС** разрабатываются в предметных областях, в которых база знаний и интерпретируемые данные не меняются во времени. Они стабильны.

**Квазидинамические ЭС** интерпретируют ситуацию, которая меняется с некоторым фиксированным интервалом времени.

**Динамические ЭС** работают в сопряжении с датчиками объектов в режиме реального времени с непрерывной интерпретацией поступаемых данных.

**Классификация по типу ЭВМ.**

На сегодняшний день существуют:

- \* ЭС для уникальных стратегических важных задач на супер-ЭВМ (Эльбрус, CRAY, CONVEX и др.);
- \* ЭС на ЭВМ средней производительности (типа ЕС ЭВМ, main-frame);
- \* ЭС на символьных процессорах и рабочих станциях (SUN, APOLLO);
- \* ЭС на мини- и супермини- ЭВМ (VAX, micro-VAX и др.);
- \* ЭС на персональных компьютерах (IBM PC, MACII и подобные).

**Классификация по степени интеграции с другими программами.**

**Автономные ЭС** работают непосредственно в режиме консультаций с пользователем для специфически «экспертных» задач, для решения некоторых не требуется привлекать традиционные методы обработки данных (расчеты, моделирование и т.д.).

**Гибридные ЭС** представляют программный комплекс, агрегирующий стандартные пакеты прикладных программ (например, математическую статистику, линейное программирование или систе-

мы управления базами данных) и средства манипулирования знаниями. Это может быть интеллектуальная надстройка над ППП или интегрированная среда для решения сложной задачи с элементами экспертных знаний.

Несмотря на внешнюю привлекательность гибридного подхода, следует отметить, что разработка таких систем является задачей, на порядок более сложную, чем разработка автономной ЭС. Стыковка не просто разных пакетов, а разных методологий (что происходит в гибридных системах) порождает целый комплекс теоретических и практических трудностей.

## **9.2. Схема взаимодействия пользователей с экспертной системой**

Выделим следующие группы пользователей экспертных систем:

- 1) пользователи-специалисты;
- 2) пользователи-неспециалисты;
- 3) эксперты;
- 4) пользователи - обучающиеся.

Всех их объединяет следующее:

- язык общения с экспертной системой- это ограниченный естественный язык, а не язык программирования;
- процесс взаимодействия пользователя с экспертной системой не сводится к обмену предложениями типа вопрос-ответ, а представляет собой разветвлённый диалог, в котором инициатива переходит от одного участника к другому.

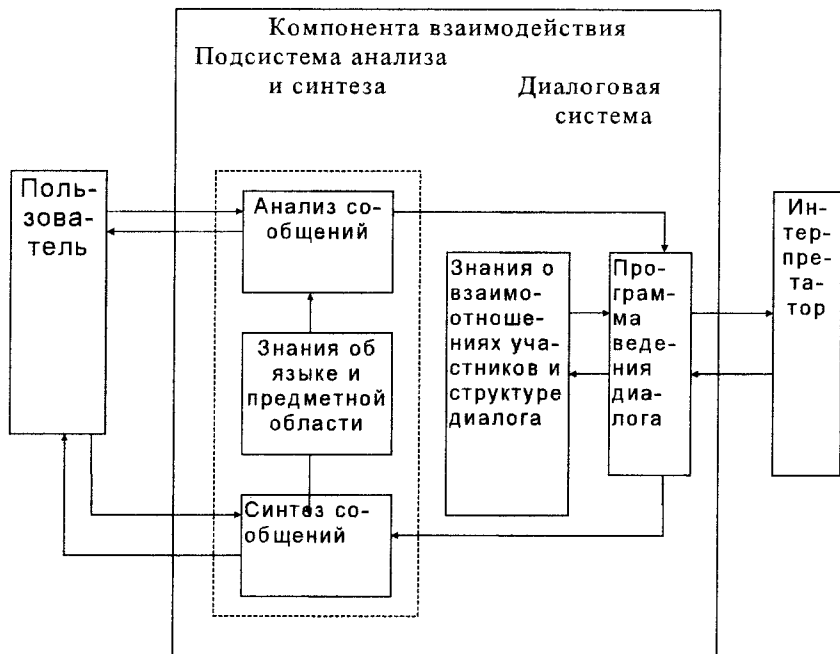


Рис.9.3. Общая схема компонента взаимодействия.

На рисунке 9.3. представлена общая схема компонента взаимодействия пользователей с экспертной системой.

В составе экспертных систем выделяется **компонента взаимодействия**. Её назначение состоит в следующем:

- 1) организовать диалог пользователь- экспертная система,
- 2) осуществить обработку отдельных сообщений с учётом текущего состояния диалога, т.е. с естественного языка во внутреннюю форму экспертной системы[26].

В настоящее время не существует общепринятой модели диалога с экспертной системой. В общем случае диалог может быть охарактеризован следующими компонентами:

1. множеством параметров, которые определяют участников

диалога и тему;

2. описанием параметров;
3. макроструктурой диалога.

### **9.3.Технология использования экспертных систем.**

Создание и использование экспертных систем является одним из концептуальных этапов развития информационных технологий. В основе интеллектуального решения проблем некоторой предметной области лежит принцип воспроизведения знаний опытных специалистов- экспертов.

Исходя из собственного опыта эксперт анализирует ситуацию и распознает наиболее полезную информацию, оптимизирует принятие решений, отсекает тупиковые пути.

Экспертная система - это совокупность методов и средств организации, накопления и применения знаний для решения сложных задач в некоторой предметной области. Экспертная система достигает более высокой эффективности за счет перебора большего числа альтернатив при выборе решения, опираясь на высококачественный опыт группы специалистов, анализирует влияние большего объема новых факторов, оценивая их при построении стратегии, добавляя возможности прогноза[14].

Основой экспертной системы является совокупность знаний (базы знаний), структурированных в целях формализации процесса принятия решений.

Экспертные системы разрабатываются с расчетом на обучение и способны обосновать логику выбора решения, то есть обладают свойствами адаптивности и ее аргументирования. У большинства экспертных систем имеется механизм объяснения. Этот механизм использует знания, необходимые для объяснения того, каким

образом системе пришла к данному решению. Очень важным является определение области применения экспертной системы, границ ее использования и действия.

Преимущества экспертных систем по сравнению с использованием опытных специалистов состоят в следующем:

- Достигнутая компетентность не утрачивается, может документироваться, передаваться, воспроизводиться и наращиваться.
- Имеют место более устойчивые результаты, отсутствуют эмоциональные и другие факторы человеческой ненадежности.
- Высокая стоимость разработки уравнивается низкой стоимостью эксплуатации, возможности копирования, а совокупности они дешевле высококвалифицированных специалистов.

Недостатком экспертных систем, характерным для их современного состояния является меньшая приспособляемость к обучению новым правилам и концепциям, к творчеству и изобретательству. Использование экспертных систем позволяет во многих случаях отказаться от высококвалифицированных специалистов, но предполагает оставить в системе место эксперту с более низкой квалификацией. Экспертные системы служат средством для расширения и усиления профессиональных возможностей конечного пользователя.

Экспертная система должна демонстрировать компетентность, то есть достигать в конкретной предметной области того же уровня, что и специалисты-эксперты. Недостаточно находить хорошие решения, это надо делать быстро. Системы должны иметь не только глубокое, но и достаточно широкое понимание предмета. Методы нахождения решений проблем достигаются на основе рассуждений, исходящих из фундаментальных принципов в случае некорректных данных или неполных наборов правил. Такие свойства наименее разработаны в компьютерных экспертных системах, но именно они



присущи специалистам высокого уровня.

Отличиями экспертных систем от обычных компьютерных является:

- Экспертные системы манипулируют знаниями, тогда как любые другие системы - данными.

- Экспертные системы, как правило, дают эффективные оптимальные решения и способны иногда ошибаться, но в отличие от традиционных компьютерных систем они имеют потенциальную способность учиться на своих ошибках.

Экспертные системы как инструмент в работе пользователей совершенствуют свои возможности решать трудные, неординарные задачи в ходе практической работы.

#### **9.4. Области применения экспертных систем.**

Экспертные системы создаются для решения разного рода проблем, типы которых можно сгруппировать в категории (таблица 9.1.) [14].

На рисунке 9.4. перечислены некоторые из предметных областей, в которых применяются экспертные системы. Их них особенно популярна медицина.

Наиболее уязвимы экспертные системы в распознавании границ своих возможностей и демонстрируют ненадежное функционирование вблизи границ их применимости. Дальнейший прогресс в области искусственного интеллекта со временем предложит способы выявления границ своих возможностей. Другим недостатком экспертных систем являются значительные трудозатраты, необходимые для пополнения базы знаний. Получение знаний от экспертов и внесение их в базу знаний представляет собой сложный процесс, сопряженный со значительными затратами времени и средств. Проектирование экспертных систем также имеет определенные трудно-

сти и ограничения, которые влияют на их разработку.

Таблица 9.1.

Типичные категории применения экспертных систем.

<i>Категория</i>	<i>Решаемая проблема</i>
Интерпретация	Описание ситуации по информации, поступающей от датчиков
Прогноз	Определение вероятных последствий заданных ситуаций
Диагностика	Выявление причин неправильного функционирования системы по результатам наблюдений
Проектирование	Построение конфигурации объектов при заданных ограничениях
Планирование	Определение последовательности действий
Наблюдение	Сравнение результатов наблюдений с ожидаемыми результатами
Отладка	Составление рецептов исправления неправильного функционирования системы
Ремонт	Выполнение последовательности предписанных исправлений
Обучение	Диагностика, отладка и исправление поведения обучаемого
Управление	Управление поведением системы как целого

Зарубежный опыт показывает, что экспертные системы разрабатываются в основном в университетах, научно-исследовательских центрах и коммерческих организациях, в том числе и для финансо-

вой индустрии. В сфере финансового обслуживания эти системы помогают страховым компаниям анализировать и оценивать коммерческий риск, устанавливать размеры ссуд при кредитовании организации, составлять сметы проектов и т.д.

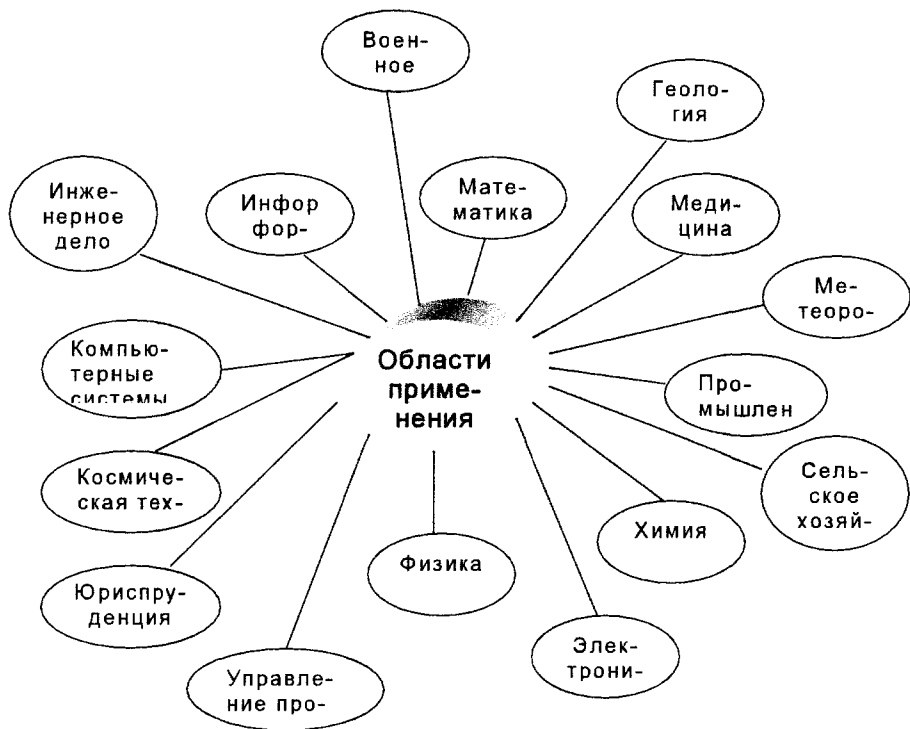


Рис 9.4. Области применения экспертных систем

Область применения экспертных систем расширяется. Кроме охвата различных областей деятельности, одним из наиболее важных последствий разработки экспертных систем является модификация знаний. По мере того как разработчики будут строить боль-

шие, сложные базы знаний, появляется рынок знаний, не зависящих от компьютерных систем. Появятся средства обучения для изучающих определенную прикладную область. Коммерческим продуктом станут метазнания, то есть знания об оптимальных стратегиях и процедурах использования предметных знаний. Развитие экспертных систем в интеллектуальные состоит в слиянии концепций обобщения, средств их создания (языков) и самих экспертных систем. Объединение интеллектуальных систем особенно эффективно в сложных инфраструктурах. Интеллектуальные системы уже разрабатываются и внедряются за рубежом только для коммерческого использования.

Экспертная система FOLIO (Стенфордский университет, США) помогает консультантам по инвестициям определять цели клиентов и подбирать портфели ценных бумаг, наиболее соответствующие этим целям. Система определяет нужды клиента в ходе интервью и затем рекомендует в каких пропорциях надо распределить капиталовложения между разными фондовыми инструментами, чтобы наилучшим образом удовлетворить запросы клиента. Система различает небольшое число классов ценных бумаг (например, ориентированные на дивиденды акций с невысоким уровнем риска или ориентированные на акции с высоким уровнем риска) и содержат знания о свойствах (например, годовых процентах на капитал) ценных бумаг каждого класса. В системе применена основанная на правилах схема представления знаний с прямой цепочкой рассуждений для вывода целей и схема линейного программирования для максимизации соответствия между целями и предлагаемым портфелем.

Искусственная компетентность экспертных систем не заменяет полностью человека. Эксперт – человек способен реорганизовать информацию и знания и использовать их для синтеза новых знаний.

В области творческой деятельности люди обладают большими способностями и возможностями по сравнению с самыми умными системами. Эксперты справляются с неожиданными поворотами событий и, используя новые подходы, способны проводить аналогии из других предметных областей. Эксперты адаптируют к изменяющимся условиям и приспособливают свои стратегии к новым обстоятельствам в более широком диапазоне проблем и задач. Экспертные системы менее приспособлены к обучению на уровне новых концепций и новых правил. Они оказываются не столь эффективны и мало пригодны в тех случаях, когда надо учитывать всю сложность реальных задач.

Эксперты могут непосредственно воспринимать весь комплекс входной информации: символьной, визуальной, графической, текстовой, звуковой, осязательной и обонятельной. У экспертной системы есть только символы, с помощью которых представлены базы знаний, воплощающие те или иные концепции. Преобразование сенсорной информации в символьную сопровождается потерей части информации.

Но главное, что огромный объем знаний, которым обладают эксперты-специалисты (профессиональные знания и знания о мире и действующих в нем законов), не удается пока строить в интеллектуальную систему, тем более столь специализированную, какой является любая экспертная система.

### **Краткие выводы.**

В данной теме представлена классификация экспертных систем, описаны инструментальные средства построения экспертных систем, определены группы пользователей экспертных систем, представлена схема взаимодействия пользователей с экспертной системой, описана технология использования и области примене-

ния экспертных систем, и дана основная терминология экспертных систем.

### **Ключевые слова:**

Классификация, задача, диагностика, мониторинг, проектирование, прогнозирование, планирование, обучение, время, статистические, квазидинамические, динамические, супер-ЭВМ, символьный процессор, мини-ЭВМ, ПЭВМ, автономные, гибридные, пользователь, компонента взаимодействия, диалог, синтез, анализ, преимущества, компетентность, устойчивость, стоимость, недостатки, категории, области применения.

### **Вопросы для обсуждения и контроля.**

- 1) По каким основным критериям можно классифицировать экспертные системы?
- 2) Определите классификацию экспертных систем по решаемой задаче?
- 3) Определите классификацию экспертных систем по связи с реальным временем?
- 4) Определите классификацию экспертных систем по типу ЭВМ?
- 5) Определите классификацию экспертных систем по степени интеграции с другими программами?
- 6) Каким образом пользователи могут взаимодействовать с экспертной системой?
- 7) В чем заключается технология использования экспертных систем?
- 8) Чем отличаются экспертные системы от обычных компьютерных систем?
- 9) В каких предметных областях применяются экспертные

системы?

### **Литература.**

1. Автоматизированные информационные технологии в экономике: Учебник/ Под ред. проф. Г. А. Титоренко. - М.: Юнити, 2006г. - 399 с.
2. Информатика: Учебник/ Под ред. Проф. Н.В. Макаровой. 3-е изд. -М.: Финансы и статистика, 2006. -768с.
3. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Учебник, М.: Финансы и статистика, 2004 г. - 424с.
4. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н. Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С. - Ташкент, 2003. – 135 с.
5. Джексон Питер. Введение в экспертные системы. Пер. с англ. Уч. пос. - М.: Издательский дом «Вильямс» 2001. – 624 с.

## **ТЕМА 10. ПЕРСПЕКТИВЫ РАЗВИТИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ.**

10.1. Направления развития искусственного интеллекта.

10.2. Нейросетевые технологии.

### **10.1. Направления развития искусственного интеллекта.**

Программные средства, базирующиеся на технологии и методах искусственного интеллекта, получили значительное распространение в мире. Их важность, и, в первую очередь, экспертных систем и нейронных сетей, состоит в том, что данные технологии существенно расширяют круг практически значимых задач, которые можно решать на компьютерах, и их решение приносит значительный экономический эффект. В то же время, технология экспертных систем является важнейшим средством в решении глобальных проблем традиционного программирования: длительность и, следовательно, высокая стоимость разработки приложений; высокая стоимость сопровождения сложных систем; повторная используемость программ и т.п. Кроме того, объединение технологий экспертных систем и нейронных сетей с технологией традиционного программирования добавляет новые качества к коммерческим продуктам за счет обеспечения динамической модификации приложений пользователем, а не программистом, большей "прозрачности" приложения (например, знания хранятся на ограниченном естественном языке, что не требует комментариев к ним, упрощает обучение и сопровождение), лучших графических средств, пользовательского интерфейса и взаимодействия.

По мнению специалистов, в недалекой перспективе экспертные системы будут играть ведущую роль во всех фазах проектирования, разработки, производства, распределения, продажи, поддержки и оказания



услуг. Их технология, получив коммерческое распространение, обеспечит революционный прорыв в интеграции приложений из готовых интеллектуально-взаимодействующих модулей. [3 9]

Коммерческий рынок продуктов искусственного интеллекта в мире в 2003 году оценивался примерно в 0,9 млрд. долларов; из них 600 млн. приходится на долю США. Выделяют несколько основных направлений этого рынка:

- 1) экспертные системы; теперь их часто обозначают еще одним термином - "системы, основанные на знаниях";
- 2) нейронные сети и "размытые" (fuzzy) логики;
- 3) естественно-языковые системы.

В США в 2003 году рынок между этими направлениями распределился так: экспертные системы - 62%, нейронные сети - 26%, естественно-языковые системы - 12%. Рынок этот можно разделить и иначе: на системы искусственного интеллекта (приложения) и инструментальные средства, предназначенные для автоматизации всех этапов существования приложения. В 2003 году в общем объеме рынка США доля приложений составила примерно две, а доля инструментария - примерно одну треть.

Одно из наиболее популярных направлений последних пяти лет связано с понятием автономных агентов. Их нельзя рассматривать как "подпрограммы", -это скорее прислуга, даже компаньон, поскольку одной из важнейших их отличительных черт является автономность, независимость от пользователя. Идея агентов опирается на понятие делегирования своих функций. Другими словами, пользователь должен довериться агенту в выполнении определенной задачи или класса задач. Всегда существует риск, что агент может что-то перепутать, сделать что-то не так. Следовательно, доверие и риск должны быть сбалансированными. Автономные агенты позволяют существенно повысить производительность

работы при решении тех задач, в которых на человека возлагается основная нагрузка по координации различных действий.

В том, что касается автономных (интеллектуальных) агентов, хотелось бы отметить один весьма прагматический проект, который сейчас ведется под руководством профессора Генри Либермана в Media-лаборатории МИТ (MIT Media Lab). Речь идет об агентах, отвечающих за автоматическое генерирование технической документации. Для решения этой задачи немало сделал в свое время академик Андрей Петрович Ершов, сформулировавший понятие деловой прозы как четко определенного подмножества естественного языка, которое может быть использовано, в частности, для синтеза технической документации (это одно из самых узких мест в любом производстве). Группа под руководством профессора Либермана исследует возможности нового подхода к решению этой проблемы, теперь уже на основе автономных агентов.

Следующее направление в области искусственной жизни - генетическое программирование (genetic programming) - является попыткой использовать метафору генной инженерии для описания различных алгоритмов. Строки (string) искусственной "генетической" системы аналогичны хромосомам в биологических системах. Законченный набор строк называется структурой (structure). Структуры декодируются в набор параметров, альтернативы решений или точку в пространстве решений. Строки состоят из характеристик, или детекторов, которые могут принимать различные значения. Детекторы могут размещаться на разных позициях в строке. Все это сделано по аналогии с реальным миром. В природных системах полный генетический пакет называется генотипом. Организм, который образуется при взаимодействии генотипа с окружающей средой, носит название фенотипа. Хромосомы состоят из генов, которые могут принимать разные значения. (Например, ген цвета для глаза животного может иметь значение "зеленый" и позицию 10).

В генетических алгоритмах роль основных строительных блоков

играют строки фиксированной длины, тогда как в генетическом программировании эти строки разворачиваются в деревья, столь знакомые специалистам в области трансляции.

Ныне одним из лидеров в области генетического программирования является группа исследователей из Стэнфордского университета (Stanford University), работающая под руководством профессора Джона Коза. Генетическое программирование вдохнуло новую жизнь в язык Lisp, который создавался группой Джона Маккарти (того самого, кто в 60-е годы ввел в наш обиход термин "искусственный интеллект") как раз для обработки списков и функционального программирования. Кстати, именно этот язык в США был и остается одним из наиболее распространенных языков программирования для задач искусственного интеллекта[31].

Среди специализированных систем, основанных на знаниях, наиболее значимы экспертные системы реального времени, или динамические экспертные системы. На их долю приходится 70 процентов этого рынка.

Значимость инструментальных средств реального времени определяется не столько их бурным коммерческим успехом в первую очередь, тем, что только с помощью подобных средств создаются стратегически значимые приложения в таких областях, как управление непрерывными производственными процессами в химии, фармакологии, производстве цемента, продуктов питания и т.п., аэрокосмические исследования, транспортировка и переработка нефти и газа, управление атомными и тепловыми электростанциями, финансовые операции, связь и многие другие.

Рассмотрим основные направления развития искусственного интеллекта.

1. Представление знаний и разработка систем, основанных на знаниях.

Это направление связано с разработкой моделей представления знаний, созданием баз знаний, образующих ядро экспертных

систем. В последнее время это направление включает в себя модели и методы извлечения и структурирования знаний и сливается с инженерией знаний, где определяется соотношение человеческого знания и его формализованного (информационного) отображения в ЭВМ.

## 2. Игры и творчество.

Традиционно искусственный интеллект включает в себя игровые интеллектуальные задачи - шахматы, шашки. В основе этого направления лежит лабиринтная модель и эвристики. В настоящее время это скорее коммерческое направление, чем научное.

## 3. Разработка естественно-языковых интерфейсов и машинный перевод.

Одной из популярных тем исследований искусственного интеллекта является область машинного перевода. Первая программа в этой области - переводчик с английского языка на русский. Первая идея - пословный перевод, оказалась не плодотворной. В настоящее время используется более сложная модель включающая анализ и синтез естественно-языковых сообщений.

## 4. Распознавание образов.

Это традиционное направление искусственного интеллекта, берущее начало у самых его истоков. Каждому объекту ставится в соответствие матрица признаков, по которой происходит его распознавание. Это направление близко к машинному обучению.

## 5. Новые архитектуры компьютеров.

Это направление занимается разработкой новых аппаратных решений и архитектур, направленных на обработку символьных и логических данных. Создаются Пролог - и Лисп - машины, компьютеры V и VI поколений. Последние разработки посвящены компьютерам баз данных и знаний и параллельным компьютерам - транспьютеры, с большим количеством процессоров.

## 6. Интеллектуальные роботы.

Роботы - это электромеханические устройства, предназначенные для автоматизации человеческого труда.

Идея создания роботов исключительно древняя. Со времени создания сменилось несколько поколений роботов.

Роботы с жесткой схемой управления. Практически все современные промышленные роботы принадлежат к первому поколению. Фактически это программируемые манипуляторы.

Адаптивные роботы с сенсорными устройствами. В настоящее время есть образцы таких роботов, но в промышленности они пока не используются.

Самоорганизующиеся, или интеллектуальные, роботы. Это конечная цель развития робототехники. Основная проблема при создании интеллектуальных роботов - проблема машинного зрения.

## 7. Специальное программное обеспечение.

В рамках этого направления разрабатываются специальные языки для решения задач не вычислительного типа. Эти языки ориентированы на символьную обработку информации - LISP, PROLOG, SMALTALK, РЕФАЛ и др. Помимо этого создаются пакеты прикладных программ, ориентированные на промышленную разработку интеллектуальных систем, например KEE, ARTS. Достаточно популярно создание так называемых пустых экспертных систем, или «оболочек», - EXSYS, MI и др., в которых можно наполнять базы знаний, создавая различные системы.

## 8. Обучение и самообучение.

Это активно развивающаяся область искусственного интеллекта. Включает модели, методы и алгоритмы, ориентированные на автоматическое накопление знаний на основе анализа и обобщения данных, а также обучение по примерам.

## 9. Нейрокибернетика.

Основную идею нейрокибернетики можно сформулировать следующим образом. Единственный объект, способный мыслить, - это человеческий мозг. Поэтому любое «мыслящее» устройство должно каким-то образом воспроизводить его структуру. Таким образом нейрокибернетика ориентирована на аппаратное моделирование структур, подобных структуре мозга. Физиологами давно установлено, что основой человеческого мозга является большое количество (до  $10^{21}$ ) связанных между собой и взаимодействующих нервных клеток - нейронов. Поэтому усилия нейрокибернетики были сосредоточены на создании элементов, аналогичных нейронам, и их объединении в функционирующие системы. Эти системы принято называть нейронными сетями или нейросетями[15].

В настоящее время создаются компьютеры VI поколения - нейрокомпьютеры, моделирующие структуру мозга человека.

## 10.2. Нейросетевые технологии.

На рынке коммерческих программных продуктов наряду с аналитическими инструментами нового поколения, основанными на применении логики нечетких множеств - от электронных таблиц до экспертных систем все больше интерес представляют аналитические информационные технологии, основанные на использовании нейронных сетей. **Нейронные сети** - это обобщенное название групп алгоритмов, которые умеют обучаться на примерах, извлекая скрытые закономерности из потока данных. Компьютерные технологии, получившие названия нейросетевых, работают по аналогии с принципами строения и функционирования нейронов головного мозга человека и позволяют решать чрезвычайно широкий круг задач: распознавание человеческой речи и абстрактных образов, классификацию состояний сложных систем, управление технологическими процессами, решением аналитических, исследовательских,

прогнозных задач, связанных с обширными информационными потоками. Являясь мощным технологическим инструментом, нейросетевые технологии облегчают специалисту процесс принятия важных и неочевидных решений в условиях неопределенности, дефицита времени и ограниченных информационных ресурсов[14].

В настоящее время одним из лидеров стал нейросетевой пакет Brain Maker американской фирмы California Scientific Software.

Свой путь на российский рынок нейронные сети начали с финансово-кредитной сферы, где заинтересованные в совершенствовании аналитической работы банки стали интенсивно включать нейронные сетевые технологии в состав финансовых приложений.

Отличительной чертой нейронных сетей является их способность менять свое поведение (обучаться) в зависимости от изменения внешней среды, извлекая скрытые закономерности из потока данных. При этом алгоритмы обучения не требуют каких-либо предварительных знаний о существующих в предметной области взаимосвязях - необходимо только подобрать достаточное число примеров, описывающих поведение моделируемой системы в прошлом. Основанная на нейросетях технология не предъявляет повышенных требований к точности входных данных как на этапе обучения, так и при ее использовании, например при распознавании симптомов приближения критических ситуаций, для краткосрочных, а иногда и долгосрочных прогнозов.

Таким образом, нейросетевая технология обладает двумя чрезвычайно полезными свойствами.

1. Способностью обучаться на конкретном множестве примеров.

2. Умение стабильно распознавать, прогнозировать новые ситуации с высокой степенью точности, причем в условиях внешних помех, например, появления противоречивых или неполных знаний

в потоках информации.

Взяв за основу работу мозга, нейросетевые технологии включили в себя ряд биологических терминов, понятий, параметров, а метод получил название генетического алгоритма.

Генетический алгоритм реализован в популярных версиях нейропакетов - широко известном Brain Maker Professional v.3.11 и менее известном, но более профессиональном Neuroforester v.5.1.

Для реализации нейросетевой технологии должны быть выполнены следующие условия: наличие IBM PC или совместимого компьютера, мыши, MS Windows 3.1 или выше, 4 Мбайт RAM (оперативной памяти).

В отличие от Brain Maker в пакете Neuroforester для решения прогнозных задач ряд процедур выполняется автоматически. В частности, автоматически выбирается оптимальное число дней, обеспечиваемых прогнозом. Пакет имеет также инструменты для предварительной обработки данных корреляционной анализ, позволяющий определять значимость входных параметров прогноза; анализ с помощью масштабных преобразований для выявления скрытых циклов данных; диаграмма - распределение зависимости прогнозируемой величины от входных параметров. Эти методы позволяют уже на этапе подготовки данных выделять наиболее существенные для прогноза параметры. Все результаты обработки представляются в графическом виде, удобном для анализа и принятия решений.

При использовании нейросетевой технологии работа строится в несколько этапов.

Первым этапом является четкое определение проблемы, т.е. того, что пользователь - аналитик собирается получить от нейросетевой технологии на выходе.

Вторым этапом является определение и подготовка исходных данных для реализации нейросетевой технологии. При этом отбира-



ется вся необходимая, адекватно и полно описывающая процесс информация.

Сложность выполнения второго этапа заключается в том, что должен быть соблюден баланс между стремлением увеличить количество входных параметров и вероятностью получить плохо обучаемую сеть, которая может исказить ожидаемые прогнозы.

Ввод данных в систему, подготовка данных, создание файлов для тестирования можно считать самостоятельным третьим этапом. На этапе подготовки данных анализируется степень их информационной насыщенности, для чего выявляется степень влияния конкретного параметра на прогнозируемую величину. Достигнув равномерного наполнения всех степеней зависимости, выявляется соответствие между прогнозируемой величиной и параметром в виде «Если..., то...; иначе...», что близко к экспертным системам.

На четвертом этапе осуществляется выбор типа нейросетевой технологии и метода ее обучения. Сеть может быть построена с помощью Net Maker в интерактивном режиме, пользуясь его подсказками, или создать файлы Brain Maker, пользуясь текстовым редактором. Наиболее трудоемким процессом является настройка нейросети на обучающую выборку данных, ибо здесь определяется оптимальное количество параметров, свойств исследуемых данных, оптимальное число дней ретроспективы и прогноза.

Необходимо отметить то, что все современные нейросетевые технологии содержат ту или иную систему конвертеров, позволяющих пользоваться данными, подготовленными в популярных исходных форматах, т.е. преобразовывать набор символов (слов и фраз) в уникальный набор чисел. В частности, Word System делает возможной также обратную операцию, т.е. представление результатов работы нейросети в виде не только чисел, но и связного текста, что позволяет генерировать результаты в виде различных информаци-

онных сообщений. Правила для обучения нейросети могут задаваться посредством их ввода в готовом виде, а также в виде чисел, требующих дополнительных преобразований данных. Причем эти ограничивающие и разрешающие правила и условия могут задаваться в процессе решения задачи.

Последними этапами можно считать проведение тестирования нейросети и ее запуск для получения прогноза. Если результаты тестирования не удовлетворяют, то просматривают набор входных данных, изменяют некоторые учебные программы или перестраивают сеть [14].

Гибкость и мощность нейронных сетей открывают перед ними практически неограниченные возможности применения, особенно в качестве аналитических инструментов в таких плохо формализуемых и многокритериальных областях, как анализ финансовой и банковской деятельности.

Среди перспективных направлений использования нейросетевых технологий можно назвать создание компьютерных моделей поведения клиента для оценки риска или перспективности работы с конкретными клиентами.

На мировом рынке аналитического программного обеспечения представлен широкий спектр нейросетевых технологий, начиная от систем, ориентированных на суперкомпьютеры, стоимость которых превышает 50 тыс. долл., до недорогих (несколько сотен долларов) нейропакетов, работающих на платформе персональных компьютеров и рабочих станций. Это делает доступной технологию нейронных сетей для приложений практически любого уровня. Ее массовое применение - вопрос ближайшего будущего.

### **Краткие выводы.**

В данной теме представлены перспективы развития интеллектуальных систем, рассмотрены основные направления их развития. Кроме того, дана характеристика одному из наиболее популярных направлений – генетическому программированию. Дано понятия нейронных сетей и нейросетевых технологий, описаны их свойства и основные этапы использования.

### **Ключевые слова:**

Знания, система, модели, база знаний, структурирование знаний, инженерия знаний, игры, творчество, интерфейс, машинный перевод, распознавание образов, матрица признаков, машинное обучение, архитектура, компьютер, роботы, схема управления, сенсорные устройства, программное обеспечение, обучение, нейрокибернетика, человеческий мозг, нейронные сети, принятие решений, прогноз, точность, внешние помехи, генетический алгоритм, корреляционный анализ, гибкость, мощность, стоимость.

### **Вопросы для обсуждения и контроля.**

- 1) Перечислите основные направления развития искусственного интеллекта?
- 2) Охарактеризуйте основные направления развития искусственного интеллекта?
- 3) Что такое нейронные сети?
- 4) По какому принципу разрабатываются нейросетевые технологии?
- 5) Какими основными свойствами обладают нейросетевые технологии?
- 6) Что такое генетический алгоритм?
- 7) Перечислите основные этапы использования нейросетевых

технологий?

8) Охарактеризуйте основные этапы использования нейросетевых технологий?

### **Литература.**

1. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления: Учебник. 2-е изд. (+CD). - СПб.: Питер, 2009. -120 с.

2. Информационные технологии управления. Учебное пособие для вузов / Под ред. проф. ГА Титоренко. — 2-е изд., доп. — М: ЮНИТИ-ДАНА, 2007. - 439 с.

3. Информатика: Учебник/ Под ред. Проф. Н.В. Макаровой. 3-е изд. -М.: Финансы и статистика, 2006. -768с.

4. Автоматизированные информационные технологии в экономике: Учебник/ Под ред. проф. Г.А.Титоренко. - М.: Юнити, 2006г. - 399 с.

5. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Учебник, М.: Финансы и статистика, 2004 г. - 424с.

6. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н. Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С. - Ташкент, 2003. – 135 с.

7. Исаев Г.Н. Информационные системы в экономике. Учебное пособие. М: Омега – Л, 2006г. – 462 с.

### **ГЛОССАРИЙ**

**Автоматизированные обучающие системы (АОС)** - это специальный вид интеллектуальных систем, базирующихся на знаниях и получившей широкое распространение с развитием и появлением рынка ПЭВМ. АОС, как правило, один или несколько языков, на которых описываются действия и структуры, используемые для

функционирования обучающей системы. АОС ориентированы на работу с двумя группами пользователей.

**Автономные ЭС** работают непосредственно в режиме консультаций с пользователем для специфически «экспертных» задач, для решения некоторых не требуется привлекать традиционные методы обработки данных (расчеты, моделирование и т.д.).

**Алгоритм** – это формальная процедура, которая гарантирует получение оптимального решения.

**База знаний** – это совокупность моделей, правил и факторов (данных), порождающих анализ и вывод для нахождения решений сложных задач в некоторой предметной области.

**Генетическое программирование** (genetic programming) - является попыткой использовать метафору генной инженерии для описания различных алгоритмов.

**Гибридные ЭС** представляют программный комплекс, агрегирующий стандартные пакеты прикладных программ (например, математическую статистику, линейное программирование или системы управления базами данных) и средства манипулирования знаниями. Это может быть интеллектуальная надстройка над ППП или интегрированная среда для решения сложной задачи с элементами экспертных знаний.

**Диагностика** - обнаружение неисправности в некоторой системе.

**Динамические ЭС** работают в сопряжении с датчиками объектов в режиме реального времени с непрерывной интерпретацией поступаемых данных.

**Диспетчер** – это часть механизма вывода, которая решает когда и в каком порядке применять правила из предметных знаний.

**Задача анализа**- это задача , которая осуществляет доопределение или переформулирование текущего состояния предметной

области.

**Задачи преобразования** - это задачи, при решении которых одно состояние предметной области преобразуется в другое.

**Задачи определения предметной области**- это задачи, в ходе решения которых система переходит из одной альтернативной области в другую.

**Знание** – это интеллектуальная информация, используемая в программе.

**Инженер знаний** – это человек, имеющий навыки в разработке искусственного интеллекта и знающий как надо строить экспертную систему. Он опрашивает эксперта, организует знания в БЗ.

К **инструментальным средствам** проектирования относят язык построения экспертной системы и поддерживающие средства, через которые пользователь взаимодействует с экспертной системой.

**Интерпретатор** – это часть механизма вывода, которая решает каким образом применять предметные знания.

**Интерпретация** - определение смысла данных, результаты которого должны быть согласованными и корректными.

**Интерфейс** – это часть системы БЗ, которая обеспечивает работу с БЗ, которая обеспечивает работу с БЗ и механизмом получения решений на языке достаточно высокого уровня, приближенного к профессиональному языку специалистов в той предметной области, к которой относится БЗ.

**Искусственный интеллект** – это некая система программных средств, имитирующая на компьютере процесс мышления человека.

**Механизм вывода** – это часть экспертной системы, содержащая в себе общие сведения о схеме управления процессом решения задачи.

**Логическое программирование** - это один из подходов к

информатике, при котором в качестве языка высокого уровня используется логика предикатов первого порядка в форме фраз Хорна.

**Логика** предикатов первого порядка - это универсальный абстрактный язык предназначенный для представления знаний и для решения задач.

**Квазидинамические ЭС** интерпретируют ситуацию, которая меняется с некоторым фиксированным интервалом времени.

**Мониторинг** - непрерывная интерпретация данных в реальном масштабе времени и сигнализация о выходе тех или иных параметров за допустимые пределы.

**Нейронные сети** - это обобщенное название групп алгоритмов, которое умеют обучаться на примерах, извлекая скрытые закономерности из потока данных.

**Коэффициент уверенности** – это число, означающее вероятность или степень уверенности, с которой можно считать данные факты и правила достоверными.

Под **планированием** понимается нахождение планов действий, относящихся к объектам, способным выполнять некоторые функции.

**Правила** – это формальный способ задания знаний в виде:

ЕСЛИ <условие> , ТО <действие>

**Предметная область** – это область знаний о задачах, а проблемная область включает в себя предметную область и сами задачи, решаемые в этой области.

**Проектирование** состоит в подготовке спецификаций на создание «объектов» с заранее определенными свойствами. Под спецификацией понимается весь набор необходимых документов - чертеж, пояснительная записка и т.д.

Последовательность рассуждений называется **прямой цепочкой** потому, что отправной точкой рассуждений служит возникшая

ситуация ( перегрев двигателя), а часть правила выполняется только в том случае, если удовлетворяется условная часть правила ЕС-ЛИ. Т.е. сначала имеет место ситуация, а потом делаются выводы.

В **обратной цепочке**, наоборот, выводы известны, а необходимо найти причины, например, автомобиль не трогается с места.

**Робастность** - способность решателя постепенно снижать качество работы по мере приближения к границам заданной надежности данных.

**Роботы** - это электромеханические устройства, предназначенные для автоматизации человеческого труда.

**Слот** – это атрибут описания свойства объекта.

**Условная вероятность** – это вероятность наступления какого-то события  $S$  при условии, что уже наступило какое-то другое событие  $e$ .

**Эксперт** – это специалист, умеющий находить эффективные решения в конкретной предметной области.

**Экспертная система** – это программа, основанная на знаниях, в которой предметные знания предусмотрены в явном виде и отделены от прочих знаний.

**Эвристики** – это правила, которые упрощают или ограничивают поиск решения в предметной области.

**Фрейм** – это метод представления знаний в виде графа, где вершинами являются объекты, а дугами их свойства.



# СПИСОК РЕКОМЕНДОВАННОЙ ЛИТЕРАТУРЫ

## 1. Законы Республики Узбекистан

1. Закон Республики Узбекистан «Об электронном документообороте» // «Народное слово», 2004 г., 30 апреля.

2. Закон Республики Узбекистан «Об электронной коммерции» // «Народное слово», 2004 г., 30 апреля.

3. Закон Республики Узбекистан «Об информатизации» // «Народное слово», 2004 г., 11-февраля.

## 2. Указы и Постановления Президента Республики Узбекистан

4. Указ Президента Республики Узбекистан от 14 июня 2005 г. «О мерах по ускорению реализации приоритетных направлений в сфере углубления рыночных реформ и дальнейшей либерализации экономики». // «Народное слово», 2005 г., 15-июня.

5. Указ Президента Республики Узбекистан «О дальнейшем развитии компьютеризации и внедрении информационно-коммуникационных технологий» // «Народное слово», 2002 г., 1-июня.

## 3. Постановления Кабинета Министров Республики Узбекистан

6. Постановление Кабинета Министров Республики Узбекистан «О совершенствовании системы подготовки кадров в сфере информационных технологий» // «Народное слово», 2005 г., 3-июня.

7. Постановление Кабинета Министров Республики Узбекистан «О дальнейшем развитии компьютеризации и внедрении информационно-коммуникационных технологий». // «Народное слово», 2002 г., 8-июня.

## 4. Труды Президента Республики Узбекистан

8. Каримов И.А. Конституция Узбекистана – прочный фундамент нашего продвижения на пути демократического развития и формирования граждан-

ского общества. Доклад Президента Ислама Каримова на торжественном собрании, посвященном 17-летию Конституции Республики Узбекистан. // Народное слово, 6 декабря 2009 г. – с. 1-3

9. Каримов И.А. Наша главная задача – дальнейшее развитие страны и повышение благосостояния народа. Доклад Президента Республики Узбекистан Ислама Каримова на заседании Кабинета Министров, посвященном итогам социально-экономического развития страны в 2009 году и важнейшим приоритетам экономической программы на 2010 год. // Народное слово, 30 января 2010 г. – с. 1-3.

10. Каримов И.А. Мировой финансово-экономический кризис, пути и меры по его преодолению в условиях Узбекистана /И.А.Каримов. Т.: Узбекистан, 2009.-48с.

## **5. Нормативно-правовые документы Министерств Республики Узбекистан**

11. Олий таълим. Меъёрий хужжатлар тўплами: /С.С.Фулумов тахрири остида; Тузувчилар: Б.Х.Рахимов, Ш.Д.Жонбоев ва бошқ. – Т.: «Шарқ», 2001. – 672 б.

12. Положение о порядке создания и использования электронных баз данных в государственных организациях республики. Утверждено Государственным комитетом Республики Узбекистан по науке и технике, от 29.20.1995.

13. Положение о порядке и правилах создания, внедрения и эксплуатации локальных, ведомственных, региональных и других информационно-вычислительных сетей на территории республики. Утверждено Государственным комитетом Республики Узбекистан по науке и технике, от 30.01.1995.

## **6. Учебники**

14. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления. Учебник. 2-е изд. (+CD).- СПб.: Питер, 2009.-320 с.

15. Автоматизированные информационные технологии в экономике. Учебник/ Под ред. проф. Г.А.Титоренко.- М.: Юнити, 2006г. - 399 с.

16. Информатика для экономистов. Учебник / под общ. Ред. В.М. Мапошка. – М.: ИНФРА-М, 2006. – 880 с.
17. Информатика. Учебник. - 3-е перераб. Изд./ Под ред. Н.В. Макаровой. –И74 М.: Финансы и статистика, 2006, - 768 с.
18. Балдин К.В., Уткин В.Б. Информационные системы в экономике. Учебник. – М.: Издательско-торговая корпорация «Дашков и К», 2005. - 395 с.
19. Мищенко А.И. Теория экономических информационных систем: Учебник. - 4-е изд., доп. И перераб. – М.: Финансы и статистика, 2005. - 240 с.
20. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. Учебник. - М.: Финансы и статистика, 2004. - 250 с.
21. Уткин В.Б., Балдин К.В. Информационные системы и технологии в экономике. Учебник. – М.: Юнити – Дана, 2003. – 335 с.
22. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных . Учебник для высших учебных заведений /Под ред. проф. А.Д. Хомоненко. изд. втор., перераб. Спб.: КОРОНАПринт, 2002. – 672 с.
23. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. Учебное пособие. - СПб.: Питер, 2001. -384 с.

## **7. Учебные пособия**

24. Информационные системы и технологии в экономике и управлении. Под ред. проф. В.В. Трофимова. Учебное пособие. М. Высшее образование, 2006 г., 480с.
25. Исаев Г.Н. Информационные системы в экономике. Учебное пособие. М: Омега – Л, 2006г. – 462 с.
26. Ивасенко А.Г., А.Ю. Гридасов, В.А. Павленко. Информационные технологии в экономике и управлении. Учебное пособие. М. Кнорус, 2005г.– 160 с.
27. Попов В.Б. Основы информационных и телекоммуникационных технологий. Учеб пособие. М.: Финансы и статистика, 2005г. – 224с.

28. А.Н. Мелихов, Л.С. Бернштейн, С.Я. Коровин. Ситуационные советующие системы с нечеткой логикой. - М.: Наука, 2004г.
29. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А.А., Пилипенко Е.Ф., Рузметова Х.Н. Введение в базы данных и знаний. Учебное пособие. / Под редакцией акад. Гулямова С.С.- Ташкент, 2003. – 135 с.
30. Джексон Питер. Введение в экспертные системы. Пер. с англ. Уч. пос. - М.: Издательский дом «Вильямс» 2001. – 624 с.
31. Дейт К. Введение в системы баз данных 7-изд./Пер. с англ. М.: Издательский дом «Вильямс», 2001. – 1072 с.

### **8. Монографии и научные статьи**

32. Интернет-порталы: содержание и технологии: Сб. науч. Ст. Вып.1/Редкл.: А.Н. Тихонов (преп) и др.; ГНИИ ИТТ «Информика». – М.: Просвещение, 2003.
33. Титарев Д.Л., Титатев Л.Г., Феданов А.Н. Портал университета// В сб.: Интернет-порталы: содержание и технологии: Сб. науч. Ст. Вып.1/ Редкол.: А.Н. Тихонов (пред.) и др.; ГНИИ ИТТ «Информика». – М.: Просвещение, 2003.
34. Бегалов Б.А. Технология процессов формирования информационно-коммуникационного рынка. Монография. – Ташкент: Фан, 2000.

### **9. Докторские и кандидатские диссертации**

35. Охунов Д.М. Исследование и разработка маркетинговых автоматизированных информационных систем предприятий. Диссертация на соискание кандидата экономических наук. - Ташкент, ТГЭУ, 2005, 138 с.
36. Бегалов Б.А. Ахборот-коммуникациялар бозорининг шаклланиш ва ривожланиш тенденцияларини эконометрик моделлаштириш. Иктисод фанлари доктори илмий даражаси даъвогарлигига диссертация иши. - Тошкент, ТДИУ, 2001, 330 б.

## **10. Сборник статей научно-практических конференции**

37. «Ахборот-коммуникациялар технологиялари асосида электрон ўқув адабиётларини яратиш: тажриба, муаммо ва истиқболлар» мавзусидаги республика илмий-амалий анжумани, Тошкент, 2004 йил, 28-апрель.

38. «Етук мутахассисларни тайёрлашда замонавий педагогик технологиялар ва интерактив усулларнинг самарадорлиги» мавзусидаги II анжуман маърузалари тезислари. Тошкент, 2003 йил.

39. "Иқтисодчи кадрлар тайёрлаш сифатини таъминлашда ахборот-коммуникациялар технологиялари", Республика илмий-амалий анжумани, Тошкент, 2003, 15-16 май.

## **11. Газеты и журналы.**

40. Информационные ресурсы России.

41. Информационные технологии.

42. Ўзбекистон иқтисодий ахборотномаси.

## **12. Сборник статистических данных.**

43. Мониторинг развития информационно-коммуникационных технологий в Узбекистане. 2003 – 2005 гг. Ташкент. 70 с.

44. Деловая среда в Узбекистане глазами представителей малого бизнеса. Ташкент. 2002-2005 гг. 170 с.

## **13. Интернет сайты**

45. <http://www.iite.ru> – сайт ЮНЕСКО «Информационные технологии в образовании».

46. <http://diamond.stup.ac.ru/ENG/F4/DIRECT/4.html> – российский образовательный сайт «Новые информационные технологии в образовании».

47. <http://www.microsoft.com/ru> – главный сервер фирмы Microsoft.

48. <http://www.sph.Runnet.ru> – сервер Санкт-Петербургского центра научно-технической информации.

49. <http://www.cctpu.edu.ru> – сайт кафедры «Информатики и проектирования систем» Томского Политехнического Университета.

50. <http://www.cnews.ru> – сайт изданий о высоких технологиях.

#### **14. Электронные учебники и учебные пособия виртуальной библиотеки.**

51. Б.А.Бегалов. Введение в базы данных. Электронное учебное пособие. Программисты: А.Бобожонов, У.Муслимов. - Ташкент. ТГЭУ. 2004 г.

52. Ахборот тизимлари ва технологиялари. Электрон дарслик. Компьютер дастурчилари: С.Аҳмедов, А.Рапопорт. - Тошкент. ТДИУ. 2003 й.

53. С.С.Ғуломов, А.Т.Шермухамедов, Б.А.Бегалов. Иқтисодий информатика. Электрон дарслик. Компьютер дастурчилари: О.Сидиқов, С.Аҳмедов. - Тошкент. ТДИУ. 2002 й.