

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

**Б1.В.ДВ.15.02 Системы искусственного интеллекта –**

**Направление подготовки 38.03.02 Менеджмент**

**Профиль образовательной программы Маркетинг**

**Форма обучения очная**

## СОДЕРЖАНИЕ

<b>1. Конспект лекций</b> .....	3
<b>1.1 Лекция № 1</b> <i>Введение в интеллектуальные системы</i> .....	3
<b>1.2 Лекция № 2</b> <i>Разработка систем, основанных на знаниях</i> .....	6
<b>1.3 Лекция № 3</b> <i>Теоретические аспекты инженерии знаний</i> .....	13
<b>1.4 Лекция № 4</b> <i>Технологии инженерии знаний</i> .....	15
<b>1.5 Лекция № 5</b> <i>Перспективы развития СИИ</i> .....	19
<b>2. Методические указания по проведению практических занятий</b> .....	23
<b>2.1 Практическое занятие № ПЗ-1</b> <i>Введение в интеллектуальные системы</i> .....	23
<b>2.2 Практическое занятие № ПЗ-2</b> <i>Разработка систем, основанных на знаниях</i> .....	23
<b>2.3 Практическое занятие № ПЗ-3</b> <i>Теоретические аспекты инженерии знаний</i> .....	23
<b>2.4 Практическое занятие № ПЗ-4</b> <i>Технологии инженерии знаний</i> .....	24
<b>2.5 Практическое занятие № ПЗ-5</b> <i>Перспективы развития СИИ</i> .....	24

# 1. КОНСПЕКТ ЛЕКЦИЙ

## Лекция № 1

### Тема: «Введение в интеллектуальные системы»

#### 1. Вопросы лекции:

1. Основные понятия и определения
2. Область применения
3. Краткий исторический обзор развития работ в области ИИ
4. Функциональная структура использования СИИ

#### 2. Краткое содержание вопросов

##### 2.1. Основные понятия и определения

Искусственный интеллект (ИИ) - это наука о концепциях, позволяющих ВМ делать такие вещи, которые у людей выглядят разумными. Но что же представляет собой интеллект человека? Есть ли эта способность размышлять? Есть ли эта способность усваивать и использовать знания? Есть ли эта способность оперировать и обмениваться идеями? Несомненно, все эти способности представляют собой часть того, что является интеллектом. На самом деле дать определение в обычном смысле этого слова, по-видимому, невозможно, потому что интеллект - это сплав многих навыков в области обработки и представления информации.

Центральные задачи ИИ состоят в том, что бы сделать ВМ более полезными и чтобы понять принципы, лежащие в основе интеллекта. Поскольку одна из задач состоит в том, чтобы сделать ВМ более полезными, ученым и инженерам, специализирующимся в вычислительной технике, необходимо знать, каким образом ИИ может помочь им в разрешении трудных проблем.

##### 2.2. Область применения

- Доказательства теорем;
- Игры;
- Распознавание образов;
- Принятие решений;
- Адаптивное программирование;
- Сочинение машинной музыки;
- Обработка данных на естественном языке;
- Обучающиеся сети (нейросети);
- Вербальные концептуальные обучения.

Планы на будущее в области применения ИИ: В сельском хозяйстве компьютеры должны оберегать посевы от вредителей, подрезать деревья и обеспечивать избирательный уход. В горной промышленности компьютеры призваны работать там, где возникают слишком опасные условия для людей. В сфере производства ВМ должны выполнять различного вида задачи по сборке и техническом контроле. В учреждениях ВМ обязаны заниматься составлением расписаний для коллективов и отдельных людей, делать краткую сводку новостей. В учебных заведениях ВМ должны рассматривать задачи, которые решают студенты, в поисках ошибок, подобно тому как ищутся ошибки в программе, и устранять их. Они должны обеспечивать студентов суперкнигами, хранящимися в памяти вычислительных систем. В больницах ВМ должны помогать ставить диагноз, направлять больных в соответствующие отделения, контролировать ход лечения. В домашнем хозяйстве ВМ должны помогать советами по готовке пищи, закупке продуктов, следить за состоянием пола в квартире и газона в саду. Конечно, в настоящее

время ни одна из этих вещей не представляется возможной, но исследования в области ИИ могут способствовать их реализации.

### **2.3. Краткий исторический обзор развития работ в области ИИ**

Начало исследований в области ИИ (конец 50-х годов) связывают с работами Ньюэлла, Саймана и Шоу, исследовавших процессы решения различных задач. Результатами их работ явились такие программы как "ЛОГИК-ТЕОРЕТИК", предназначенная для доказательства теорем в исчислении высказываний, и "ОБЩИЙ РЕШАТЕЛЬ ЗАДАЧ". Эти работы положили начало первому этапу исследований в области ИИ, связанному с разработкой программ, решающих задачи на основе применения разнообразных эвристических методов.

Эвристический метод решения задачи при этом рассматривался как свойственный человеческому мышлению "вообще", для которого характерно возникновение догадок о пути решения задачи с последующей проверкой их. Ему противопоставлялся используемый в ЭВМ алгоритмический метод, который интерпретировался как механическое осуществление заданной последовательности шагов, детерминированно приводящей к правильному ответу. Трактовка эвристических методов решения задач как сугубо человеческой деятельности и обусловила появление и дальнейшее распространение термина ИИ. Так, при описании своих программ Ньюэлл и Саймон приводили в качестве доводов, подтверждающих, что их программы моделируют человеческое мышление, результаты сравнения записей доказательств теорем в виде программ с записями рассуждения <думающего вслух> человека. В начале 70-х годов они опубликовали много данных подобного рода и предложили общую методику составления программ, моделирующих мышление. Примерно в то время, когда работы Ньюэлла и Саймона стали привлекать к себе внимание, в Массачусетском технологическом институте, Стэнфордском университете и Стэнфордском исследовательском институте также сформировались исследовательские группы в области ИИ. В противоположность ранним работам Ньюэлла и Саймона эти исследования больше относились к формальным математическим представлениям. Способы решения задач в этих исследованиях развивались на основе расширения математической и символической логики. Моделированию же человеческого мышления придавалось второстепенное значение.

На дальнейшие исследования в области ИИ большое влияние оказало появление метода резолюций Робинсона, основанного на доказательстве теорем в логике предикатов и являющегося исчерпывающим методом доказательства. При этом определение термина ИИ претерпело существенное изменение. Целью исследований, проводимых в направлении ИИ, стала разработка программ, способных решать "человеческие задачи". Так, один из видных исследователей ИИ того времени Р. Бенерджи в 1969 году писал: "Область исследований, обычно называемую ИИ, вероятно, можно представить как совокупность методов и средств анализа и конструирования машин, способных выполнять задания, с которыми до недавнего времени мог справиться только человек. При этом по скорости и эффективности машины должны быть сравнимы с человеком." Функциональный подход к направленности исследований по ИИ сохранился в основном до настоящего времени, хотя еще и сейчас ряд ученых, особенно психологов, пытаются оценивать результаты работ по ИИ с позиций их соответствия человеческому мышлению.

Исследовательским полигоном для развития методов ИИ на первом этапе явились всевозможные игры, головоломки, математические задачи. Некоторые из этих задач стали классическими в литературе по ИИ (задачи об обезьяне и бананах, миссионерах и людоедах, Ханойской башне игра в 15 и другие). Выбор таких задач обуславливался простотой и ясностью проблемной среды (среды, в которой разворачивается решение задачи), ее относительно малой громоздкостью, возможностью достаточно легкого подбора и даже искусственного конструирования "под метод". Основной расцвет такого рода исследований приходится на конец 60-х годов, после чего стали делаться первые

попытки применения разработанных методов для задач, решаемых не в искусственных, а в реальных проблемных средах. Необходимость исследования систем ИИ при их функционировании в реальном мире привело к постановке задачи создания интегральных роботов. Проведение таких работ можно считать вторым этапом исследований по ИИ. В Стэнфордском университете, Стэнфордском исследовательском институте и некоторых других местах были разработаны экспериментальные роботы, функционирующие в лабораторных условиях. Проведение этих экспериментов показало необходимость решения кардинальных вопросов, связанных с проблемой представления знаний о среде функционирования, и одновременно недостаточную исследованность таких проблем, как зрительное восприятие, построение сложных планов поведения в динамических средах, общение с роботами на естественном языке. Эти проблемы были более ясно сформулированы и поставлены перед исследователями в середине 70-х гг, связанных с началом третьего этапа исследований систем ИИ. Его характерной чертой явилось смещение центра внимания исследователей с создания автономно функционирующих систем, самостоятельно решающих в реальной среде поставленные перед ними задачи, к созданию человеко-машинных систем, интегрирующих в единое целое интеллект человека и способности ВМ для достижения общей цели - решение задачи, поставленной перед интегральной человеко-машинной решающей системой. Такое смещение обуславливалось двумя причинами:

- К этому времени выяснилось, что даже простые на первый взгляд задачи, возникающие перед интегральным роботом при его функционировании в реальном времени, не могут быть решены методами, разработанными для экспериментальных задач специально сформированных проблемных средах;
- Стало ясно, что сочетание дополняющих друг друга возможностей человека и ЭВМ позволяет обойти острые углы путем перекладывания на человека тех функций, которые пока еще не доступны для ЭВМ. На первый план выдвигалась не разработка отдельных методов машинного решения задач, а разработка методов средств, обеспечивающих тесное взаимодействие человека и вычислительной системы в течение всего процесса решения задачи с возможностью оперативного внесения человеком изменений в ходе этого процесса.

Развитие исследований по ИИ в данном направлении обуславливалось также резким ростом производства средств вычислительной техники и также резким их удешевлением, делающим их потенциально доступными для более широких кругов пользователей.

## 2.4. Функциональная структура использования СИИ



Эта структура состоит из трех комплексов вычислительных средств (см. рисунок). Первый комплекс представляет собой совокупность средств, выполняющих программы (исполнительную систему), спроектированных с позиций эффективного решения задач, имеет в ряде случаев проблемную ориентацию. Второй комплекс - совокупность средств интеллектуального интерфейса, имеющих гибкую структуру, которая обеспечивает возможность адаптации в широком спектре интересов конечных пользователей. Третьим комплексом средств, с помощью которых организуется взаимодействие первых двух, является база знаний, обеспечивающая использование вычислительными средствами первых двух комплексов целостной и независимой от обрабатываемых программ системы знаний о проблемной среде. Исполнительная система (ИС) объединяет всю совокупность средств, обеспечивающих выполнение сформированной программы. Интеллектуальный интерфейс - система программных и аппаратных средств, обеспечивающих для конечного пользователя использование компьютера для решения задач, которые возникают в среде его профессиональной деятельности либо без посредников либо с незначительной их помощью. База знаний (БЗ) - занимает центральное положение по отношению к остальным компонентам вычислительной системы в целом, через БЗ осуществляется интеграция средств ВС, участвующих в решении задач.

## **Лекция № 2**

### **Тема: «Разработка систем, основанных на знаниях»**

#### **1. Вопросы лекции:**

1. Предметные области для Экспертных систем
2. Обобщенная структура ЭС. Основные понятия и определения
3. Классификация ЭС
4. Инструментальные средства построения ЭС

#### **2. Краткое содержание вопросов**

##### **2.1. Предметные области для Экспертных систем**

В начале восьмидесятых годов в исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название "экспертные системы" (ЭС). Цель исследований по ЭС состоит в разработке программ, которые при решении задач, трудных для эксперта-человека, получают результаты, не уступающие по качеству и эффективности решениям, получаемым экспертом. Исследователи в области ЭС для названия своей дисциплины часто используют также термин "инженерия знаний", введенный Е.Фейгенбаумом как "привнесение принципов и инструментария исследований из области искусственного интеллекта в решение трудных прикладных проблем, требующих знаний экспертов".

Программные средства (ПС), базирующиеся на технологии экспертных систем, или инженерии знаний (в дальнейшем будем использовать их как синонимы), получили значительное распространение в мире. Важность экспертных систем состоит в следующем:

технология экспертных систем существенно расширяет круг практически значимых задач, решаемых на компьютерах, решение которых приносит значительный экономический эффект;

технология ЭС является важнейшим средством в решении глобальных проблем традиционного программирования: длительность и, следовательно, высокая стоимость разработки сложных приложений;

высокая стоимость сопровождения сложных систем, которая часто в несколько раз превосходит стоимость их разработки; низкий уровень повторной используемости программ и т.п.;

объединение технологии ЭС с технологией традиционного программирования добавляет новые качества к программным продуктам за счет: обеспечения динамичной модификации приложений пользователем, а не программистом; большей "прозрачности" приложения (например, знания хранятся на ограниченном ЕЯ, что не требует комментариев к знаниям, упрощает обучение и сопровождение); лучшей графики; интерфейса и взаимодействия.

По мнению ведущих специалистов, в недалекой перспективе ЭС найдут следующее применение:

ЭС будут играть ведущую роль во всех фазах проектирования, разработки, производства, распределения, продажи, поддержки и оказания услуг;

технология ЭС, получившая коммерческое распространение, обеспечит революционный прорыв в интеграции приложений из готовых интеллектуально-взаимодействующих модулей.

ЭС предназначены для так называемых неформализованных задач, т.е. ЭС не отвергают и не заменяют традиционного подхода к разработке программ, ориентированного на решение формализованных задач.

Неформализованные задачи обычно обладают следующими особенностями:

ошибочностью, неоднозначностью, неполнотой и противоречивостью исходных данных;

ошибочностью, неоднозначностью, неполнотой и противоречивостью знаний о проблемной области и решаемой задаче;

большой размерностью пространства решения, т.е. перебор при поиске решения весьма велик;

динамически изменяющимися данными и знаниями.

Следует подчеркнуть, что неформализованные задачи представляют большой и очень важный класс задач. Многие специалисты считают, что эти задачи являются наиболее массовым классом задач, решаемых ЭВМ.

Экспертные системы и системы искусственного интеллекта отличаются от систем обработки данных тем, что в них в основном используются символьный (а не числовой) способ представления, символьный вывод и эвристический поиск решения (а не исполнение известного алгоритма).

Экспертные системы применяются для решения только трудных практических (не игрушечных) задач. По качеству и эффективности решения экспертные системы не уступают решениям эксперта-человека. Решения экспертных систем обладают "прозрачностью", т.е. могут быть объяснены пользователю на качественном уровне. Это качество экспертных систем обеспечивается их способностью рассуждать о своих знаниях и умозаключениях. Экспертные системы способны пополнять свои знания в ходе взаимодействия с экспертом. Необходимо отметить, что в настоящее время технология экспертных систем используется для решения различных типов задач (интерпретация, предсказание, диагностика, планирование, конструирование, контроль, отладка, инструктаж, управление) в самых разнообразных проблемных областях, таких, как финансы, нефтяная и газовая промышленность, энергетика, транспорт, фармацевтическое производство, космос, металлургия, горное дело, химия, образование, целлюлозно-бумажная промышленность, телекоммуникации и связь и др.

Коммерческие успехи к фирмам-разработчикам систем искусственного интеллекта (СИИ) пришли не сразу. На протяжении 1960 - 1985 гг. успехи ИИ касались в основном

исследовательских разработок, которые демонстрировали пригодность СИИ для практического использования. Начиная примерно с 1985 г. (в массовом масштабе с 1988 - 1990 гг.), в первую очередь ЭС, а в последние годы системы, воспринимающие естественный язык (ЕЯ-системы), и нейронные сети (НС) стали активно использоваться в коммерческих приложениях.

Следует обратить внимание на то, что некоторые специалисты (как правило, специалисты в программировании, а не в ИИ) продолжают утверждать, что ЭС и СИИ не оправдали возлагавшихся на них ожиданий и умерли. Причины таких заблуждений состоят в том, что эти авторы рассматривали ЭС как альтернативу традиционному программированию, т.е. они исходили из того, что ЭС в одиночестве (в изоляции от других программных средств) полностью решают задачи, стоящие перед заказчиком. Надо отметить, что на заре появления ЭС специфика используемых в них языков, технологии разработки приложений и используемого оборудования (например, Lisp-машины) давала основания предполагать, что интеграция ЭС с традиционными, программными системами является сложной и, возможно, невыполнимой задачей при ограничениях, накладываемых реальными приложениями. Однако в настоящее время коммерческие инструментальные средства (ИС) для создания ЭС разрабатываются в полном соответствии с современными технологическими тенденциями традиционного программирования, что снимает проблемы, возникающие при создании интегрированных приложений.

Причины, приведшие СИИ к коммерческому успеху, следующие.

**Интегрированность.** Разработаны инструментальные средства искусственного интеллекта (ИС ИИ), легко интегрирующиеся с другими информационными технологиями и средствами (с CASE, СУБД, контроллерами, концентраторами данных и т.п.).

**Открытость и переносимость.** ИС ИИ разрабатываются с соблюдением стандартов, обеспечивающих открытость и переносимость [14].

**Использование языков традиционного программирования и рабочих станций.** Переход от ИС ИИ, реализованных на языках ИИ (Lisp, Prolog и т.п.), к ИС ИИ, реализованным на языках традиционного программирования (C, C++ и т.п.), упростил обеспечение интегрированности, снизил требования приложений ИИ к быстродействию ЭВМ и объемам оперативной памяти. Использование рабочих станций (вместо ПК) резко увеличило круг приложений, которые могут быть выполнены на ЭВМ с использованием ИС ИИ.

**Архитектура клиент-сервер.** Разработаны ИС ИИ, поддерживающие распределенные вычисления по архитектуре клиент-сервер, что позволило: снизить стоимость оборудования, используемого в приложениях, децентрализовать приложения, повысить надежность и общую производительность (так как сокращается количество информации, пересылаемой между ЭВМ, и каждый модуль приложения выполняется на адекватном ему оборудовании).

**Проблемно/предметно-ориентированные ИС ИИ.** Переход от разработок ИС ИИ общего назначения (хотя они не утратили свое значение как средство для создания ориентированных ИС) к проблемно/предметно-ориентированным ИС ИИ [9] обеспечивает: сокращение сроков разработки приложений; увеличение эффективности использования ИС; упрощение и ускорение работы эксперта; повторную используемость информационного и программного обеспечения (объекты, классы, правила, процедуры).

## **2.2. Обобщенная структура ЭС. Основные понятия и определения**

Типичная статическая ЭС состоит из следующих основных компонентов (рис. 1.):

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



- объяснительного компонента;
- диалогового компонента.

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

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

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

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

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



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

В разработке ЭС участвуют представители следующих специальностей:

эксперт в проблемной области, задачи которой будет решать ЭС;

инженер по знаниям - специалист по разработке ЭС (используемые им технологии, методы называют технологией (методами) инженерии знаний);

программист по разработке инструментальных средств (ИС), предназначенных для ускорения разработки ЭС.

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

**Эксперт** определяет знания (данные и правила), характеризующие проблемную область, обеспечивает полноту и правильность введенных в ЭС знаний.

**Инженер по знаниям** помогает эксперту выявить и структурировать знания, необходимые для работы ЭС; осуществляет выбор того ИС, которое наиболее подходит для данной проблемной области, и определяет способ представления знаний в этом ИС;

выделяет и программирует (традиционными средствами) стандартные функции (типичные для данной проблемной области), которые будут использоваться в правилах, вводимых экспертом.

**Программист** разрабатывает ИС (если ИС разрабатывается заново), содержащее в пределе все основные компоненты ЭС, и осуществляет его сопряжение с той средой, в которой оно будет использовано.

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

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

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

### 2.3. Классификация ЭС

История развития ИС для создания ЭС реального времени началась в 1985 г., когда фирма Lisp Machine Inc. выпустила систему Picon для символьных ЭВМ Symbolics. Успех этого ИС привел к тому, что группа ведущих разработчиков Picon в 1986 г. образовала частную фирму Gensym, которая, значительно развив идеи, заложенные в Picon, в 1988 г. вышла на рынок с ИС под названием G2, версия 1.0. В настоящее время функционирует версия 4.2 и готовится к выпуску версия 5.0.

Основное предназначение программных продуктов фирмы Gensym (США) - помочь предприятиям сохранять и использовать знания и опыт их наиболее талантливых и квалифицированных сотрудников в интеллектуальных системах реального времени, повышающих качество продукции, надежность и безопасность производства и снижающих производственные издержки. О том, как фирме Gensym удастся справиться с этой задачей, говорит хотя бы то, что сегодня ей принадлежат 50% мирового рынка экспертных систем, используемых в системах управления.

С отставанием от Gensym на 2 - 3 года другие фирмы начали создавать свои ИС для ЭС РВ. С точки зрения независимых экспертов NASA, проводивших комплексное исследование характеристик и возможностей некоторых из перечисленных систем, в настоящее время наиболее продвинутой ИС, безусловно, остается G2 (Gensym, США); следующие места со значительным отставанием (реализовано менее 50% возможностей G2) занимают RTWorks - фирма Talarian (США), COMDALE/C (Comdale Techn. - Канада), COGSYS (SC - США), ILOG Rules (ILOG - Франция).

Классы задач, для которых предназначена G2 и подобные ей системы:

- мониторинг в реальном масштабе времени;
- системы управления верхнего уровня;
- системы обнаружения неисправностей;
- диагностика;
- составление расписаний;
- планирование;

- оптимизация;
- системы - советчики оператора;
- системы проектирования.

Инструментальные средства фирмы Gensym являются эволюционным шагом в развитии традиционных экспертных систем от статических предметных областей к динамическим. Немалую долю успеха фирме Gensym обеспечивают основные принципы, которых она придерживается в своих новых разработках:

- проблемно/предметная ориентация;
- следование стандартам;
- независимость от вычислительной платформы;
- совместимость снизу-вверх с предыдущими версиями;
- универсальные возможности, не зависящие от решаемой задачи;
- обеспечение технологической основы для прикладных систем;
- комфортная среда разработки;
- поиск новых путей развития технологии;
- распределенная архитектура клиент-сервер;
- высокая производительность.

Основным достоинством оболочки экспертных систем G2 для российских пользователей является возможность применять ее как интегрирующий компонент, позволяющий за счет открытости интерфейсов и поддержки широкого спектра вычислительных платформ легко объединить уже существующие, разрозненные средства автоматизации в единую комплексную систему управления, охватывающую все аспекты производственной деятельности - от формирования портфеля заказов до управления технологическим процессом и отгрузки готовой продукции. Это особенно важно для отечественных предприятий, парк технических и программных средств которых формировался по большей части бессистемно, под влиянием резких колебаний в экономике.

Кроме системы G2, как базового средства разработки, фирма Gensym предлагает комплекс проблемно/предметно-ориентированных расширений для быстрой реализации сложных динамических систем на основе специализированных графических языков, включающих параметризуемые операторные блоки для представления элементов технологического процесса и типовых задач обработки информации. Набор инструментальных сред фирмы Gensym, сгруппированный по проблемной ориентации, охватывает все стадии производственного процесса и выглядит следующим образом:

- интеллектуальное управление производством - G2, G2 Diagnostic Assistant (GDA), NeurOn-Line (NOL), Statistical Process Control (SPC), BatchDesign\_Kit;

- оперативное планирование - G2, G2 Scheduling Toolkit (GST), Dynamic Scheduling Packadge (DSP);

- разработка и моделирование производственных процессов - G2, ReThink, BatchDesign\_Kit;

- управление операциями и корпоративными сетями - G2, Fault Expert.

Несмотря на то, что первая версия системы G2 появилась не так давно - в 1988 г., ее даже в богатой Америке никто не назовет дешевой. G2 можно назвать бестселлером на рынке программных продуктов - на начало 1996 г. в мире было установлено более 5000 ее копий. Фирма Gensym обслуживает более 30 отраслей - от аэрокосмических исследований до производства пищевых продуктов. Список пользователей G2 выглядит как справочник Who-Is-Who в мировой промышленности. 25 самых крупных индустриальных мировых корпораций используют G2. На базе G2 написано более 500 действующих приложений.

## **2.4. Инструментальные средства построения ЭС**

Комплекс ЭКО включает три компонента.

Ядром комплекса является *интегрированная оболочка* экспертных систем ЭКО, которая обеспечивает быстрое создание эффективных приложений для решения задач анализа в статических проблемных средах типа 1 и 2.

При разработке средств представления знаний оболочки преследовались две основные цели: эффективное решение достаточно широкого и практически значимого класса задач средствами персональных компьютеров; гибкие возможности по описанию пользовательского интерфейса и проведению консультации в конкретных приложениях. При представлении знаний в оболочке используются специализированные (частные) - утверждения типа "атрибут - значение" и частные правила, что позволяет исключить ресурсоемкую операцию сопоставления по образцу и добиться эффективности разрабатываемых приложений. Выразительные возможности оболочки удалось существенно расширить за счет интегрированности, обеспечиваемой путем вызова внешних программ через сценарий консультации и стыковки с базами данных (ПИРС и dBase IV) и внешними программами. В оболочке ЭКО обеспечивается слабая структуризация БЗ за счет ее разделения на отдельные компоненты - для решения отдельных подзадач в проблемной среде - модели (понятию "модель" ЭКО соответствует понятие "модуль" базы знаний системы G2).

С точки зрения технологии разработки ЭС оболочка поддерживает подходы, основанные на поверхностных знаниях и структурировании процесса решения.

Оболочка функционирует в двух режимах: в режиме приобретения знаний и в режиме консультации (решения задач). В первом режиме разработчик ЭС средствами диалогового редактора вводит в БЗ описание конкретного приложения в терминах языка представления знаний оболочки. Это описание компилируется в сеть вывода с прямыми адресными ссылками на конкретные утверждения и правила. Во втором режиме оболочка решает конкретные задачи пользователя в диалоговом или пакетном режиме. При этом решения выводятся от целей к данным (обратное рассуждение).

Для расширения возможностей оболочки по работе с глубинными знаниями комплекс ЭКО может быть дополнен компонентом К-ЭКО (*конкретизатором знаний*), который позволяет описывать закономерности в проблемных средах в терминах общих (абстрактных) объектов и правил. К-ЭКО используется на этапе приобретения знаний вместо диалогового редактора оболочки для преобразования общих описаний в конкретные сети вывода, допускающие эффективный вывод решений средствами оболочки ЭКО. Таким образом, использование конкретизатора обеспечивает возможность работы с проблемными средами типа 2 (см. гл.3).

Третий компонент комплекса - *система ИЛИС*, позволяющая создавать ЭС в статических проблемных средах за счет индуктивного обобщения данных (примеров) и предназначенная для использования в тех приложениях, где отсутствие правил, отражающих закономерности в проблемной среде, возмещается обширным экспериментальным материалом. Система ИЛИС обеспечивает автоматическое формирование простейших конкретных правил и автономное решение задач на их основе; при этом используется жесткая схема диалога с пользователем. Поскольку при создании реальных приложений эксперты представляют, как правило, и знания о закономерностях в проблемной среде, и экспериментальный материал (для решения частных подзадач), возникает необходимость в использовании правил, сформированных системой ИЛИС, в рамках более сложных средств представления знаний. Комплекс ЭКО обеспечивает автоматический перевод таких правил в формат оболочки ЭКО. В результате удастся получить полное (адекватное) представление реальной проблемной среды, кроме того, задать гибкое описание организации взаимодействия ЭС с конечным пользователем.

### **Лекция № 3**

#### **Тема: «Теоретические аспекты инженерии знаний»**

## 1. Вопросы лекции:

1. Стратегии получения знаний
2. Теоретические аспекты структурирования знаний

## 2. Краткое содержание вопросов

### 2.1. Стратегии получения знаний

Информация, с которой имеют дело ЭВМ, разделяется на *процедурную* и *декларативную*. Процедурная информация овеществлена в *программах*, которые выполняются в процессе решения задач, декларативная информация - в *данных*, с которыми эти программы работают. Стандартной формой представления информации в ЭВМ является *машинное слово*, состоящее из определенного для данного типа ЭВМ числа двоичных разрядов - *битов*. Машинное слово для представления данных и машинное слово для представления команд, образующих программу, могут иметь одинаковое или разное число разрядов. В последнее время для представления данных и команд используются одинаковые по числу разрядов машинные слова. Однако в ряде случаев машинные слова разбиваются на группы по восемь двоичных разрядов, которые называются *байтами*.

Одинаковое число разрядов в машинных словах для команд и данных позволяет рассматривать их в ЭВМ в качестве одинаковых информационных единиц и выполнять операции над командами, как над данными. Содержимое памяти образует *информационную базу*.

В большинстве существующих ЭВМ возможно извлечение информации из любого подмножества разрядов машинного слова вплоть до одного бита. Во многих ЭВМ можно соединять два или более машинного слова в слово с большей длиной. Однако машинное слово является основной характеристикой информационной базы, т.к. его длина такова, что каждое машинное слово хранится в одной стандартной ячейке памяти, снабженной индивидуальным именем - адресом ячейки. По этому имени происходит извлечение информационных единиц из памяти ЭВМ и записи их в нее.

Параллельно с развитием структуры ЭВМ происходило развитие информационных структур для представления данных. Появились способы описания данных в виде векторов и матриц, возникли списочные структуры, иерархические структуры. В настоящее время в языках программирования высокого уровня используются *абстрактные типы данных*, структура которых задается программистом. Появление *баз данных* (БД) знаменовало собой еще один шаг на пути организации работы с декларативной информацией. В базах данных могут одновременно храниться большие объемы информации, а специальные средства, образующие *систему управления базами данных* (СУБД), позволяют эффективно манипулировать с данными, при необходимости извлекать их из базы данных и записывать их в нужном порядке в базу.

По мере развития исследований в области ИС возникла концепция знаний, которые объединили в себе многие черты процедурной и декларативной информации.

В ЭВМ *знания* так же, как и *данные*, отображаются в знаковой форме - в виде формул, текста, файлов, информационных массивов и т.п. Поэтому можно сказать, что знания - это особым образом организованные данные. Но это было бы слишком узкое понимание. А между тем, в системах ИИ знания являются основным объектом формирования, обработки и исследования. *База знаний*, наравне с базой данных, - необходимая составляющая программного комплекса ИИ. Машины, реализующие алгоритмы ИИ, называются *машинами, основанными на знаниях*, а подраздел теории ИИ, связанный с построением экспертных систем, - *инженерией знаний*.

## 2.2. Теоретические аспекты структурирования знаний

Система ИИ в определенном смысле моделирует интеллектуальную деятельность человека и, в частности, - логику его рассуждений. В грубо упрощенной форме наши логические построения при этом сводятся к следующей схеме: из одной или нескольких посылок (которые считаются истинными) следует сделать "логически верное" заключение (вывод, следствие). Очевидно, для этого необходимо, чтобы и посылки, и заключение были представлены на понятном языке, адекватно отражающем предметную область, в которой проводится вывод. В обычной жизни это наш естественный язык общения, в математике, например, это язык определенных формул и т.п. Наличие же языка предполагает, во - первых, наличие алфавита (словаря), отображающего в символьной форме весь набор базовых понятий (элементов), с которыми придется иметь дело и, во - вторых, набор синтаксических правил, на основе которых, пользуясь алфавитом, можно построить определенные выражения.

Логические выражения, построенные в данном языке, могут быть истинными или ложными. Некоторые из этих выражений, являющиеся всегда истинными. Объявляются *аксиомами* (или *постулатами*). Они составляют ту базовую систему посылок, исходя из которой и пользуясь определенными правилами вывода, можно получить заключения в виде новых выражений, также являющихся истинными.

Если перечисленные условия выполняются, то говорят, что система удовлетворяет требованиям *формальной теории*. Ее так и называют *формальной системой* (ФС). Система, построенная на основе формальной теории, называется также *аксиоматической системой*.

Формальная теория должна, таким образом, удовлетворять следующему определению: всякая формальная теория  $F = (A, V, W, R)$ , определяющая некоторую аксиоматическую систему, характеризуется:

наличием алфавита (словаря),  $A$ ,  
множеством синтаксических правил,  $V$ ,  
множеством аксиом, лежащих в основе теории,  $W$ ,  
множеством правил вывода,  $R$ .

Исчисление высказываний (ИВ) и исчисление предикатов (ИП) являются классическими примерами аксиоматических систем. Эти ФС хорошо исследованы и имеют прекрасно разработанные модели логического вывода - главной метапроцедуры в интеллектуальных системах. Поэтому все, что может и гарантирует каждая из этих систем, гарантируется и для прикладных ФС как моделей конкретных предметных областей. В частности, это гарантии непротиворечивости вывода, алгоритмической разрешимости (для исчисления высказываний) и полуразрешимости (для исчислений предикатов первого порядка).

ФС имеют и недостатки, которые заставляют искать иные формы представления. Главный недостаток - это "закрытость" ФС, их негибкость. Модификация и расширение здесь всегда связаны с перестройкой всей ФС, что для практических систем сложно и трудоемко. В них очень сложно учитывать происходящие изменения. Поэтому ФС как модели представления знаний используются в тех предметных областях, которые хорошо локализуются и мало зависят от внешних факторов.

### Лекция № 4

#### Тема: «Технологии инженерии знаний»

#### 1. Вопросы лекции:

##### 1. Классификация методов практического извлечения знаний

2. Коммуникативные методы
3. Текстологические методы
4. Простейшие методы структурирования знаний

## **2. Краткое содержание вопросов**

### **2.1. Классификация методов практического извлечения знаний**

В настоящее время в исследованиях по искусственному интеллекту (ИИ) выделились шесть направлений:

1. Представление знаний.
2. Манипулирование знаниями.
3. Общение.
4. Восприятие.
5. Обучение.
6. Поведение.

В рамках направления "Представление знаний" решаются задачи, связанные с формализацией и представлением знаний в памяти интеллектуальной системы (ИС). Для этого разрабатываются специальные модели представления знаний и языки для описания знаний, выделяются различные типы знаний. Изучаются источники, из которых ИС может черпать знания, и создаются процедуры и приемы, с помощью которых возможно приобретение знаний для ИС. Проблема представления знаний для ИС чрезвычайно актуальна, т.к. ИС - это система, функционирование которой опирается на знания о проблемной области, которые хранятся в ее памяти.

### **2.2. Коммуникативные методы**

Информация, с которой имеют дело ЭВМ, разделяется на *процедурную* и *декларативную*. Процедурная информация овеществлена в *программах*, которые выполняются в процессе решения задач, декларативная информация - в *данных*, с которыми эти программы работают. Стандартной формой представления информации в ЭВМ является *машинное слово*, состоящее из определенного для данного типа ЭВМ числа двоичных разрядов - *битов*. Машинное слово для представления данных и машинное слово для представления команд, образующих программу, могут иметь одинаковое или разное число разрядов. В последнее время для представления данных и команд используются одинаковые по числу разрядов машинные слова. Однако в ряде случаев машинные слова разбиваются на группы по восемь двоичных разрядов, которые называются *байтами*.

Одинаковое число разрядов в машинных словах для команд и данных позволяет рассматривать их в ЭВМ в качестве одинаковых информационных единиц и выполнять операции над командами, как над данными. Содержимое памяти образует *информационную базу*.

В большинстве существующих ЭВМ возможно извлечение информации из любого подмножества разрядов машинного слова вплоть до одного бита. Во многих ЭВМ можно соединять два или более машинного слова в слово с большей длиной. Однако машинное слово является основной характеристикой информационной базы, т.к. его длина такова, что каждое машинное слово хранится в одной стандартной ячейке памяти, снабженной индивидуальным именем - адресом ячейки. По этому имени происходит извлечение информационных единиц из памяти ЭВМ и записи их в нее.

Параллельно с развитием структуры ЭВМ происходило развитие информационных структур для представления данных. Появились способы описания данных в виде векторов и матриц, возникли списочные структуры, иерархические структуры. В настоящее время в языках программирования высокого уровня

используются *абстрактные типы данных*, структура которых задается программистом. Появление *баз данных* (БД) знаменовало собой еще один шаг на пути организации работы с декларативной информацией. В базах данных могут одновременно храниться большие объемы информации, а специальные средства, образующие *систему управления базами данных* (СУБД), позволяют эффективно манипулировать с данными, при необходимости извлекать их из базы данных и записывать их в нужном порядке в базу.

По мере развития исследований в области ИС возникла концепция знаний, которые объединили в себе многие черты процедурной и декларативной информации.

В ЭВМ *знания* так же, как и *данные*, отображаются в знаковой форме - в виде формул, текста, файлов, информационных массивов и т.п. Поэтому можно сказать, что знания - это особым образом организованные данные. Но это было бы слишком узкое понимание. А между тем, в системах ИИ знания являются основным объектом формирования, обработки и исследования. *База знаний*, наравне с базой данных, - необходимая составляющая программного комплекса ИИ. Машины, реализующие алгоритмы ИИ, называются *машинами, основанными на знаниях*, а подраздел теории ИИ, связанный с построением экспертных систем, - *инженерией знаний*.

### 2.3.Текстологические методы

Особенности знаний:

1. *Внутренняя интерпретируемость*. Каждая информационная единица должна иметь уникальное имя, по которому ИС находит ее, а также отвечает на запросы, в которых это имя упомянуто. Когда данные, хранящиеся в памяти, были лишены имен, то отсутствовала возможность их идентификации системой. Данные могла идентифицировать лишь программа, извлекающая их из памяти по указанию программиста, написавшего программу.

2. *Структурированность*. Информационные единицы должны обладать гибкой структурой. Для них должен выполняться "принцип матрешки", т.е. рекурсивная вложенность одних информационных единиц в другие. Каждая информационная единица может быть включена в состав любой другой, и из каждой информационной единицы можно выделить некоторые составляющие ее информационные единицы. Другими словами, должна существовать возможность произвольного установления между отдельными информационными единицами отношений типа "часть - целое", "род - вид" или "элемент - класс".

3. *Связность*. В информационной базе между информационными единицами должна быть предусмотрена возможность установления связей различного типа. Прежде всего эти связи могут характеризовать отношения между информационными единицами. Семантика отношений может носить декларативный или процедурный характер. Например, две или более информационные единицы могут быть связаны отношением "одновременно", две информационные единицы - отношением "причина - следствие" или отношением "быть рядом". Приведенные отношения характеризуют декларативные знания. Если между двумя информационными единицами установлено отношение "аргумент - функция", то оно характеризует процедурное знание, связанное с вычислением определенных функций. Далее будем различать *отношения структуризации*, *функциональные отношения*, *каузальные отношения* и *семантические отношения*. С помощью первых задаются иерархии информационных единиц, вторые несут процедурную информацию, позволяющую находить (вычислять) одни информационные единицы через другие, третьи задают причинно - следственные связи, четвертые соответствуют всем остальным отношениям.

Между информационными единицами могут устанавливаться и иные связи, например, определяющие порядок выбора информационных единиц из памяти или



указывающие на то, что две информационные единицы несовместимы друг с другом в одном описании.

Перечисленные три особенности знаний позволяют ввести общую модель представления знаний, которую можно назвать *семантической сетью*, представляющей собой иерархическую сеть, в вершинах которой находятся информационные единицы. Эти единицы снабжены индивидуальными именами. Дуги семантической сети соответствуют различным связям между информационными единицами. При этом иерархические связи определяются отношениями структуризации, а неиерархические связи - отношениями иных типов.

4. *Семантическая метрика*. На множестве информационных единиц в некоторых случаях полезно задавать отношение, характеризующее ситуационную близость информационных единиц, т.е. силу ассоциативной связи между информационными единицами. Его можно было бы назвать *отношением релевантности* для информационных единиц. Такое отношение дает возможность выделять в информационной базе некоторые типовые ситуации (например, "покупка", "регулирование движения на перекрестке"). Отношение релевантности при работе с информационными единицами позволяет находить знания, близкие к уже найденным.

5. *Активность*. С момента появления ЭВМ и разделения используемых в ней информационных единиц на данные и команды создалась ситуация, при которой данные пассивны, а команды активны. Все процессы, протекающие в ЭВМ, инициируются командами, а данные используются этими командами лишь в случае необходимости. Для ИС эта ситуация не приемлема. Как и у человека, в ИС актуализации тех или иных действий способствуют знания, имеющиеся в системе. Таким образом, выполнение программ в ИС должно инициироваться текущим состоянием информационной базы. Появление в базе фактов или описаний событий, установление связей может стать источником активности системы.

Перечисленные пять особенностей информационных единиц определяют ту грань, за которой данные превращаются в знания, а базы данных перерастают в *базы знаний* (БЗ). Совокупность средств, обеспечивающих работу с знаниями, образует *систему управления базой знаний* (СУБЗ). В настоящее время не существует баз знаний, в которых в полной мере были бы реализованы внутренняя интерпретируемость, структуризация, связность, введена семантическая мера и обеспечена активность знаний.

## 2.4. Простейшие методы структурирования знаний

Существуют два типа методов представления знаний (ПЗ):

1. Формальные модели ПЗ;
2. Неформальные (семантические, реляционные) модели ПЗ.

Очевидно, все методы представления знаний, которые рассмотрены выше, включая продукции (это система правил, на которых основана продукционная модель представления знаний), относятся к неформальным моделям. В отличие от формальных моделей, в основе которых лежит строгая математическая теория, неформальные модели такой теории не придерживаются. Каждая неформальная модель годится только для конкретной предметной области и поэтому не обладает универсальностью, которая присуща моделям формальным. Логический вывод - основная операция в СИИ - в формальных системах строг и корректен, поскольку подчинен жестким аксиоматическим правилам. Вывод в неформальных системах во многом определяется самим исследователем, который и отвечает за его корректность.

Каждому из методов ПЗ соответствует свой способ описания знаний.

1. *Логические модели*. В основе моделей такого типа лежит *формальная система*, задаваемая четверкой вида:  $M = \langle T, P, A, B \rangle$ . Множество  $T$  есть *множество базовых элементов* различной природы, например слов из некоторого ограниченного словаря,

деталей детского конструктора, входящих в состав некоторого набора и т.п. Важно, что для множества  $T$  существует некоторый способ определения принадлежности или непринадлежности произвольного элемента к этому множеству. Процедура такой проверки может быть любой, но за конечное число шагов она должна давать положительный или отрицательный ответ на вопрос, является ли  $x$  элементом множества  $T$ . Обозначим эту процедуру  $\Pi(T)$ .

Множество  $P$  есть множество *синтаксических правил*. С их помощью из элементов  $T$  образуют *синтаксически правильные совокупности*. Например, из слов ограниченного словаря строятся синтаксически правильные фразы, из деталей детского конструктора с помощью гаек и болтов собираются новые конструкции. Декларируется существование процедуры  $\Pi(P)$ , с помощью которой за конечное число шагов можно получить ответ на вопрос, является ли совокупность  $X$  синтаксически правильной.

В множестве синтаксически правильных совокупностей выделяется некоторое подмножество  $A$ . Элементы  $A$  называются *аксиомами*. Как и для других составляющих формальной системы, должна существовать процедура  $\Pi(A)$ , с помощью которой для любой синтаксически правильной совокупности можно получить ответ на вопрос о принадлежности ее к множеству  $A$ .

Множество  $B$  есть множество *правил вывода*. Применяя их к элементам  $A$ , можно получать новые синтаксически правильные совокупности, к которым снова можно применять правила из  $B$ . Так формируется *множество выводимых* в данной формальной системе *совокупностей*. Если имеется процедура  $\Pi(B)$ , с помощью которой можно определить для любой синтаксически правильной совокупности, является ли она выводимой, то соответствующая формальная система называется *разрешимой*. Это показывает, что именно правило вывода является наиболее сложной составляющей формальной системы.

Для знаний, входящих в базу знаний, можно считать, что множество  $A$  образуют все информационные единицы, которые введены в базу знаний извне, а с помощью правил вывода из них выводятся новые *производные знания*. Другими словами формальная система представляет собой генератор порождения новых знаний, образующих множество *выводимых* в данной системе *знаний*. Это свойство логических моделей делает их притягательными для использования в базах знаний. Оно позволяет хранить в базе лишь те знания, которые образуют множество  $A$ , а все остальные знания получать из них по правилам вывода.

2. *Сетевые модели*. В основе моделей этого типа лежит конструкция, названная ранее семантической сетью. Сетевые модели формально можно задать в виде  $H = \langle I, C_1, C_2, \dots, C_n, \Gamma \rangle$ . Здесь  $I$  есть множество информационных единиц;  $C_1, C_2, \dots, C_n$  - множество типов связей между информационными единицами. Отображение  $\Gamma$  задает между информационными единицами, входящими в  $I$ , связи из заданного набора типов связей.

В зависимости от типов связей, используемых в модели, различают *классифицирующие сети*, *функциональные сети* и *сценарии*. В классифицирующих сетях используются отношения структуризации. Такие сети позволяют в базах знаний вводить разные иерархические отношения между информационными единицами. Функциональные сети характеризуются наличием функциональных отношений. Их часто называют *вычислительными моделями*, т.к. они позволяют описывать процедуры "вычислений" одних информационных единиц через другие. В сценариях используются каузальные отношения, а также отношения типов "средство - результат", "орудие - действие" и т.п. Если в сетевой модели допускаются связи различного типа, то ее обычно называют семантической сетью.

3. *Продукционные модели*. В моделях этого типа используются некоторые элементы логических и сетевых моделей. Из логических моделей заимствована идея правил вывода, которые здесь называются *продукциями*, а из сетевых моделей - описание знаний в виде семантической сети. В результате применения правил вывода к фрагментам

сетевого описания происходит трансформация семантической сети за счет смены ее фрагментов, наращивания сети и исключения из нее ненужных фрагментов. Таким образом, в продукционных моделях процедурная информация явно выделена и описывается иными средствами, чем декларативная информация. Вместо логического вывода, характерного для логических моделей, в продукционных моделях появляется *вывод на знаниях*.

4. *Фреймовые модели*. В отличие от моделей других типов во фреймовых моделях фиксируется жесткая структура информационных единиц, которая называется *протофреймом*. В общем виде она выглядит следующим образом:

(Имя фрейма:

Имя слота 1(значение слота 1)

Имя слота 2(значение слота 2)

.....

Имя слота К (значение слота К)).

Значением *слота* может быть практически что угодно (числа или математические соотношения, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма или других фреймов). В качестве значения слота может выступать набор слотов более низкого уровня, что позволяет во фреймовых представлениях реализовать "принцип матрешки".

## Лекция № 5

### Тема: «Перспективы развития СИИ»

#### 1. Вопросы лекции:

1. Проблемы ИИ связаны с ресурсами
2. Интеллектуальные интернет – технологии

#### 2.Краткое содержание вопросов

##### 2.1. Проблемы ИИ связаны с ресурсами

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 [1], 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.

Коммерческий рынок продуктов искусственного интеллекта в мире в 1993 году оценивался примерно в 0,9 млрд. долларов; из них 600 млн. приходится на долю США [2]. 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 1993 году рынок между этими направлениями распределился так [2]: 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 1993 году в общем объеме рынка США доля приложений составила примерно две, а доля инструментария - примерно одну треть [2].

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 MIT (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. Например, выражение  $a+b*c$  выглядит так:

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 хорошенько уже подзабытый язык LISP (List Processing), 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.

## **2.2. Интеллектуальные интернет – технологии**

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 (хотя и это достойно тщательного анализа), но, в первую очередь, тем, что только с помощью подобных средств создаются стратегически значимые приложения в таких областях, как управление непрерывными производственными процессами 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.

Классы задач, решаемых экспертными системами реального времени, таковы: мониторинг в реальном масштабе времени, системы управления верхнего уровня, системы обнаружения неисправностей, диагностика, составление расписаний, планирование, оптимизация, системы-советчики оператора, системы проектирования.

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

1. Представлять изменяющиеся во времени данные, поступающие от внешних источников, обеспечивать хранение и анализ изменяющихся данных.

2. Выполнять временные рассуждения о нескольких различных асинхронных процессах одновременно (т.е. планировать в соответствии с приоритетами обработку поступивших в систему процессов).

3. Обеспечивать механизм рассуждения при ограниченных ресурсах (время, память). Реализация этого механизма предъявляет требования к высокой скорости работы системы, способности одновременно решать несколько задач (т.е. операционные системы UNIX, VMS, Windows NT, но не MS-DOS).

4. Обеспечивать "предсказуемость" поведения системы, т.е. гарантию того, что каждая задача будет запущена и завершена в строгом соответствии с временными ограничениями. Например, данное требование не допускает использования в экспертной системе реального времени механизма "сборки мусора", свойственного языку Lisp.

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

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

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

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

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

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

Подчеркнем, что кроме этих десяти требований средства создания экспертных систем реального времени должны удовлетворять и перечисленным выше общим требованиям.

### **3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ**

#### **3.1 Практическое занятие 1 (ПЗ-1) - Введение в интеллектуальные системы**

##### **3.1.1 Задание для работы:**

1. Основные понятия и определения
2. Область применения
3. Краткий исторический обзор развития работ в области ИИ
4. Функциональная структура использования СИИ

##### **3.1.2 Краткое описание проводимого занятия:**

1. С помощью устного опроса и (или) тестирования оценить уровень усвоения студентами изученного материала.

##### **3.1.3 Результаты и выводы:**

Усвоение студентами знаний по теме практического занятия.

#### **1.2 Практическое занятие 2 (ПЗ-2) - Разработка систем, основанных на знаниях**

##### **3.2.1 Задание для работы:**

1. Предметные области для Экспертных систем
2. Обобщенная структура ЭС. Основные понятия и определения
3. Классификация ЭС
4. Инструментальные средства построения ЭС

##### **3.2.2 Краткое описание проводимого занятия:**

1. С помощью устного опроса и (или) тестирования оценить уровень усвоения студентами изученного материала.

##### **3.2.3 Результаты и выводы:**

Усвоение студентами знаний по теме практического занятия.

#### **3.3 Практическое занятие 3 (ПЗ-3) - Теоретические аспекты инженерии знаний**

### **3.3.1 Задание для работы:**

1. Стратегии получения знаний
2. Теоретические аспекты структурирования знаний

### **3.3.2 Краткое описание проводимого занятия:**

1. С помощью устного опроса и (или) тестирования оценить уровень усвоения студентами изученного материала.

### **3.3.3 Результаты и выводы:**

Усвоение студентами знаний по теме практического занятия.

## **3.4 Практическое занятие 4 (ПЗ-4) - Технологии инженерии знаний**

### **3.4.1 Задание для работы:**

1. Классификация методов практического извлечения знаний
2. Коммуникативные методы
3. Текстологические методы
4. Простейшие методы структурирования знаний

### **3.4.2 Краткое описание проводимого занятия:**

1. С помощью устного опроса и (или) тестирования оценить уровень усвоения студентами изученного материала.

### **3.4.3 Результаты и выводы:**

Усвоение студентами знаний по теме практического занятия.

## **3.5 Практическое занятие 5 (ПЗ-5) - Перспективы развития СИИ**

### **3.5.1 Задание для работы:**

1. Проблемы ИИ связаны с ресурсами
2. Интеллектуальные интернет – технологии

### **3.5.2 Краткое описание проводимого занятия:**



1. С помощью устного опроса и (или) тестирования оценить уровень усвоения студентами изученного материала.

### **3.5.3 Результаты и выводы:**

Усвоение студентами знаний по теме практического занятия.