

Московский Энергетический Институт  
Институт Автоматики и Вычислительной Техники

# Отчет по курсовому проекту

---

## Архитектурные подходы искусственного интеллекта. Системы, основанные на правилах.

---

Выполнил студент: КОРОЛЕВ Юрий, группа А-13-06

КУРС: ПРОЕКТИРОВАНИЕ КРУПНЫХ РАСПРЕДЕЛЕННЫХ ПРОГРАММНЫХ СИСТЕМ

ПРЕПОДАВАТЕЛЬ: КУРИЛЕНКО И. Е., к. т. н.

Москва, 2010.

## Оглавление

Список сокращений, принятых в тексте .....	2
Введение .....	3
Различные подходы к построению систем искусственного интеллекта .....	7
Определение искусственного интеллекта .....	7
Задачи искусственного интеллекта.....	10
Построение систем искусственного интеллекта.....	15
Состояние и тенденции развития искусственного интеллекта .....	19
Успехи систем искусственного интеллекта и их причины .....	22
Архитектура систем, основанных на правилах, на примере экспертной системы .....	25
Общие понятия экспертных систем .....	28
Элементы экспертной системы.....	30
Производственные системы.....	39
Производственные системы Поста.....	40
Rete-алгоритм .....	42
Современные разработки в области искусственного интеллекта .....	45
Применение технологии многоагентных систем для интеллектуальной поддержки принятия решения.....	47
Технология многоагентных систем.....	48
Свойства интеллектуального агента .....	52
Классы архитектур .....	53
Концепции использования технологии многоагентных интеллектуальных систем .....	54
Технология программной реализации интеллектуальных многоагентных систем .....	55
Направления в разработке многоагентных систем.....	57
Заключение .....	60
Список использованной литературы .....	62

## Список сокращений, принятых в тексте

- ИИ – искусственный интеллект
- ЭВМ – электронно-вычислительная машина
- НС – нейронные сети
- ЧЯ – черный ящик
- ЭС – экспертная система
- МАС – многоагентная интеллектуальная система
- БЗ – база знаний
- ИСППР – интеллектуальная система поддержки принятия решения

Теория вычислительных систем – дисциплина эмпирическая. Можно было бы назвать ее экспериментальной наукой, но, подобно астрономии, экономике и геологии, некоторые из ее оригинальных форм испытаний и наблюдений невозможно втиснуть в узкий стереотип экспериментального метода. Тем не менее, это эксперименты. Конструирование каждого нового компьютера – это эксперимент. Сам факт создания машины ставит вопрос перед природой; и мы получаем ответ на него, наблюдая за машиной в действии, анализируя ее всеми доступными способами. Каждая новая программа – это эксперимент. Она ставит вопрос природе, и ее поведение дает нам ключи к разгадке. Ни машины, ни программы не являются "черными ящиками", это творения наших рук, спроектированные как аппаратно, так и программно; мы можем снять крышку и заглянуть внутрь. Мы можем соотнести их структуру с поведением и извлечь множество уроков из одного-единственного эксперимента.

– Ньюэлл (A. Newell) и Саймон (H. A. Simon), лекция по случаю вручения премии Тьюринга, 1976.

## **Введение.**

Для многих людей наиболее удивительным аспектом работы в сфере искусственного интеллекта (ИИ) является степень, в которой ИИ, да и большая часть теории вычислительных систем, оказывается эмпирической дисциплиной. Этот аспект удивителен, поскольку большинство рассматривает эти области в терминах своего математического или инженерного образования. Пунктуальным математикам свойственно желание применить к конструированию интеллектуальных устройств привычные им логические рассуждения и анализ. С точки зрения инженеров, задача часто состоит лишь в создании систем, которые общество назвало бы "разумными". Но сложность интеллектуальных программ и неопределенность, присущая их взаимодействию с миром природы и человеческой деятельности, делают невозможным анализ с чисто математической или чисто инженерной точек зрения.

Более того, если мы пытаемся довести исследования искусственного интеллекта до уровня науки и сделать их неотъемлемой частью теории интеллектуальных систем (science of intelligent systems), то в процессе конструирования, использования и анализа артефактов должны применять смесь из аналитических и эмпирических методов. С этой точки зрения

каждая программа ИИ должна рассматриваться как эксперимент: он ставит вопрос перед природой, и ответ на него - это результат выполнения программы. Отклик природы на заложенные конструкторские и программные принципы формирует наше понимание формализма, закономерностей и самой сути мышления.

В отличие от многих традиционных наук, изучающих человеческое познание, разработчики разумных компьютерных систем могут исследовать внутренние механизмы своих "подопытных". Они могут останавливать выполнение программы, изучать ее внутреннее состояние и как угодно модифицировать ее структуру. Устройство компьютеров и компьютерных программ предопределяет их потенциальное поведение, возможность всестороннего исследования, и доступность для понимания. Сила компьютеров как инструментов для изучения интеллекта проистекает из этой двойственности. Соответствующим образом запрограммированные компьютеры способны достигнуть высокой степени сложности как в семантике, так и в поведении. Такие системы естественно охарактеризовать в терминах физиологии. Кроме того, можно исследовать их внутренние состояния, что в большинстве случаев не могут осуществить ученые, занимающиеся разумными формами жизни.

К счастью для работ в сфере ИИ, равно как и для становления теории интеллектуальных систем, современные физиологические методы, в особенности относящиеся к нейрофизиологии, пролили свет на многие аспекты человеческого мышления. Например, сегодня мы знаем, что функция человеческого интеллекта не цельна и однородна. Она, скорее, является модульной и распределенной. Достоинства этого подхода проявляются в работе органов чувств, например, сетчатки глаза, которая умеет фильтровать и предварительно обрабатывать визуальную информацию. Точно так же обучение нельзя назвать однородной, гомогенной способностью. Скорее, оно является функцией множества различных систем, каждая из которых адаптирована для специфических целей. Магнитно-

резонансное сканирование, позитронная эмиссионная томография и другие методы получения изображений нервной системы дают яркую и точную картину внутреннего устройства естественных интеллектуальных систем.

В системах искусственного интеллекта имеются подсистемы, которые мы уже сейчас можем реализовать, даже не зная о том, как они реализованы у человека. Причем можем это сделать не хуже, чем у прототипа, а зачастую и лучше. Например, искусственный глаз (а равно и блок первичной обработки видеoinформации, основанные на простейших фильтрах или др. сравнительно несложных устройствах) не устает, может видеть в любом диапазоне волн, легко заменяется на новый, видит при свете звезд.

За одно и тоже время, компьютер произведет гораздо больше арифметических операций и с большей точностью, чем человек. Устройства обработки звука позволяют улавливать девиацию голоса человека в 1-2 Герца. Данное изменение частоты происходит при повышенном возбуждении вегетативной нервной системы, которое в свою очередь часто обусловлено волнением человека. На данном принципе основаны современные детекторы лжи, которые позволяют обнаружить с высокой вероятностью даже записанные на пленку много лет назад ложные высказывания.

Современные системы управления электродвигателем позволяют с высокой точностью держать заданные координаты даже при ударном изменении нагрузки. А ведь это примерно тоже, что держать на длинной палке баскетбольный мяч, по которому то слева, то справа кидают теннисные мячи. Антиблокировочная система на автомобилях позволяет держать тормоза на грани заклинивания колеса, что дает наибольшее трение с дорогой, а это без АБС по силам только очень опытным водителям.

В принципе такие примеры, где техника оказывается ничуть не хуже человека, можно продолжать до бесконечности. Общий же смысл сказанного в том, что при конструировании искусственного интеллекта, мы не связаны одним набором элементарных составляющих, как природа. В каждом

конкретном случае желательно применять то, что даст самый большой эффект. В той области, где у человека господствуют рефлексы (чихание, быстрое напряжение быстро растягиваемой мышцы, переваривание пищи, регулировка температуры), мы вообще можем применить жесткие системы управления, с раз и навсегда заданным алгоритмом функционирования. При этом вполне можно ожидать увеличения точности и уменьшение времени обучения их до нуля. При этом ядро нашей системы искусственного интеллекта будет решать уже не настолько глобальные задачи.

Данный принцип разбиения задачи на подзадачи уже давно используется природой. К примеру, мы далеко не полностью используем все возможности наших мышц в области разнообразия движений. Мы не можем заставить наши глаза смотреть в разные стороны, не говоря уже о том, чтобы делать это на разном уровне (левый глаз - влево-вверх, правый - вправо-вниз). При ходьбе мы часто используем далеко не оптимальный набор движений и далеко не все сочетания вариантов напряжения мышц мы опробуем. На самом деле наша система управления построена по иерархическому принципу, когда задача распределяется между несколькими уровнями. Высший уровень нервной системы (связанный с большими полушариями мозга) ставит лишь общую задачу, скажем, переложить книгу на стол. Этот уровень вообще не контролирует действие отдельных двигательных единиц, направленных на решение поставленной задачи. Здесь уместна аналогия: командующий армией, ставя перед своими войсками некую общую задачу, отнюдь не предписывает каждому солдату и офицеру, что именно он должен делать в каждый момент операции.

В общем ситуация схожа с той, когда программист использует библиотеку подпрограмм. При этом ему не важно, какой алгоритм они используют, если программа работает нормально. А на написание своей библиотеки тратится драгоценное время. Кроме того, еще неизвестно, будет ли она работать так же хорошо.

## Различные подходы к построению систем искусственного интеллекта

### Определение искусственного интеллекта

Имеется, по крайней мере, две точки зрения на то, что следовало бы считать искусственным интеллектом. Первую можно назвать нейробионической. Ее сторонники ставят перед собой цель воспроизвести искусственным образом те процессы, которые протекают в мозгу человека, — это путь изучения естественного мозга, выявление способов его работы, создания технических средств для повторения биологических структур и протекающих в них процессов.

Вторая точка зрения, доминирующая в проблеме искусственного интеллекта, может быть названа информационной. Сторонники информационного исхода считают, что основной целью работ в искусственном интеллекте является не построение технического аналога биологической системы, а создание средств для решения задач, традиционно считающихся интеллектуальными.

Информационная точка зрения в свою очередь неоднородна. В ней можно выделить три направления.

Часть специалистов считает, что можно найти свой способ ее решения на ЭВМ, который даст либо результат, подобный человеческому, либо даже лучший. Специалисты этого направления неоднократно демонстрировали свое искусство по созданию программ такого рода. Достаточно назвать, например, программы для игры в шахматы, которые играют в эту игру лучше подавляющего большинства людей, проводящих время за шахматной доской. Но делают это программы совсем не так, как люди.

Другая часть специалистов считает, что искусственный интеллект должен имитировать не решение отдельных (пусть и весьма творческих) задач. Ибо естественный интеллект человека — это его способность при необходимости обучаться тому или иному виду творческой деятельности,

значит, и программы, создаваемые в искусственном интеллекте, должны быть ориентированы не на решение конкретных задач, а на создание для автоматического построения необходимых программ решения конкретных задач, когда в этом возникает необходимость. Именно эта группа исследователей сейчас определяет лицо искусственного интеллекта, составляя основную массу специалистов этого профиля.

Третья часть специалистов – это программисты, чьими руками делают программы для решения задач искусственного интеллекта. Они склонны рассматривать область своей деятельности как новый виток развития программирования. Они считают, что средства, разрабатываемые для написания программ решения интеллектуальных задач, в конце концов, есть средства, позволяющие по описанию задачи на профессиональном естественном языке построить нужную программу на основании тех стандартных программных модулей, которые хранятся в памяти машины. Все метасредства, которые предлагают те, кто рассматривает искусственный интеллект как способ разобраться на информационном уровне, какие функции реализует естественный интеллект, когда он решает задачу, программисты видят сквозь призму своей цели — создания интеллектуального программного обеспечения (по существу, комплекса средств, автоматизирующих деятельность самого программиста).

В те годы, когда возникали ЭВМ, мало кто предполагал, что они очень быстро вытеснят из вычислительной сферы все остальные вычислительные устройства. Дж. Фон-Нейман, с именем которого связана идея архитектуры классической ЭВМ, в те годы интересовался и другой организацией процесса вычислений, использующей аналоги нейроподобных структур; первые модели искусственных нейронов были предложены Мак-Калокком и Питсом.

По сути, эти элементы реализовали пороговую функцию, сигнал на выходе элемента возникал лишь тогда, когда взвешенная сумма разрешающих входных сигналов превышала взвешенную сумму



запрещающих входных сигналов более чем на величину, определяемую значением порога элемента. Варьируя значения весов и порога, можно было добиться нужного срабатывания формального нейрона. Объединенные в сети, такие нейроны представлялись весьма мощным способом реализации различных процедур. Одним из наиболее известных нейробионических устройств был персептрон, предложенный Ф. Розенблатом. Он породил целое семейство конструкций, в основе которых лежала идея первоначального устройства. Метод, который лежал в основе функционирования персептрона, похож на те приемы, которые используются в распознавании образов. Это научное направление весьма близко соприкасается с исследованиями по искусственному интеллекту. Строго говоря, нет никаких оснований не включать его в состав нового научного направления. Во всяком случае, нет особых возражений. Но, традиционно, возникшее гораздо ранее направление, связанное с распознаванием образов, существует отдельно. Хотя во многих пограничных вопросах эти две области научных исследований перекрываются (например, в методах формирования решающих правил при обучении на примерах и контрпримерах, как это происходит в персептронах, или в задачах анализа зрительных сцен).

Дальнейшие исследования в области нейробионических устройств шли по пути увеличения числа слоев из формальных нейронов, изменения и усложнения способа функционирования нейронов и построения решающего правила; параллельно развивалась теория персептронов. Но два обстоятельства затормозили эти работы. Очень быстро при решении практических задач распознавания стало понятно, что возможности устройств типа персептронов ограничены. Например, они не могли разложить изображение, являющееся комбинацией двух ранее персептрону известных, на составляющие. Это заставляло рассматривать подобную комбинацию как новое изображение. С другой стороны, Минский Н. и Пейперт С. доказали ряд теорем о персептронах, в которых обосновали их принципиальную ограниченность, а отсутствие новых идей

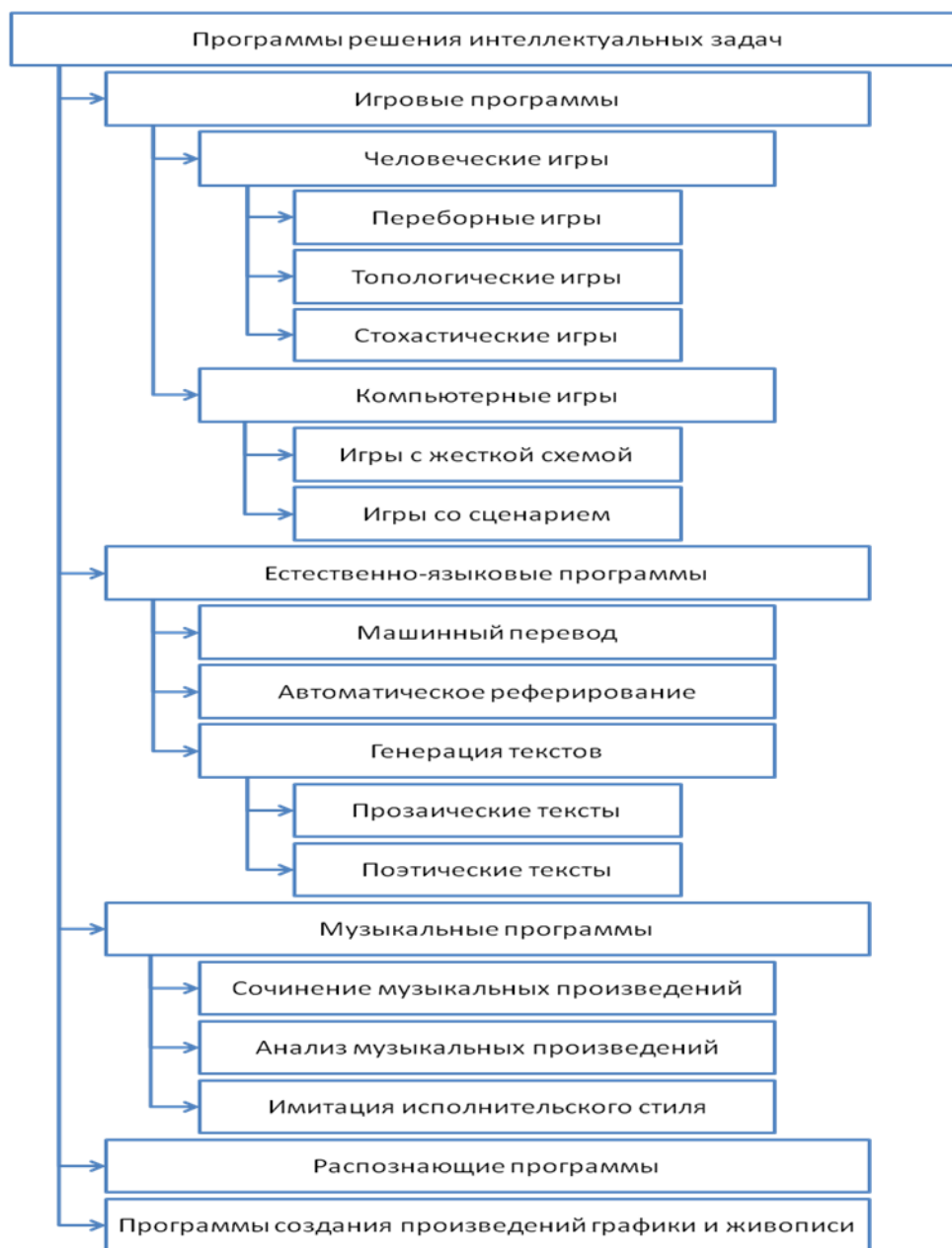
нейробионических устройств в течение десятка лет не давало повода для развития этих исследований. Но успехи микроэлектроники последних лет, сделавшие возможным создание большого числа нейроподобных элементов в малом объеме, вновь возродило надежды сторонников этого подхода. Появились нейрокомпьютеры, в которых процесс решения задачи развертывается на сети искусственных нейронов. Этот процесс может включать в себя множество параллельно и асинхронно протекающих подпроцессов, что сулит высокую эффективность решения задач на нейрокомпьютерах. Проблема состоит только в том, что пока неизвестны регулярные приемы программирования решения задач для ЭВМ такой архитектуры.

### **Задачи искусственного интеллекта**

Программы для решения интеллектуальных задач могут быть разделены на несколько групп, которые определяются типом задач, решаемых этими программами (рис. 1). Первую группу составляют игровые программы, они, в свою очередь, делятся на две подгруппы: человеческие игры и компьютерные игры. Особенностью всех программ для имитации человеческих игр является большая роль поисковых процедур — поиск лучшего или локально лучшего хода требует в сложных играх типа шахмат просмотра большого числа вариантов. Недаром шахматные программы являются специальным тестом для проверки эффективности поисковых процедур.

Интересно отметить, что именно поисковые процедуры казались на первом этапе развития работ по интеллектуальным программам той метапроцедурой, с помощью которой можно будет решать все интеллектуальные задачи. Первая программа, которая обобщила эту идею, называлась "Общий решатель задач". В этой программе, созданной А. Ньюэллом, Дж. Шоу и Г. Саймоном, поиск с локальными критериями успеха

был основной процедурой. Решение всех задач, по мысли авторов, могло быть сведено к поиску пути в лабиринте альтернативных возможностей.



*Рис. 1. Программы решения интеллектуальных задач*

И хотя эти надежды не оправдались, цикл подобных исследований оказался весьма полезным. Были созданы достаточно эффективные процедуры поиска, используемые специалистами по искусственному интеллекту не только при решении игровых задач, но и во многих других

областях (например, при планировании целесообразной деятельности в интеллектуальных системах).

Переборные игры составляют, по-видимому, большинство во множестве распространенных среди людей игр. Существенно меньшую часть составляют топологические игры, в которых необходимо учитывать не только дерево игры, задаваемое возможными последовательностями ходов противников, но и структурой самой позиции как целого. Примером такой игры может служить го. В этой игре оценка позиции не может быть сведена, как, например, в шахматах, к описанию множества фигур и их расположения на игровом поле. Для го важно не конкретное расположение камней по тем или иным полям, а те конфигурации, которые они образуют на плоскости игрового поля. Программирование таких игр требует создания в памяти ЭВМ эталонных образов тех или иных областей, занятых камнями противников. А это куда более сложная и до конца пока не решенная задача, нежели организация поиска по дереву альтернативных возможностей.

Стохастические игры появляются тогда, когда в процессе игры возникают вероятностные шаги или очередная ситуация формируется при участии некоторого вероятностного механизма. С программированием таких игр (например, карточной игры в очко) связано развитие методов правдоподобного оценивания вариантов, получившего в искусственном интеллекте заметное использование. Во всех таких ситуациях важно уметь пересчитать оценку правдоподобия результирующей ситуации после выбора определенного хода с учетом оценок правдоподобия текущей ситуации и выбора противника.

К стохастическим играм примыкают и игры с неполной информацией, когда при принятии решения необходимо как-то оценивать недостающую информацию. Эти приемы постоянно используются при обращении к содержимому памяти в интеллектуальных системах, когда в ней отсутствует

нужная информация, что является почти стандартной ситуацией при функционировании таких систем в сложных предметных областях.

Компьютерные игры, получившие в последнее время столь широкое распространение, вообще говоря, не относятся традиционно к работам по искусственному интеллекту. Хотя эта ситуация столь же случайна, как и ситуация с распознаванием образов. Конечно, игры с жесткой схемой, в которых "интеллекта" практически нет, не представляют для работ по искусственному интеллекту интереса, но сценарные игры уж точно относятся к рассматриваемой области науки. В них используются сценарии развития игры, движение по которым определяется обоими партнерами. Эти же принципы применяются и в таких типичных для искусственного интеллекта задачах, как организация диалога интеллектуальной системы с пользователем на ограниченном естественном языке, интересны сценарии и для планирования целесообразной деятельности в интеллектуальных работах и других системах искусственного интеллекта.

С самого начала появления ЭВМ стали создаваться программы для машинного перевода и автоматического реферирования текстов. Создание этих программ оказало значительное влияние на развитие искусственного интеллекта, заложило основы тех работ, которые были непосредственно связаны с естественно-языковым общением пользователей с интеллектуальными системами. В системах машинного перевода были разработаны модели и методы, позволяющие автоматически проводить морфологический, синтаксический и во многом семантический анализ фраз естественного языка, нащупаны приемы анализа связного текста. Все эти результаты активно используются при обработке естественно-языковых текстов в интеллектуальных системах. В работах по автоматическому реферированию были заложены основы понимания общей структуры текста как целого, от идеи "что говорится" был сделан переход к идее "о чем говорится". Это позволило на более высоком уровне создавать программы генерации текстов. Если первые программы такого вида основывались на

жестких моделях порождения или вероятностных механизмах, то более поздние программы генерации текстов стали опираться на идеи сценариев, а также на приемы, наработанные в программах по автоматическому реферированию. Сейчас качество прозаических текстов, создаваемых с помощью ЭВМ, достаточно высоко, если тексты имеют жесткую внутреннюю структуру, определяемую их назначением. Таковы, например, волшебные сказки, в основе которых лежит жесткий сценарий поведения действующих лиц, таковы хроникальные заметки или документы, но созданы и достаточно любопытные программы, порождающие поэтические тексты, в которых наблюдается иная крайность — почти полное отсутствие смысловой структуры при достаточно жесткой структуре формы.

Музыкальные программы, пожалуй, наиболее известны широкой публике, так как первые опыты по созданию таких программ сразу дали весьма обнадеживающие результаты. Этот успех связан опять-таки с наличием, с одной стороны, жестких правил при построении мелодии, а с другой стороны, во многом вероятностными моделями, порождающими остальные элементы музыкального произведения. Менее известны широкой публике программы, ориентированные на музыковедов, в которых имитируются стили исполнения или исследуется "анатомия" музыкальных произведений и процесса их сочинения. Однако весь комплекс музыкальных программ, хотя и не оказал прямого влияния на работы по искусственному интеллекту, стал полезным для формирования общего взгляда на природу творческих процессов и их моделирования.

Узнающие программы зародились в недрах исследований по распознаванию образов. Но, как уже говорилось, многие из них оказали значительное влияние на идеи, характерные для работ по созданию интеллектуальных систем, особенно при создании обучающих систем. При их разработке были найдены методы оценивания похожести одних объектов на другие, заложены основы рассуждений по аналогии и ассоциации, использования обучающих последовательностей примеров и контрпримеров;

все это вошло в фонд методов, которыми пользуется специалист по искусственному интеллекту.

Несколько особняком стоят программы, с помощью которых создаются машинные произведения в области графики и живописи. Эти исследования связаны, в основном, с появлением специальных программных и в меньшей мере аппаратных средств для устройств графического вывода. Но косвенно эти программы оказывают влияние на те разделы искусственного интеллекта, которые связаны с использованием зрительных образов при решении задач.

### **Построение систем искусственного интеллекта**

Третье основное направление в создании искусственного интеллекта образует его фундамент. Именно здесь создается теория данного научного направления, решаются основные проблемы, связанные с центральным объектом изучения искусственного интеллекта — знаниями.

Существуют различные подходы к построению систем ИИ. Это разделение не является историческим, когда одно мнение постепенно сменяет другое, и различные подходы существуют и сейчас. Кроме того, поскольку по-настоящему полных систем ИИ в настоящее время нет, то нельзя сказать, что какой-то подход является правильным, а какой-то — ошибочным.

Кратко рассмотрим логический подход. Основой для данного логического подхода служит Булева алгебра. Каждый программист знаком с нею и с логическими операторами с тех пор, когда он осваивал оператор IF. Свое дальнейшее развитие Булева алгебра получила в виде исчисления предикатов — в котором она расширена за счет введения предметных символов, отношений между ними, кванторов существования и всеобщности. Практически каждая система ИИ, построенная на логическом принципе, представляет собой машину доказательства теорем. При этом исходные данные хранятся в базе данных в виде аксиом, правила логического вывода существуют как отношения между ними. Кроме того, каждая такая машина

имеет блок генерации цели, и система вывода пытается доказать данную цель как теорему. Если цель доказана, то трассировка примененных правил позволяет получить цепочку действий, необходимых для реализации поставленной цели. Мощность такой системы определяется возможностями генератора целей и машиной доказательства теорем. Конечно, можно сказать, что выразительности алгебры высказываний не хватит для полноценной реализации ИИ, но стоит вспомнить, что основой всех существующих ЭВМ является бит — ячейка памяти, которая может принимать значения только 0 и 1. Таким образом, было бы логично предположить, что все, что возможно реализовать на ЭВМ, можно было бы реализовать и в виде логики предикатов.

Добиться большей выразительности логическому подходу позволяет такое сравнительно новое направление, как нечеткая логика. Основным ее отличием является то, что правдивость высказывания может принимать в ней, кроме "да/нет" (1/0), еще и промежуточные значения — "не знаю" (0.5), "пациент скорее жив, чем мертв" (0.75), "пациент скорее мертв, чем жив" (0.25). Данный подход больше похож на мышление человека, поскольку мы на вопросы редко отвечаем только "да" или "нет".

Для большинства логических методов характерна большая трудоемкость, поскольку во время поиска доказательства возможен полный перебор вариантов. Поэтому данный подход требует эффективной реализации вычислительного процесса, и хорошая работа обычно гарантируется при сравнительно небольшом размере базы данных.

Под структурным подходом мы подразумеваем здесь попытки построения ИИ путем моделирования структуры человеческого мозга. Одной из первых таких попыток был персептрон Френка Розенблатта. Основной моделируемой структурной единицей в персептронах (как и в большинстве других вариантов моделирования мозга) является нейрон. Позднее возникли и другие модели, которые в просторечии обычно известны под термином



"нейронные сети" (НС). Эти модели различаются по строению отдельных нейронов, по топологии связей между ними и по алгоритмам обучения. Среди наиболее известных сейчас вариантов НС можно назвать НС с обратным распространением ошибки, сети Хопфилда, стохастические нейронные сети. НС наиболее успешно применяются в задачах распознавания образов, в том числе сильно зашумленных, однако имеются и примеры успешного использования их для построения собственно систем ИИ. Для моделей, построенных по аналогии с человеческим мозгом, характерна не слишком большая выразительность, легкое распараллеливание алгоритмов и связанная с этим высокая производительность параллельно реализованных НС. Также для таких сетей характерно одно свойство, которое очень сближает их с человеческим мозгом, — нейронные сети работают даже при условии неполной информации об окружающей среде, то есть, как и человек, они на вопросы могут отвечать не только "да" и "нет", но и "не знаю точно, но скорее да".

Довольно большое распространение получил и эволюционный подход. При построении систем ИИ по данному подходу основное внимание уделяется построению начальной модели, и правилам, по которым она может изменяться (эволюционировать). Причем модель может быть составлена по самым различным методам, это может быть и нейронная сеть, и набор логических правил, и любая другая модель. После этого мы включаем компьютер, и он на основании проверки моделей отбирает самые лучшие из них, на основе которых по самым различным правилам генерируются новые модели, из которых опять выбираются самые лучшие и т. д.

В принципе, можно сказать, что эволюционных моделей как таковых не существует, существуют только эволюционные алгоритмы обучения, но модели, полученные при эволюционном подходе, имеют некоторые характерные особенности, что позволяет выделить их в отдельный класс. Такими особенностями являются перенесение основной деятельности разработчика с построения модели на алгоритм ее модификации и то, что

полученные модели практически не сопутствуют извлечению новых знаний о среде, окружающей систему ИИ, то есть она становится как бы вещью в себе.

Еще один широко используемый подход к построению систем ИИ — имитационный. Данный подход является классическим для кибернетики с одним из ее базовых понятий — "черным ящиком" (ЧЯ). ЧЯ — устройство, программный модуль или набор данных, информация о внутренней структуре и содержании которых отсутствуют полностью, но известны спецификации входных и выходных данных. Объект, поведение которого имитируется, как раз и представляет собой такой "черный ящик". Не важно, что у него и у модели внутри и как он функционирует, главное, чтобы модель в аналогичных ситуациях вела себя точно так же.

Таким образом, здесь моделируется другое свойство человека — способность копировать то, что делают другие, не вдаваясь в подробности, зачем это нужно. Зачастую эта способность экономит ему массу времени, особенно в начале его жизни. Основным недостатком имитационного подхода также является низкая информационная способность большинства моделей, построенных с его помощью.

На практике же очень четкой границы между различными методами и подходами к построению систем ИИ нет. Очень часто встречаются смешанные системы, где часть работы выполняется по одному типу, а часть — по-другому.

## **Состояние и тенденции 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.

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. Выделяют несколько основных направлений этого рынка:

- экспертные системы (ЭС); их часто обозначают еще одним термином - "системы, основанные на знаниях";
- нейронные сети и нечеткие логики;
- естественно-языковые системы.

В США в 1993 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 1993 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 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 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.

### Успехи систем искусственного интеллекта и их причины

Использование экспертных систем и нейронных сетей приносит значительный экономический эффект. Так, например:

- American Express сократила свои потери на 27 млн. долларов в год благодаря экспертной системе, определяющей целесообразность выдачи или отказа в кредите той или иной фирме;
- DEC ежегодно экономит 70 млн. долларов в год благодаря системе XCON/XSEL, которая по заказу покупателя составляет конфигурацию вычислительной системы VAX. Ее использование сократило число ошибок от 30% до 1%;
- Sira сократила затраты на строительство трубопровода в Австралии на 40 млн. долларов за счет управляющей трубопроводом экспертной системы, реализованной на базе системы G2.

Коммерческие успехи к экспертным системам и нейронным сетям пришли не сразу. На протяжении ряда лет (с 1960-х годов) успехи касались в

основном исследовательских разработок, демонстрировавших пригодность систем искусственного интеллекта для практического использования. Начиная примерно с 1985 (а в массовом масштабе, вероятно, с 1988-1990 годов), в первую очередь, экспертные системы, а в последние два года и нейронные сети стали активно использоваться в реальных приложениях.

Причины, приведшие системы искусственного интеллекта к коммерческому успеху, следующие:

1. Специализация. Переход от разработки инструментальных средств общего назначения к проблемно/предметно специализированным средствам, что обеспечивает сокращение сроков разработки приложений, увеличивает эффективность использования инструментария, упрощает и ускоряет работу эксперта, позволяет повторно использовать информационное и программное обеспечение (объекты, классы, правила, процедуры).
2. Использование языков традиционного программирования и рабочих станций. Переход от систем, основанных на языках искусственного интеллекта (Lisp, Prolog и т.п.), к языкам традиционного программирования (C, C++ и т.п.) упростил "интегрированность" и снизил требования приложений к быстродействию и емкости памяти. Использование рабочих станций вместо ПК резко увеличило круг возможных приложений методов искусственного интеллекта.
3. Интегрированность. Разработаны инструментальные средства искусственного интеллекта, легко интегрирующиеся с другими информационными технологиями и средствами (с CASE, СУБД, контроллерами, концентраторами данных и т.п.).
4. Открытость и переносимость. Разработки ведутся с соблюдением стандартов, обеспечивающих данные характеристики.
5. Архитектура клиент/сервер. Разработка распределенной информационной системы в данной архитектуре позволяет снизить стоимость оборудования, используемого в приложении,

децентрализовать приложения, повысить надежность и общую производительность, поскольку сокращается объем информации, пересылаемой между ЭВМ, и каждый модуль приложения выполняется на адекватном оборудовании.

Перечисленные причины могут рассматриваться как общие требования к инструментальным средствам создания систем искусственного интеллекта.

Из пяти факторов, обеспечивших их успех в передовых странах, в России, пожалуй, полностью не реализованы четыре с половиной (хотя в некоторых отечественных системах осуществлен переход к языкам традиционного программирования). Кроме того, в России и СНГ в ряде направлений исследования практически не ведутся, и, следовательно, в этих направлениях (нейронные сети; гибридные системы; рассуждения, основанные на прецедентах; рассуждения, основанные на ограничениях) нельзя ожидать и появления коммерческих продуктов.

Итак, в области искусственного интеллекта наибольшего коммерческого успеха достигли экспертные системы и средства для их разработки. В свою очередь, в этом направлении наибольшего успеха достигли проблемно/предметно специализированные средства. Если в 1988 году доход от них составил только 3 млн. долларов, то в 1993 году - 55 млн. долларов.



## **Архитектура систем, основанных на правилах, на примере экспертной системы**

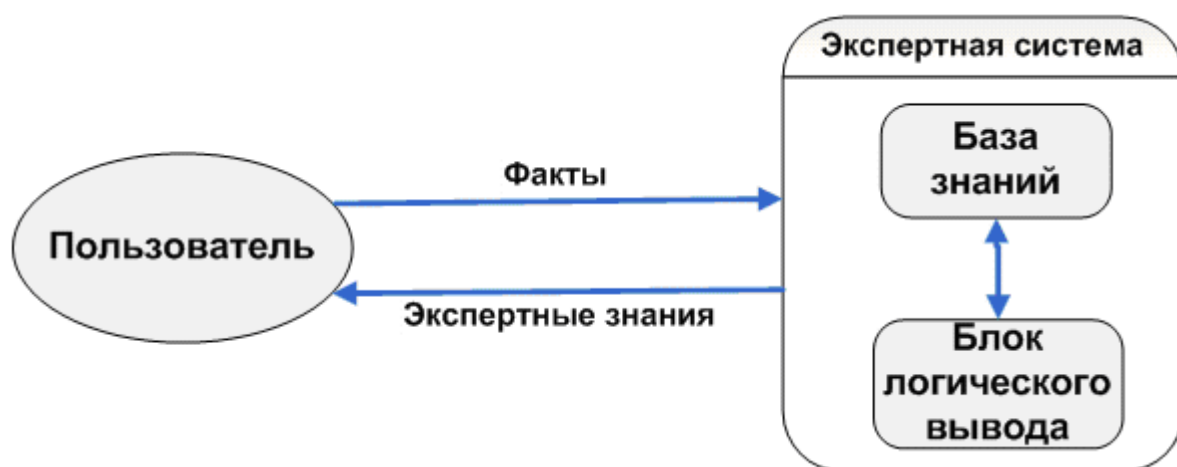
Экспертная система – это интеллектуальная компьютерная программа, в которой используются знания и процедуры логического вывода для решения достаточно трудных задач и требующая для своего решения значительного объема экспертных знаний человека. Таким образом, экспертная система – это компьютерная система, которая эмулирует способности эксперта к принятию решений. Экспертная система обязана действовать во всех отношениях как эксперт-человек. Понятие эмуляции является гораздо более строгим, чем моделирование, поскольку моделирующая система обязана действовать подобно реальному объекту лишь в определенных отношениях.

В экспертных системах для решения задач на уровне эксперта-человека широко используются специализированные знания. Термином «эксперт» обозначается личность, обладающая экспертными знаниями в определенной области. Это означает, что эксперт имеет знания или специальные навыки, которые неизвестны или недоступны для большинства людей. Эксперт способен решать задачи, которые большинство людей не способны решить вообще, или решает их гораздо более эффективно. После того как были впервые разработаны экспертные системы, они содержали исключительно только экспертные знания. Однако в наши дни термин экспертная система часто применяется по отношению к любой системе, в которой используется технология экспертных систем. Технология экспертных систем может включать специальные языки экспертных систем, а также программные и аппаратные средства, предназначенные для обеспечения разработки и эксплуатации экспертных систем.

В качестве знаний в экспертных системах могут применяться либо экспертные знания, либо обычные общедоступные знания, которые могут быть получены из книг, журналов и от хорошо осведомленных людей. В этом

смысле обычные знания рассматриваются как понятие более низкого уровня по сравнению с более редкими экспертными знаниями. Термины экспертная система, система, основанная на знаниях, и экспертная система, основанная на знаниях, часто используются как синонимы. Но большинство людей используют только термин экспертная система, просто потому, что оно короче, даже несмотря на то, что в экспертной системе, о которой идет речь, могут быть представлены не экспертные, а всего лишь обычные знания.

Принципы работы экспертной системы, основанной на знаниях, иллюстрируются на рис. 2. Пользователь передает в экспертную систему факты или другую информацию и получает в качестве результата экспертный совет или экспертные знания. По своей структуре экспертная система подразделяется на два основных компонента - базу знаний и машину логического вывода. База знаний содержит знания, на основании которых машина логического вывода формирует заключения. Эти заключения представляют собой ответы экспертной системы на запросы пользователя, желающего получить экспертные знания.



*Рис. 2. Основные принципы функционирования экспертной системы*

Кроме того, разработаны полезные системы, основанные на знаниях, которые предназначены для использования в качестве интеллектуального

помощника для эксперта-человека. Эти интеллектуальные помощники проектируются на основе технологии экспертных систем, поскольку такая технология обеспечивает значительные преимущества при разработке. Чем больше знаний будет введено в базу знаний интеллектуального помощника, тем в большей степени его действия будут напоминать действия эксперта. Разработка интеллектуального помощника может стать полезным промежуточным шагом перед созданием полноценной экспертной системы. К тому же интеллектуальный помощник позволяет освободить для эксперта больше полезного времени, поскольку его применение способствует ускоренному решению задач.

Знания эксперта относятся только к одной предметной области, и в этом состоит отличие методов, основанных на использовании экспертных систем, от общих методов решения задач. Предметная область - это специальная проблемная область, такая как медицина, финансы, наука и техника, в которой может очень хорошо решать задачи лишь определенный эксперт. Экспертные системы, как и эксперты-люди, в целом предназначены для использования в качестве экспертов в одной предметной области. Например, обычно нельзя рассчитывать на то, что эксперт в области шахмат будет обладать экспертными знаниями, относящимися к медицине. Экспертные знания в одной предметной области не переносятся автоматически на другую область.

Знания эксперта, касающиеся решения конкретных задач, называются областью знаний эксперта. Например, медицинская экспертная система, предназначенная для диагностирования инфекционных заболеваний, должна обладать большим объемом знаний об определенных симптомах, вызванных инфекционными заболеваниями. В этом случае областью знаний является медицина, а сами знания состоят из сведений о заболеваниях, симптомах и методах лечения. Связь между предметной областью и областью знаний показана на рис. 3. На данном рисунке область знаний полностью включена в предметную область. Часть, выходящая за пределы области знаний,

символизирует область, в которой отсутствуют знания о какой-либо из задач, относящихся к данной предметной области.



*Рис. 3. Связь между предметной областью и областью знаний*

В области знаний экспертная система проводит рассуждения или делает логические выводы по такому же принципу, как рассуждал бы эксперт-человек или приходил логическим путем к решению задачи. Это означает, что на основании определенных фактов путем рассуждений формируется логичное, оправданное заключение, которое следует из этих фактов. Например, если ваш близкий родственник не разговаривал с вами месяц, вы могли бы прийти к выводу, что ему просто нечего вам сказать. Но это лишь один из нескольких возможных логических выводов.

### **Общие понятия экспертных систем**

Знания могут быть представлены в экспертной системе многими способами. Одним из широко применяемых методов представления знаний являются правила в форме IF-THEN, как показано ниже.

IF горит красный свет THEN стоять

Если обнаруживается факт, что на светофоре горит красный свет, то этот факт согласуется с шаблоном "на светофоре горит красный свет". Условие правила удовлетворяется и выполняется обусловленное в нем действие "стоять".

Классическая экспертная система воплощает в себе неписанные знания, которые должны быть получены от эксперта с помощью интервью, проводимых инженером по знаниям в течение длительного периода времени. Такой процесс создания экспертной системы называется инженерией знаний и осуществляется инженером по знаниям. Инженерией знаний называют получение знаний от эксперта-человека или из других источников и последующее представление знаний в экспертной системе.

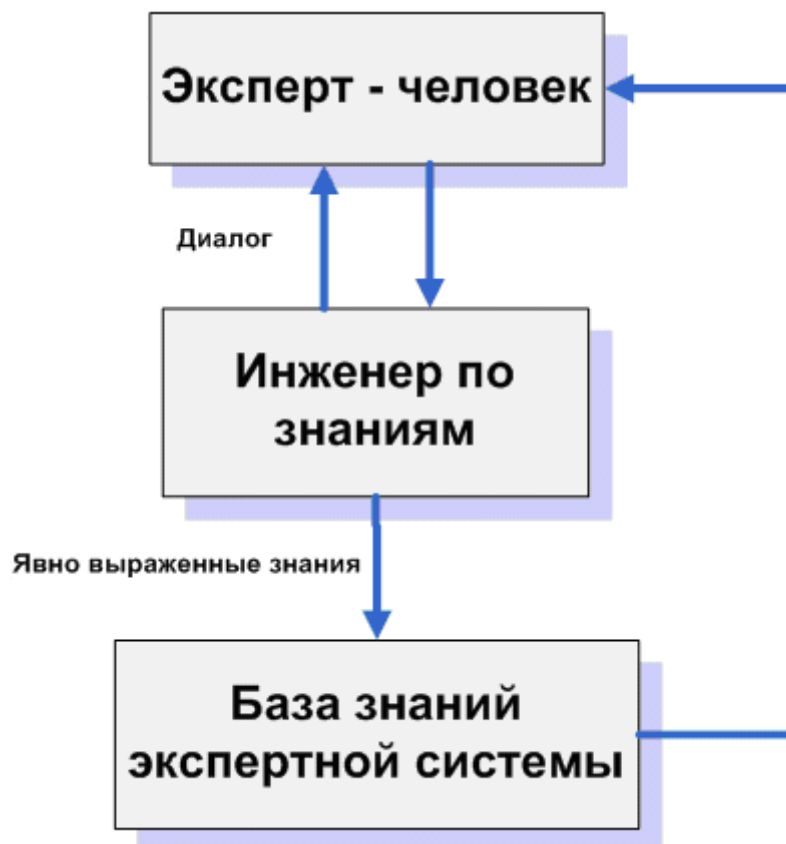


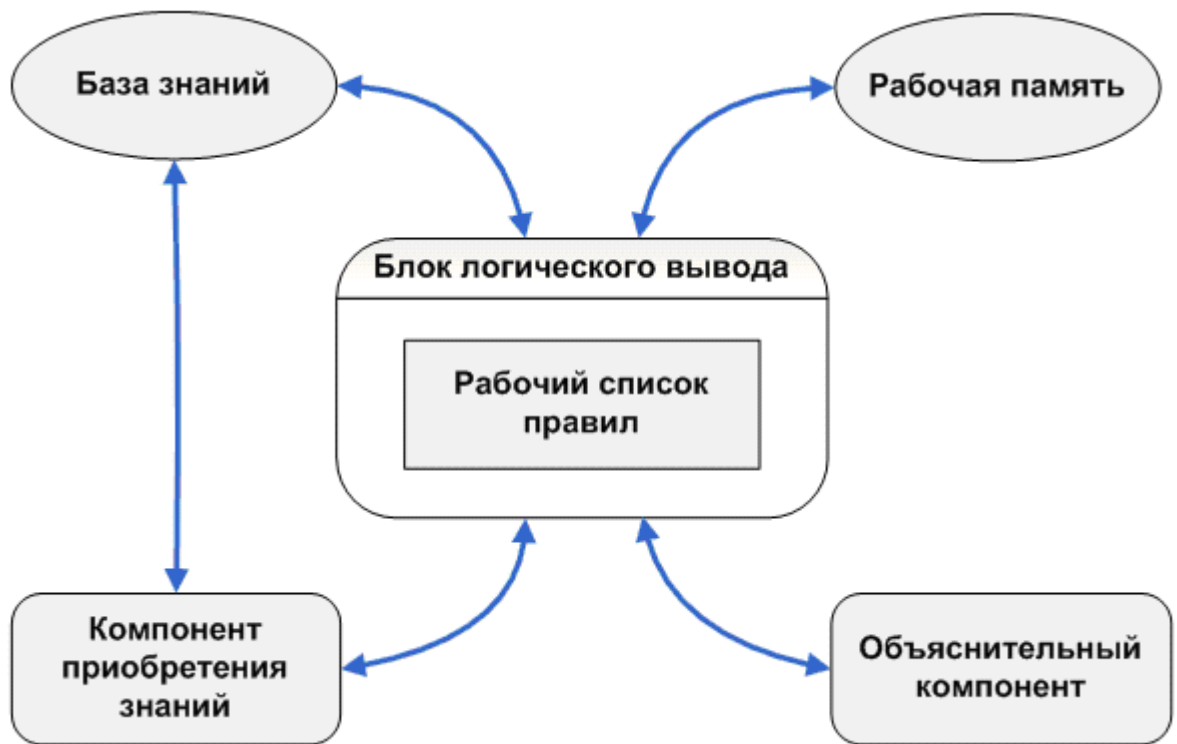
Рис. 4. Процесс разработки экспертной системы

Основные этапы разработки экспертной системы показаны на рис. 4. Вначале инженер по знаниям устанавливает диалог с экспертом-человеком, чтобы выявить знания эксперта. Этот этап аналогичен этапу работы, выполняемому системным проектировщиком при обычном программировании в ходе обсуждения требований к системе с клиентом, для которого создается программа. Затем инженер по знаниям представляет знания в явном виде для внесения в базу знаний. После этого эксперт проводит оценку экспертной системы и передает критические замечания инженеру по знаниям. Такой процесс повторяется снова и снова, до тех пор, пока эксперт не оценит результаты работы системы как удовлетворительные.

Вообще говоря, процесс создания экспертных систем намного отличается от процесса разработки обычных программ, ведь в экспертных системах рассматриваются задачи, не имеющие удовлетворительного алгоритмического решения, поэтому для достижения приемлемого решения используется логический вывод. Поскольку в основе функционирования экспертной системы лежит логический вывод, такая система должна обладать способностью объяснить свои рассуждения, чтобы можно было их проверить. Поэтому неотъемлемой частью любой сложной экспертной системы является средство объяснения. В действительности могут быть разработаны сложные средства объяснения, позволяющие пользователю исследовать многочисленные строки с вопросами наподобие "Что будет, если... ", называемые гипотетическими рассуждениями.

### **Элементы экспертной системы**

Элементы типичной экспертной системы показаны на рис. 5. В системе, основанной на правилах, знания в проблемной области, необходимые для решения задач, закодированы в форме правил и содержатся в базе знаний. Безусловно, для представления знаний наиболее широко применяются правила.



*Рис. 5. Структура экспертной системы, основанной на правилах*

Экспертная система состоит из описанных ниже компонентов.

- Пользовательский интерфейс. Механизм, с помощью которого происходит общение пользователя и экспертной системы.
- Средство объяснения. Компонент, позволяющий объяснить пользователю ход рассуждений системы.
- Рабочая память. Глобальная база фактов, используемых в правилах.
- Машина логического вывода. Программный компонент, который обеспечивает формирование логического вывода (принимая решение о том, каким правилам удовлетворяют факты или объекты), располагает выполняемые правила по приоритетам и выполняет правило с наивысшим приоритетом.
- Рабочий список правил. Созданный машиной логического вывода и расположенный по приоритетам список правил, шаблоны которых удовлетворяют фактам или объектам, находящимся в рабочей памяти.

- Средство приобретения знаний. Автоматизированный способ, позволяющий пользователю вводить знания в систему, а не привлекать к решению задачи явного кодирования знаний инженера по знаниям.

Во многих системах имеется необязательное средство приобретения знаний, Это инструментальное средство в некоторых экспертных системах способно обучаться, осуществляя вывод правил по методу индукции на основании примеров, и автоматически вырабатывать правила. Для выработки правил в машинном обучении применялись также другие методы и алгоритмы, такие как искусственные нейронные сети и генетические алгоритмы. Основная проблема, возникающая при использовании машинного обучения для выработки правил, состоит в том, что отсутствует какое-либо объяснение, почему были созданы эти правила. В отличие от человека, способного объяснить причины, по которым было выбрано то или иное правило, системы машинного обучения никогда не были в состоянии объяснить свои действия, а это может повлечь за собой появление непредсказуемых результатов. Однако в целом для создания деревьев решений лучше всего подходят примеры, представленные в виде простых таблиц. Общие правила, подготовленные инженером по знаниям, могут быть намного сложнее по сравнению с простыми правилами, полученными путем вывода правил по методу индукции.

В экспертной системе, основанной на правилах, базу знаний называют также продукционной памятью. В качестве простого примера рассмотрим задачу принятия решения о переходе через дорогу. Ниже приведены продукции для двух правил, в которых стрелки означают, что система осуществит действия справа от стрелки, если условия слева от стрелки будут истинными.

горит красный свет —> стоять

горит зеленый свет —> двигаться



Производственные правила могут быть выражены в эквивалентном формате псевдокода IF-THEN следующим образом:

Правило: red\_light

IF горит красный свет THEN стоять

Правило: green\_light

IF горит зеленый свет THEN двигаться

Каждое правило обозначается именем. Вслед за именем находится часть IF правила. Участок правила между частями IF и THEN правила упоминается под разными именами, такими как антецедент, условная часть, часть шаблона или левая часть (left-hand-side — LHS). Такое отдельно взятое условие, как “горит красный свет” называется условным элементом, или шаблоном.

В системе, основанной на правилах, машина логического вывода определяет, какие антецеденты правил (если таковые вообще имеются) выполняются согласно фактам. В качестве стратегий решения задач в экспертных системах обычно используются два общих метода логического вывода: прямой логический вывод и обратный логический вывод. В число других методов, применяемых для выполнения более конкретных методов, могут входить анализ целей и средств, упрощение задачи, перебор с возвратами, метод "запланировать-выработать-проверить", иерархическое планирование и принцип наименьшего вклада, а также обработка ограничений.

Прямой логический вывод представляет собой метод формирования рассуждений от фактов к заключениям, которые следуют из этих фактов. Например, если перед выходом из дома вы обнаружите, что идет дождь (факт), то должны взять с собой зонтик (заключение).

Обратный логический вывод предусматривает формирование рассуждений в обратном направлении - от гипотезы (потенциального заключения, которое должно быть доказано) к фактам, которые

подтверждают гипотезу. Например, если вы не выглядываете наружу, но кто-то вошел в дом с влажными ботинками и зонтиком, то можно принять гипотезу, что идет дождь. Чтобы подтвердить эту гипотезу, достаточно спросить данного человека, идет ли дождь. В случае положительного ответа будет доказано, что гипотеза истинна, поэтому она становится фактом. Как уже было сказано выше, гипотеза может рассматриваться как факт, истинность которого вызывает сомнение и должна быть установлена. В таком случае гипотеза может интерпретироваться как цель, которая должна быть доказана.

В зависимости от проекта экспертной системы в машине логического вывода осуществляется либо прямой, либо обратный логический вывод, либо обе эти формы логического вывода. Например, язык CLIPS спроектирован в расчете на применение прямого логического вывода, в языке PROLOG осуществляется обратный логический вывод, а в версии CLIPS, называемой Eclipse, разработанной Полом Хэйли (Paul Haley), осуществляется и прямой, и обратный логический выводы. Выбор машины логического вывода зависит от типа задачи. Диагностические задачи лучше всего решать с помощью обратного логического вывода, в то время как задачи прогнозирования, текущего контроля и управления проще всего поддаются решению с помощью прямого логического вывода.

Рабочая память может содержать факты, касающиеся текущего состояния светофора, такие как "горит зеленый свет" или "горит красный свет". В рабочей памяти могут присутствовать любой из этих фактов или оба факта одновременно. Если светофор работает нормально, то в рабочей памяти будет находиться только один факт. Но возможно также, что в рабочей памяти будут присутствовать оба факта, если светофор неисправен. Обратите внимание на то, в чем состоит различие между базой знаний и рабочей памятью. Факты не взаимодействуют друг с другом. Факт "горит зеленый свет" не воздействует на факт "горит красный свет". С другой

стороны, наши знания о работе светофоров говорят о том, что если одновременно присутствуют оба факта, то светофор неисправен.

Если в рабочей памяти имеется факт "горит зеленый свет", машина логического вывода обнаруживает, что этот факт удовлетворяет условной части правила `green_light` и помещает это правило в рабочий список правил. А если правило имеет несколько шаблонов, то все эти шаблоны должны быть удовлетворены одновременно для того, чтобы правило можно было поместить в рабочий список правил. В качестве условия удовлетворения некоторых шаблонов можно даже указать отсутствие определенных фактов в рабочей памяти.

Правило, все шаблоны которого удовлетворены, называется активизированным, или реализованным. В рабочем списке правил может одновременно присутствовать несколько активизированных правил. В этом случае машина логического вывода должна выбрать одно из правил для запуска.

Вслед за частью THEN правила находится список действий, которые должны быть выполнены после запуска правила. Эта часть правила называется консеквентом, или правой частью (Right-Hand Side — RHS). Если происходит запуск правила `red_light`, выполняется его действие "стоять". Аналогичным образом после запуска правила `green_light` его действием становится "двигаться". В состав конкретных действий обычно входит добавление или удаление фактов из рабочей памяти либо вывод результатов. Формат описания этих действий зависит от синтаксиса языка экспертной системы. Например, в языке CLIPS действие по добавлению в рабочую память нового факта, называемого "stop" (стоять), принимает вид (`assert stop`).

Машина логического вывода работает в режиме осуществления циклов "распознавание-действие". Для описания указанного режима работы применяются также другие термины, такие как цикл "выборка-выполнение",

цикл “ситуация-отклик” и цикл “ситуация-действие”. Но как бы ни назывался такой цикл, машина логического вывода снова и снова выполняет некоторые группы задач до выявления определенных критериев, которые вызывают прекращение выполнения. При этом решаются общие задачи, обозначенные в приведенном ниже псевдокоде как разрешение конфликтов, действие, согласование и проверка условий останова.

WHILE работа не закончена

Разрешение конфликтов.

Если имеются активизированные правила, то выбрать правило с наивысшим приоритетом; в противном случае работа закончена.

Действие.

Последовательно осуществить действия, указанные в правой части выбранного активизированного правила. В данном цикле проявляется непосредственное влияние тех действий, которые изменяют содержимое рабочей памяти. Удалить из рабочего список правил только что запущенное правило.

Согласование.

Обновить рабочий список правил путем проверки того, выполняется ли левая часть каких-либо правил. В случае положительного ответа активизировать соответствующие правила. Удалить активизированные правила, если левая часть соответствующих правил больше не выполняется.

Проверка условий останова.

Если осуществлено действие halt или дана команда break, то работа закончена.

END-WHILE

Принять новую команду пользователя.

В течение каждого цикла могут быть активизированы и помещены в рабочий список правил многочисленные правила. Кроме того, в рабочем списке правил остаются результаты активизации правил от предыдущих циклов, если не происходит деактивизация этих правил в связи с тем, что их

левые части больше не выполняются. Таким образом в ходе выполнения программы количество активизированных правил в рабочем списке правил изменяется. В зависимости от программы ранее активизированные правила могут всегда оставаться в рабочем списке правил, но никогда не выбираться для запуска. Аналогичным образом некоторые правила могут никогда не становиться активизированными. В подобных случаях следует повторно проверять назначение этих правил, поскольку либо такие правила не нужны, либо их шаблоны неправильно спроектированы.

Машина логического вывода выполняет действия активизированного правила с наивысшим приоритетом из рабочего списка правил, затем – действия активизированного правила со следующим по порядку приоритетом и т.д., до тех пор, пока в списке не останется больше активизированных правил. Для инструментальных средств экспертных систем разработаны различные системы приоритетов, но, вообще говоря, все инструментальные средства позволяют инженеру по знаниям определять приоритеты правил.

В рабочем списке правил возникают конфликты, если различные активизированные правила имеют одинаковый приоритет и машина логического вывода должна принять решение о том, какое из этих правил необходимо запустить. В различных командных интерпретаторах для решения этой проблемы применяются разные способы. Ньюэлл и Саймон использовали такой подход, что правила, введенные в систему в первую очередь, приобретают по умолчанию наивысший приоритет. В языке CLIPS правила имеют по умолчанию одинаковый приоритет, если каким-то из них не присваивается другой приоритет инженером по знаниям.

После завершения выполнения всех правил управление возвращается к интерпретатору команд верхнего уровня, чтобы пользователь мог выдать командному интерпретатору экспертной системы дополнительные инструкции. Работа в режиме верхнего уровня соответствует применяемому по умолчанию режиму, в котором пользователь взаимодействует с

экспертной системой, и обозначается как задача "Accept a new user command" (Принять новую команду пользователя). Прием новых команд происходит именно на верхнем уровне.

Верхний уровень представляет собой пользовательский интерфейс к командному интерпретатору в тот период, когда происходит разработка приложения экспертной системы. Но обычно разрабатываются более сложные пользовательские интерфейсы, позволяющие упростить работу с экспертной системой. В действительности проектирование и реализация пользовательского интерфейса могут потребовать больше усилий, чем создание базы знаний экспертной системы, особенно на стадии разработки прототипа. В зависимости от возможностей командного интерпретатора экспертной системы пользовательский интерфейс может быть реализован с помощью правил или с применением операторов на другом языке, вызываемых из экспертной системы.

Главной особенностью экспертной системы является предусмотренное в ней средство объяснения, которое дает возможность пользователю задавать вопросы о том, как система пришла к определенному заключению и для чего ей требуется определенная информация. Система, основанная на правилах, способна легко ответить на вопрос о том, как было получено определенное заключение, поскольку хронология активизации правил и содержимое рабочей памяти можно сохранять в стеке. Но такая возможность не столь легко достижима при использовании искусственных нейронных сетей, генетических алгоритмов или других систем, разработка которых еще продолжается. Безусловно, были сделаны попытки предусмотреть в некоторых системах возможность объяснения, но созданные при этом средства объяснения не могут сравниться по своей наглядности со средствами любой экспертной системы, спроектированной человеком. Развитые средства объяснения могут дать возможность пользователю задавать вопросы типа "что, если" и изучать альтернативные пути формирования рассуждений по принципу гипотетических рассуждений.

## Продукционные системы

В настоящее время экспертными системами наиболее широко применяемого типа являются системы, основанные на правилах. В системах, основанных на правилах, знания представлены не с помощью относительно декларативного, статического способа (как ряд истинных утверждений), а в форме многочисленных правил, которые указывают, какие заключения должны быть сделаны или не сделаны в различных ситуациях. Система, основанная на правилах, состоит из правил IF-THEN, фактов и интерпретатора, который управляет тем, какое правило должно быть вызвано в зависимости от наличия фактов в рабочей памяти.

Системы, основанные на правилах, относятся к двум главным разновидностям: системы с прямым логическим выводом и системы с обратным логическим выводом. Система с прямым логическим выводом начинает свою работу с известных начальных фактов и продолжает работу, используя правила для вывода новых заключений или выполнения определенных действий. Система с обратным логическим выводом начинает свою работу с некоторой гипотезы, или цели, которую пользователь пытается доказать, и продолжает работу, отыскивая правила, которые позволят доказать истинность гипотезы. Для разбиения крупной задачи на мелкие фрагменты, которые можно будет более легко доказать, создаются новые подцели. Системы с прямым логическим выводом в основном являются управляемыми данными, а системы с обратным логическим выводом - управляемыми целями.

Широкое применение систем, основанных на правилах, обусловлено описанными ниже причинами.

- Модульная организация. Благодаря такой организации упрощается представление знаний и расширение экспертной системы по методу инкрементной разработки.

- Наличие средств объяснения. Такие экспертные системы позволяют легко создавать средства объяснения с помощью правил, поскольку антецеденты правила точно указывают, что необходимо для активизации правила. Средство объяснения позволяет следить за тем, запуск каких правил был осуществлен, поэтому дает возможность восстановить ход рассуждений, которые привели к определенному заключению.
- Наличие аналогии с познавательным процессом человека. Согласно результатам, полученным Ньюэллом и Саймоном, правила, по-видимому, представляют собой естественный способ моделирования процесса решения задач человеком. А при осуществлении попытки выявить знания, которыми обладают эксперты, проще объяснить экспертам структуру представления знаний, поскольку применяется простое представление правил IF-THEN.

### *Продукционные системы Поста*

Продукционные системы были впервые использованы в символической логике Постом (Post), поэтому имя этого ученого вошло в название указанных систем. Пост доказал такой важный и неожиданный результат, что любая система математики или логики может быть оформлена в виде системы продукционных правил определенного типа. Этот результат показал огромные возможности применения продукционных правил для представления важных классов знаний, а это означает, что продукционные правила не сводятся к нескольким ограниченным типам. Кроме того, продукционные правила, обозначаемые термином правила подстановки, используются также в лингвистике как способ определения грамматики языка. Компьютерные языки обычно определяются с помощью формы продукционных правил, известной как нормальная форма Бэкуса-Наура (Backus-Naur Form — BNF). Основная идея Поста заключалась в том, что любая математическая или логическая система представляет собой набор



правил, который указывает, как преобразовать одну строку символов в другой последовательный набор символов. Это означает, что продукционное правило после получения входной строки (антецедента) способно выработать новую строку (консеквент). Такая идея является также действительной по отношению к программам и экспертным системам, в которых начальная строка символов представляет собой входные данные, а выходная строка становится результатом определенных преобразований, которым были подвергнуты входные данные.

В качестве очень простого случая можно представить себе, что если входной строкой является "у пациента имеется высокая температура", то выходной строкой может быть "пациент должен принять аспирин". Обратите внимание на то, что за этими строками не закреплен какой-либо смысл. Иными словами, манипуляции со строками основаны на синтаксисе, а не на семантике, т.е. не на понимании того, что скрывается за словами "высокая температура", "аспирин" и "пациент". Люди знают, что означают эти строки в реальном мире, а продукционная система Поста применяется лишь в качестве способа преобразования одной строки в другую. Для данного примера может быть предусмотрено следующее продукционное правило:

антецедент —> консеквент

у пациента имеется высокая температура —> пациент должен принять аспирин

В этом правиле стрелка означает, что одна строка должна быть преобразована в другую. Указанное правило можно интерпретировать с помощью более знакомой системы обозначений IF-THEN следующим образом:

IF у пациента имеется высокая температура THEN пациент должен принять аспирин.

## Rete-алгоритм

Если требуется создать экспертную систему для решения реальных задач, содержащую сотни или тысячи правил, то проблема эффективности приобретает наибольшую важность. Независимо от того, насколько приемлемыми являются все прочие характеристики системы, если пользователю придется долго ожидать ответа, то он не станет работать с такой системой. Поэтому фактически требуется алгоритм, который имеет полную информацию обо всех правилах и может применить любое нужное правило, не предпринимая попытки последовательно проверять каждое правило.

Решением этой проблемы является rete-алгоритм, разработанный Чарльзом Л. Форги (Charles L. Forgy) в университете Карнеги-Меллона в 1979 году в рамках диссертации по командному интерпретатору экспертной системы OPS (Official Production System — стандартная производственная система) на получение степени доктора философии. Термин rete-алгоритм происходит от латинского слова rete, которое означает сеть. Rete-алгоритм функционирует как сеть, предназначенная для хранения большого объема информации и обеспечивающая значительное сокращение времени отклика и повышение быстродействия при запуске правил по сравнению с большими группами правил IF-THEN, которые должны проверяться один за другим в обычной системе, основанной на правилах. Rete-алгоритм основан на использовании динамической структуры данных, подобной стандартному дереву B+, которая автоматически реорганизуется в целях оптимизации поиска.

Rete-алгоритм представляет собой очень быстрое средство сопоставления с шаблонами, высокое быстродействие которого достигается благодаря хранению в оперативной памяти информации о правилах, находящихся в сети. Этот алгоритм предназначен для повышения быстродействия систем с прямым логическим выводом, основанных на

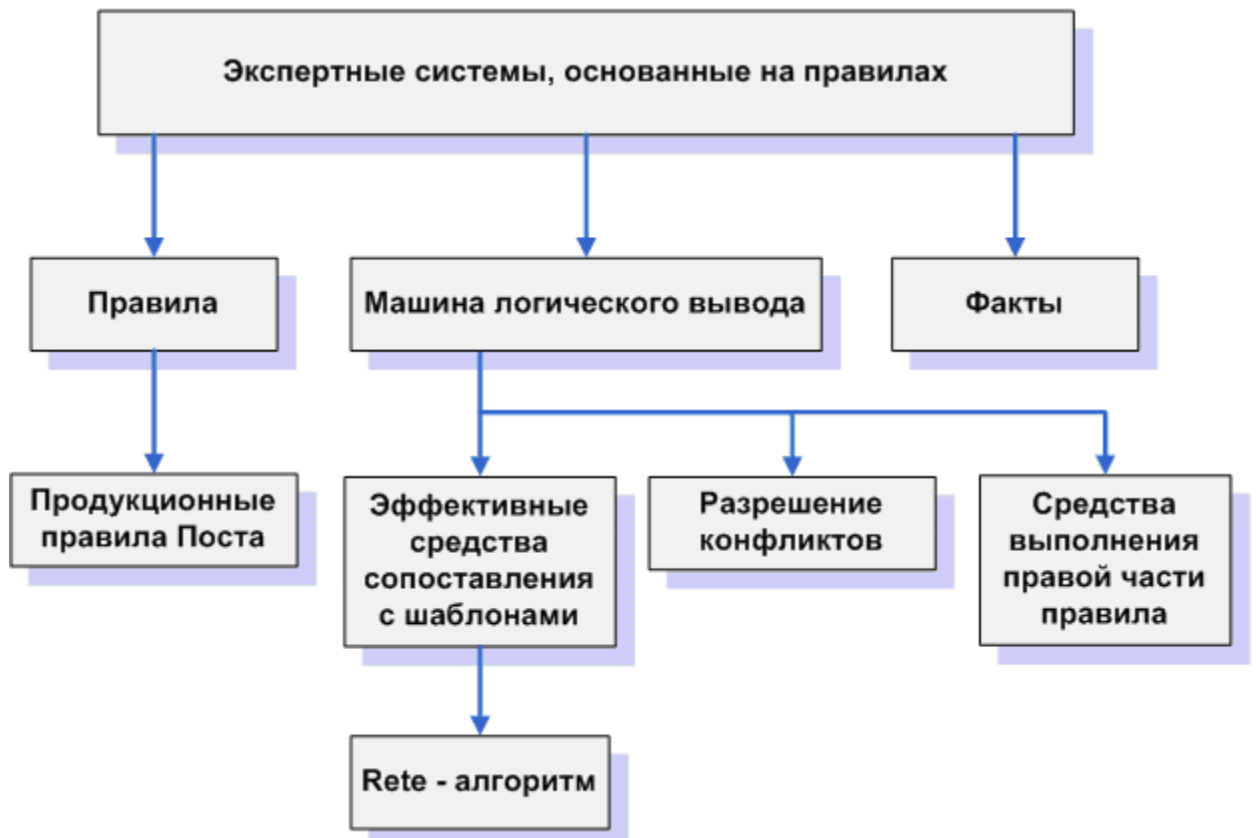
правилах, благодаря ограничению объема работы, требуемой для повторного вычисления конфликтного множества после запуска одного из правил. Недостатком этого алгоритма является его высокие потребности в памяти, но в наши дни, когда микросхемы памяти стали такими дешевыми, этот недостаток не имеет большого значения. В *rete*-алгоритме воплощены два описанных ниже эмпирических наблюдения, на основании которых была предложена структура данных, лежащая в его основе.

- **Временная избыточность.** Изменения, возникающие в результате запуска одного из правил, обычно затрагивают лишь несколько фактов, а каждое из этих изменений влияет только на несколько правил.
- **Структурное подобие.** Один и тот же шаблон часто обнаруживается в левой части больше чем одного правила.

Если в системе заданы сотни или тысячи правил, то подход к организации работы, в котором компьютер последовательно проверяет вероятность того, должен ли быть выполнен запуск каждого правила, становится очень неэффективным. Благодаря разработке *rete*-алгоритма появилась практическая возможность создания инструментальных средств экспертных систем даже на тех медленных компьютерах, которые применялись в 1970-х годах. В наши дни *rete*-алгоритм продолжает оставаться важным средством повышения быстродействия в тех случаях, когда экспертная система содержит много правил.

В *rete*-алгоритме в каждом цикле контролируются только изменения в согласованиях, поэтому в каждом цикле "распознавание-действие" не приходится согласовывать факты с каждым правилом. Благодаря этому существенно повышается скорость согласования фактов с антецедентами, поскольку статические данные, которые не изменяются от цикла к циклу, могут быть проигнорированы.

На рис. 6 показаны технологии, которые образуют фундамент современных экспертных систем, основанных на правилах.



*Рис. 6. Технологии, образующие фундамент современных экспертных систем, основанных на правилах*

## Современные разработки в области искусственного интеллекта

Какие возможности предоставляет искусственный интеллект в наши дни? Краткий ответ на этот вопрос сформулировать сложно, поскольку в этом научном направлении существует слишком много подобластей, в которых выполняется очень много исследований. Ниже приведен краткий обзор современных разработок с применением технологий искусственного интеллекта.

- Автономное планирование и составление расписаний. Работающая на удалении в сотни миллионов километров от Земли программа Remote Agent агентства NASA стала первой бортовой автономной программой планирования, предназначенной для управления процессами составления расписания операций для космического аппарата. Программа Remote Agent вырабатывала планы на основе целей высокого уровня, задаваемых с Земли, а также контролировала работу космического аппарата в ходе выполнения планов: обнаруживала, диагностировала и устраняла неполадки по мере их возникновения.
- Ведение игр. Программа Deep Blue компании IBM стала первой компьютерной программой, которой удалось победить чемпиона мира в шахматном матче, после того как она обыграла Гарри Каспарова со счетом 3.5:2.5 в показательном матче. Каспаров заявил, что ощущал напротив себя за шахматной доской присутствие «интеллекта нового типа». Журнал Newsweek описал этот матч под заголовком «Последний оборонительный рубеж мозга». Стоимость акций IBM выросла на 18 миллиардов долларов.
- Автономное управление. Система компьютерного зрения Alvinn была обучена вождению автомобиля, придерживаясь определенной полосы движения. В университете CMU эта система была размещена в микроавтобусе, управляемом компьютером NavLab, и использовалось для проезда по Соединенным Штатам; на протяжении 4586.6 км

система обеспечивала рулевое управление автомобилем в течение 98% времени. Человек брал на себя управление лишь в течение остальных 2%, главным образом на выездных пандусах. Компьютер NavLab был оборудован видеокамерами, которые передавали изображения дороги в систему Alvin, а затем эта система вычисляла наилучшее направление движения, основываясь на опыте, полученном в предыдущих учебных пробегах.

- **Диагностика.** Медицинские диагностические программы, основанные на вероятностном анализе, сумели достичь уровня опытного врача в нескольких областях медицины. Хекерман описал случай, когда ведущий специалист в области патологии лимфатических узлов не согласился с диагнозом программы в особо сложном случае. Создатели программы предложили, чтобы этот врач запросил у компьютера пояснения по поводу данного диагноза. Машина указала основные факторы, повлиявшие на ее решение, и объяснила нюансы взаимодействия нескольких симптомов, наблюдавшихся в данном случае. В конечном итоге эксперт согласился с решением программы.
- **Планирование снабжения.** Во время кризиса в Персидском заливе в 1991 году в армии США была развернута система DART (Dynamic Analysis and Replanning) для обеспечения автоматизированного планирования поставок и составления графиков перевозок. Работа этой системы охватывала одновременно до 50 000 автомобилей, единиц груза и людей; в ней приходилось учитывать пункты отправления и назначения, маршруты, а также устранять конфликты между всеми параметрами. Методы планирования на основе искусственного интеллекта позволяли вырабатывать в течение считанных часов такие планы, для составления которых старыми методами потребовались бы недели. Представители агентства DARPA (Defense Advanced Research Project Agency – Управление перспективных исследовательских программ) заявили, что одно лишь это приложение стоицей окупило

тридцатилетние инвестиции в искусственный интеллект, сделанные этим агентством.

- Робототехника. Многие хирурги теперь используют роботов-ассистентов в микрохирургии. Например, HipNav – это система, в которой используются методы компьютерного зрения для создания трехмерной модели анатомии внутренних органов пациента, а затем применяется робототехническое управление для руководства процессом вставки протеза, заменяющего тазобедренный сустав.
- Понимание естественного языка и решение задач. Программа Proverb – это компьютерная программа, которая решает кроссворды намного лучше, чем большинство людей; в ней используются ограничения, определяющие состав возможных заполнителей слов, большая база с данными о встречавшихся ранее кроссвордах, а также множество различных источников информации, включая словари и оперативные базы данных, такие как списки кинофильмов и актеров, которые играли в этих фильмах. Например, эта программа способна определить, что одним из решений, подходящих для ключа «Nice Story», является слово «ETAGE», поскольку ее база данных содержит пару ключ-решение «Story in France/ETAGE», а сама программа распознает, что шаблоны «Nice X» и «X in France» часто имеют одно и то же решение. Программа не знает, что Nice (Ницца) – город во Франции, но способна разгадать эту головоломку.

Выше приведено лишь несколько примеров систем искусственного интеллекта, которые существуют в настоящее время.

### **Применение технологии многоагентных систем для интеллектуальной поддержки принятия решения**

Главным отличием современных интеллектуальных систем является их распределенность, обеспечение обработки и применение распределенных знаний. Главный смысл происходящих в настоящее время смен концепций

(парадигмы) создания и использования средств искусственного интеллекта – переход от предположений, справедливых только для изолированных систем искусственного интеллекта, от индивидуальных систем к распределенной обработке информации и разработке многоагентных интеллектуальных систем (МАС).

При практической реализации распределенных систем возникли серьезные трудности с проектированием и описанием объединенных в единую сеть разнородных локальных компьютерных узлов. В последнее время в рамках общего научного направления «искусственный интеллект» активно ведутся исследования под объединенным названием «многоагентные системы».

Многоагентные системы зародились на пересечении теории систем и теории систем распределенного искусственного интеллекта (Distributed Artificial Intelligence (DAI)-систем).

С одной стороны, речь идет об открытых, активных, развивающихся системах, в которых главное внимание уделяется процессу взаимодействия агентов как причине возникновения системы с новыми качествами (концепция возникновения).

С другой стороны, многоагентные системы могут строиться по принципам распределенного интеллекта как объединение отдельных интеллектуальных систем, обладающих своими базами знаний и средствами рассуждений.

### *Технология многоагентных систем*

Технология многоагентных систем – это новая парадигма информационной технологии, ориентированной на совместное использование научных и технических достижений и преимуществ, которые дают идеи и методы искусственного интеллекта, современные локальные и глобальные компьютерные сети, распределенные базы данных и распределенные вычисления, аппаратные и программные средства



поддержки теории распределенности и открытости. Принципиальным отличием новой парадигмы построения прикладных систем является то, что в ней определяющим являются данные (факты), которые указывают направление вычислений.

Агент – это развитие известного понятия «объект», представляющего абстракцию множества экземпляров предметов реального мира, имеющих одни и те же свойства и правила поведения. Точное определение агента на сегодняшний день отсутствует. В основном используется определение, принятое на конференции международной ассоциации по лингвистике FIRA (Federation of Intelligent Physical Agents) в октябре 1996 года в Токио: «Агент – это сущность, которая находится в некоторой среде, интерпретирует их и исполняет команды, воздействующие на среду. Агент может содержать программные и аппаратные компоненты. Отсутствие четкого определения мира агентов и присутствие большого количества атрибутов, с ним связанных, а также существование большого разнообразия примеров агентов говорят о том, что агенты – это достаточно общая технология, которая аккумулирует в себе несколько различных областей». Агент – это программный модуль, способный выполнять определенные ему функции или функции другого агента (человека, чьи функции он воспроизводит). В данной работе под агентом понимается аппаратная или программная сущность, способная действовать в интересах достижения целей, поставленных перед ним владельцем или пользователем.

Фактически, используя понятие «агент», каждый автор или сообщество определяют своего агента с конкретным набором свойств в зависимости от целей разработки, решаемых задач, техники реализации, критериев. Как следствие, в рамках данного направления появилось множество типов агентов, например: автономные агенты, персональные ассистенты, интеллектуальные агенты, социальные агенты и т.д. А в зависимости от степени возможности внутреннего представления внешнего мира и способа поведения агенты классифицируются как локальные, сетевые, мобильные,

интерфейсные, транслирующие, маршрутизации и т.д. В таблице 1 приведен пример классификации агентов.

Характеристики	Типы агентов			
	Простые	Смышленные (smart)	Интеллектуальные (intelligent)	Действительно интеллектуальные (truly)
Автономное выполнение	+		+	+
Взаимодействие с другими агентами и/или пользователями	+	+	+	+
Слежение за окружением	+	+	+	+
Способность использования абстракций		+	+	+
Способность использования предметных знаний		+	+	
Возможность адаптивного поведения для достижения целей			+	+
Обучение из окружения			+	+
Толерантность к ошибкам и/или неверным входным сигналам			+	
Real-time исполнения			+	
ЕЯ-взаимодействие			+	

*Таблица 1. Пример классификации агентов*

Как следует из приведенной таблицы, собственно целесообразное поведение появляется только на уровне интеллектуальных агентов. Для него необходимо не только наличие целей функционирования, но и возможность использования достаточно сложных знаний о среде, партнерах и о себе.

Свойства агента (объекта) описываются исходной системой, а правила поведения – порождающей системой. Состояние объекта определяется перечнем его свойств с текущими значениями.

При создании информационных систем определяющей является парадигма интеллектуальных агентов, поведение которых определяется базой знаний (БЗ). В настоящее время проводятся исследования по расширению возможностей агента в целях обеспечения кооперативного действия многих агентов.

Уровень интеллектуальности (уровень интеллекта) определенного агента можно оценить как способность агента использовать старые знания в новых, может быть, заранее неизвестных ему ситуациях и проблемных областях, где оцениваемый агент приемлем как активный решатель задач.

Как правило, каждый агент (модуль) работает с определенной метафорой, определяющей функции и особенности исполнителя (человека). Идея каждой метафоры – это инструментальная система (прототип), отражающая схему взаимодействия между исполнителями в данной ситуации.

Под интеллектуальным агентом в информатике и искусственном интеллекте понимаются любые физические или виртуальные единицы, способные, по крайней мере, поддерживать взаимодействие с окружающим миром, получая от него информацию, и, реагируя на нее своими действиями, проявлять собственную инициативу, посылать и получать сообщения от других агентов и вступать с ними во взаимодействие, действовать без вмешательства извне, в том числе и без вмешательства человека.

Интеллектуальные агенты, обладая развитым внутренним представлением внешней среды и возможностями рассуждений, способны запоминать и анализировать различные ситуации, предвидеть возможные реакции на свои действия, делать из этого выводы, полезные для дальнейших действий, и в результате прогнозировать свое будущее и изменения внешней среды. Интеллектуальные способности позволяют таким агентам строить виртуальные миры, работая в которых, они формируют планы действий.

## *Свойства интеллектуального агента*

Интеллектуальный агент должен обладать следующими свойствами:

- автономность – способность функционировать без вмешательства со стороны своего владельца и осуществлять контроль внутреннего состояния и своих действий;
- социальное поведение – возможность взаимодействия и коммуникации с другими агентами;
- реактивность – адекватное восприятие среды и соответствующие реакции на ее изменения;
- активность – способность генерировать цели и действовать рациональным образом для их достижения;
- базовые знания – знания агента о себе, окружающей среде, включая других агентов, которые не меняются в рамках жизненного цикла агента;
- убеждения – переменная часть базовых знаний, которые могут меняться во времени, хотя агент может об этом не знать и продолжать их использовать для своих целей;
- цели – совокупность состояний, на достижение которых направлено текущее поведение агента;
- желания – состояния и/или ситуации, достижение которых для агента важно;
- обязательства – задачи, которые берет на себя агент по просьбе и/или поручению других агентов;
- намерения – то, что агент должен делать в силу своих обязательств и/или желаний.

Иногда в этот же перечень добавляются и такие свойства, как рациональность, правдивость, благожелательность, а также мобильность, хотя последнее характерно не только для интеллектуальных агентов.

## *Классы архитектур*

В зависимости от концепции, выбранной для организации МАС, обычно выделяются три базовых класса архитектур:

1. архитектуры, которые базируются на принципах и методах работы со знаниями;
2. архитектуры, основанные на поведенческих моделях типа «стимул-реакция»;
3. гибридные архитектуры.

Архитектуры многоагентных интеллектуальных систем и их характеристики, широко используемые в настоящее время, представлены в Таблице 2.

<b>Архитектура</b>	<b>Представление знаний</b>	<b>Модель мира</b>	<b>Решатель</b>
<b>Интеллектуальная</b>	Символьное	Исчисление	Логический
<b>Реактивная</b>	Автоматное	Граф	Автомат
<b>Гибридная</b>	Смешанное	Гибридная	Машина вывода

*Таблица 2. Архитектуры МАС и их характеристики*

Организация МАС на принципах искусственного интеллекта имеет преимущества с точки зрения удобства использования методов и средств символического представления знаний, разработанных в рамках искусственного интеллекта. Но в то же время создание точной и полной модели представления мира, процессов и механизмов рассуждения в нем представляют здесь существенные трудности.

Гибридные архитектуры позволяют гибко комбинировать возможности всех подходов. Вот почему в последнее время явно прослеживается тенденция разработки и использования именно гибридных МАС–архитектур и систем агентов.

## *Концепции использования технологии многоагентных интеллектуальных систем*

При исследовании сложных систем с использованием технологии МАС реализуются следующие фундаментальные идеи:

1. **Агенты.** В сложных системах существуют автономные агенты, которые взаимодействуют друг с другом при выполнении своих определенных задач.
2. Агенты изменяют свое поведение на основе полученной информации и реагируют на свою среду.
3. Структура сложных систем формируется в результате взаимодействия между агентами. Результаты функционирования возникающей структуры могут быть как положительными, так и отрицательными, в силу чего их необходимо анализировать при разработке системы на базе агентов.
4. Системы с возникающими структурами часто существуют на грани порядка и хаоса. Для них характерно промежуточное состояние между упорядоченным состоянием и хаосом.
5. При создании систем на базе агентов необходимо учитывать их паразитизм, симбиоз, репродукцию, генетику, естественный отбор, т.е. подходы, которые сложились у природы при решении комбинаторных задач.

Агенты могут работать как неинтерактивные особи или как коллектив. В первом случае система очень проста: агенты делают то, о чем их просят. Для получения результата большего, чем сумма работ, выполняемых индивидуально отдельными агентами, необходимо их взаимодействие.

При взаимодействии агентов создаются логически связанные структуры, отражая их динамику. В данном случае реализуются следующие свойства.

1. В возникающих структурах агенты организуются в целое, которое больше, чем просто сумма составляющих.
2. Простые правила могут генерировать логически связанное новое явление.
3. При проектировании систем на базе агентов для слабопрогнозируемых ситуаций наиболее применим подход снизу вверх, а не традиционный подход сверху вниз. Такой подход объясняется тем, что большинство систем возникают из популяции более простых систем.
4. Устойчивые возникающие структуры могут стать компонентами более сложных возникающих структур и приводят к иерархии возникающих структур и их масштабируемости.
5. Агенты и их взаимодействующие структуры могут формировать двунаправленную связь, т.е. взаимное влияние структуры на агентов.
6. Явление возникновения – это, как правило, устойчивые связи с меняющимися компонентами.
7. Множество агентов могут быть гомогентными или гетерогентными. Большинство систем создается на основе гетерогентности используя действие различных видов агентов.

В настоящее время многоагентные системы рассматриваются уже не только как принципиально новая информационная технология, сформированная на базе слияния информационных и телекоммуникационных технологий, но и как новая парадигма программирования, в какой-то мере альтернативная объектно-ориентированному программированию.

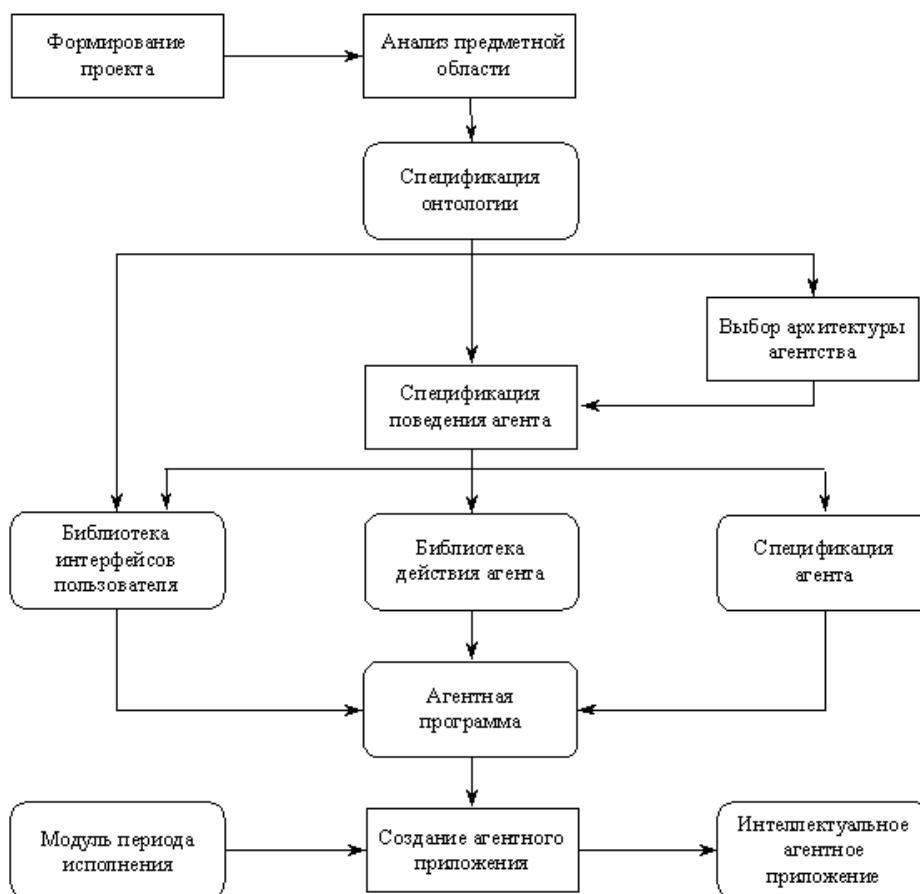
### *Технология программной реализации интеллектуальных многоагентных систем*

Для построения МАС необходим инструментарий, состоящий из двух компонентов;

- средств разработки;

- окружения периода исполнения.

Первый компонент ориентирован на поддержку процессов анализа предметной области, создаваемой МАС, и проектирование агентов с заданным поведением. Второй – обеспечивает эффективную среду для выполнения агентно–ориентированных программ.



*Рис. 7. Технологическая схема процесса разработки агентно-ориентированных приложений*

Общая схема проектирования и реализации агентно-ориентированных приложений представлена на рис.7.

Этот инструментарий имеет средства для организации предметной области, создаваемой МАС, средства спецификации архитектуры агентства и поведения агентов, а также средства отладки агентных приложений и наблюдения за поведением созданных агентов.



Процесс обработки информации агентом («жизненный цикл» агента) включает следующие основные шаги:

- обработка новых сообщений;
- определение, какие правила поведения применимы в текущей ситуации;
- выполнение действий, специфицированных этими правилами;
- обновление ментальной модели в соответствии с заданными правилами;
- планирование.

В рамках данной модели правила поведения фиксируют множество возможных откликов агента на текущее состояние среды, что предписывается соответствующими гипотезами.

Правила поведения для данной модели описываются конструкциями вида «когда – если – то».

«Когда» часть правила адресована новым событиям, возникающим в окружении агента, и включает новые сообщения, полученные от других агентов.

«Если» часть сравнивает текущую ментальную модель с условиями применимости правила. Образцы в «Если» – части работают на намерениях, гипотезах, обязательствах и возможностях, определенной в ментальной модели.

«То» – часть определяет действия в ответ на текущие события и состояния ментальной модели и внешнего окружения. Они могут включать обновление ментальной модели, коммуникативные и внутренние действия.

### ***Направления в разработке многоагентных систем***

Разработка многоагентных систем идет по двум основным направлениям: системы на основе распределенного искусственного интеллекта и системы на основе сценариев.

Многоагентные системы на основе распределенного искусственного интеллекта строятся на базе систем, основанных на правилах, и рассуждениях на основе прецедентов. В них каждый агент рассматривается как система (нераспределенная), основанная на знаниях с добавлением компонентов, обеспечивающих безопасность, мобильность, качество обслуживания, взаимодействие с другими агентами, сетевыми ресурсами и пользователями. Данные МАС характеризуются скоординированным интеллектуальным поведением в сообществе интеллектуальных агентов. МАС на основе распределенного искусственного интеллекта рассматриваются как наиболее многообещающие МАС. Исследования в этом направлении в настоящее время идут по пути интенсивных теоретических исследований и прикладных разработок.

Многоагентные системы на основе сценариев ориентированы на использование в больших компьютерных сетях с большим разнообразием предоставляемого сервиса. Агенты данного класса систем разрабатываются с помощью языков сценариев, таких как Tcl, Java Language и др. Эти языки ориентированы на реализацию асинхронного процесса и удаленное исполнение приложений, инициируемых электронной почтой. Разрабатываемые здесь технологии должны поддерживать мобильность агента. Нужно отметить, что МАС на основе сценариев рассматриваются как основной подход для реализации мобильной телекоммуникации через компьютерные сети общего пользования с помощью переносных компьютеров.

По сути дела многоагентные системы можно рассматривать как совокупность взаимосвязанных программных модулей (агентов), являющихся фрагментами знаний, доступных другим агентам. Это, можно сказать, некие «программные роботы», удовлетворяющие различным информационным и вычислительным потребностям конечных пользователей. Они координируют свои знания, цели, умения и планы при решении

проблем. Таким образом, можно рассматривать МАС как некие программно-исполнительные устройства.

Методология ситуационного управления использована при создании модели интеллектуальной системы поддержки принятия решения (ИСППР), которая функционирует в двух контурах – контуре организации управления (адаптация структуры системы) и контуре функционирования. При создании ИСППР реализованы принципы целевой направленности системы, оцениваемости (распознаваемости) ситуаций, обоснования решений, открытости системы, интегрального представления видов информации, адаптивности. ИСППР включает инструментальный и прикладной программные комплексы.

Предложенная модель конструирования организации управления включает в себе процесс создания конкретной (прикладной) ИСППР на основе базовой (инструментальной) системы. Областью приложений разрабатываемого проекта выбран класс задач оценки и прогнозирования чрезвычайных ситуаций и оперативного реагирования – класс задач конфликтных ситуаций.

С учетом специфики предметной области сформулированы требования к модели представления знаний по системе понятий, адекватности содержания и соответствия формулируемых знаний исследуемым процессам и пригодности для выполнения требуемых действий.

## Заключение

Ключевым фактором, определяющим сегодня развитие ИИ-технологий, считается темп роста вычислительной мощности компьютеров, так как принципы работы человеческой психики по-прежнему остаются неясными (на доступном для моделирования уровне детализации). Поэтому тематика ИИ-конференций выглядит достаточно стандартно и по составу почти не меняется уже довольно давно. Но рост производительности современных компьютеров в сочетании с повышением качества алгоритмов периодически делает возможным применение различных научных методов на практике. Так случилось с интеллектуальными игрушками, так происходит с домашними роботами.

Снова будут интенсивно развиваться временно забытые методы простого перебора вариантов (как в шахматных программах), обходящиеся крайне упрощенным описанием объектов. Но с помощью такого подхода (главный ресурс для его успешного применения - производительность) удастся решить, как ожидается, множество самых разных задач (например, из области криптографии). Уверенно действовать автономным устройствам в сложном мире помогут достаточно простые, но ресурсоемкие алгоритмы адаптивного поведения. При этом ставится цель разрабатывать системы, не внешне похожие на человека, а действующие, как человек.

Ученые пытаются заглянуть и в более отдаленное будущее. Можно ли создать автономные устройства, способные при необходимости самостоятельно собирать себе подобные копии (размножаться)? Способна ли наука создать соответствующие алгоритмы? Сможем ли мы контролировать такие машины? Ответов на эти вопросы пока нет.

Продолжится активное внедрение формальной логики в прикладные системы представления и обработки знаний. В то же время такая логика не способна полноценно отразить реальную жизнь, и произойдет интеграция различных систем логического вывода в единых оболочках. При этом,

возможно, удастся перейти от концепции детального представления информации об объектах и приемов манипулирования этой информацией к более абстрактным формальным описаниям и применению универсальных механизмов вывода, а сами объекты будут характеризоваться небольшим массивом данных, основанных на вероятностных распределениях характеристик.

Сфера ИИ, ставшая зрелой наукой, развивается постепенно - медленно, но неуклонно продвигаясь вперед. Поэтому результаты достаточно хорошо прогнозируемы, хотя на этом пути не исключены и внезапные прорывы, связанные со стратегическими инициативами. Например, в 80-х годах национальная компьютерная инициатива США вывела немало направлений ИИ из лабораторий и оказала существенное влияние на развитие теории высокопроизводительных вычислений и ее применение во множестве прикладных проектов. Такие инициативы будут появляться скорее всего на стыках разных математических дисциплин – теории вероятности, нейронных сетей, нечеткой логики, многоагентных систем.

## Список использованной литературы

1. Люгер, Джордж, Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание. : Пер. с англ. - М.: Издательский дом "Вильямс", 2003. с.777-807.
2. Сотник С. Л. Конспект лекций по курсу " Проектирование систем искусственного интеллекта ", 1997-1998, [www.intuit.ru/department/expert/artintell/](http://www.intuit.ru/department/expert/artintell/)
3. Э. В. Попов, Экспертные системы реального времени, 1995, [www.osp.ru/os/1995/02/178608/](http://www.osp.ru/os/1995/02/178608/)
4. [www.itfru.ru/index.php/expert-systems](http://www.itfru.ru/index.php/expert-systems)
5. Чекинов Г.П., Чекинов С.Г. Применение технологии многоагентных систем для интеллектуальной поддержки принятия решения, сетевой электронный журнал «Системотехника», №1, 2003 год, [www.systech.miem.edu.ru](http://www.systech.miem.edu.ru)
6. Трахтенгеру Э.А. Компьютерная поддержка принятия решений. – М.: Наука, 1998.
7. Поспелов Д.А. Многоагентные системы – настоящее и будущее, Информационные технологии и вычислительные системы, № 1, 1998, с.14-21.
8. Евгеньев Г.Б. Мультиагентные системы компьютерной инженерной деятельности, Информационные технологии, № 4, 2000, с.2-7.
9. Чекинов С.Г. Интеллектуальные программные исполнительные устройства (агенты) в системах связи, Информационные технологии, № 4, 2001, с.6-11.
10. Чекинов Г.П., Куляница А.Л., Бондаренко В.В. Применение ситуационного управления в информационной поддержке принятия решений при проектировании организационно-технических систем, Информационные технологии в проектировании и производстве, № 2, 2003.
11. Перспективы и тенденции развития искусственного интеллекта, PC Week/RE №32, 2001 г., стр. 32, [www.ai.obrazec.ru/ai\\_perspective1.htm](http://www.ai.obrazec.ru/ai_perspective1.htm)
12. Рассел С., Норвиг П. Искусственный интеллект: современный подход Artificial Intelligence: a Modern Approach Пер. с англ. и ред. К. А. Птицына. — 2-е изд. — М.: Вильямс, 2006.
13. Осипов Г. В. Искусственный интеллект: состояние исследований и взгляд в будущее, [www.raai.org/about/persons/osipov/pages/ai/ai.html](http://www.raai.org/about/persons/osipov/pages/ai/ai.html)
14. Портал искусственного интеллекта, [www.aiportal.ru](http://www.aiportal.ru)
15. Википедия, [www.wikipedia.org](http://www.wikipedia.org)