

**А. В. ОСТРОУХ,
А. Б. НИКОЛАЕВ**

ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

МОНОГРАФИЯ

Издание второе, стереотипное



ЛАНЬ

**САНКТ-ПЕТЕРБУРГ • МОСКВА • КРАСНОДАР
2021**

УДК 004.42

ББК 32.973.26я73

О 79 **Остроух А. В.** Интеллектуальные информационные системы и технологии : монография / А. В. Остроух, А. Б. Николаев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 308 с. — Текст : непосредственный.

ISBN 978-5-8114-8578-9

В монографии изложены концептуальные основы и методы представления знаний в интеллектуальных системах. Рассмотрены различные подходы, применяемые при проектировании и разработке интеллектуальных систем и технологий в транспортном комплексе, а также тенденции развития систем искусственного интеллекта.

Монография может быть использована для формирования профессиональной компетентности студентов высших учебных заведений, аспирантов и научных сотрудников, обучающихся и ведущих научные исследования в области разработки и практического применения систем искусственного интеллекта по укрупнённой группе направлений подготовки «Информатика и вычислительная техника».

УДК 004.42

ББК 32.973.26я73

Обложка
Ю. В. ГРИГОРЬЕВА

© Издательство «Лань», 2021
© А. В. Остроух, А. Б. Николаев, 2021
© Издательство «Лань»,
художественное оформление, 2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	8
1. СОСТАВ И СТРУКТУРА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	13
1.1. Роль интеллектуальных систем в процессе решения трудноформализуемых задач.....	13
1.2. Задачи предметной области и методы их решения	16
1.3. Основные принципы организации интеллектуальных систем.....	19
1.4. Модели представления знаний	24
1.4.1. Неформальные (семантические) модели	25
1.4.2. Формальные модели представления знаний	30
1.5. Архитектура интеллектуальных систем	32
1.6. Перспективы интеллектуализации информационных систем	33
Контрольные вопросы.....	41
2. ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ	43
2.1. Понятие интеллектуальной информационной технологии.....	43
2.2. Классификация интеллектуальных информационных систем	48
2.3. Перспективные информационные технологии проектирования, создания, анализа и сопровождения интеллектуальных систем	50
2.3.1. Системы с интеллектуальным интерфейсом	50
2.3.2. Экспертные системы	52

2.3.3. Самообучающиеся системы	59
2.3.4. Адаптивные информационные системы.....	63
Контрольные вопросы.....	64
3. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ В ОБЛАСТИ ПРИМЕНЕНИЯ И УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫМИ СИСТЕМАМИ	66
3.1. Процессы по развитию функциональных возможностей интеллектуальных систем на всех стадиях их жизненного цикла	66
3.1.1. Инструментальные средства разработки MAC	67
3.1.2. AgSDK: среда разработки и реализации MAC	70
3.2. Основные тенденции развития интеллектуальных систем, связанных с изменениями условий в области применения	79
3.3. Обеспечение информационной безопасности в интеллектуальных системах	91
3.3.1. Цели обеспечения информационной безопасности.....	91
3.3.2. Задачи системы информационной безопасности.....	92
3.3.3. Принципы обеспечения информационной безопасности.....	93
3.4. Управляемость интеллектуальных систем	96
3.5. Задачи управления интеллектуальными системами.....	99
3.5.1. Управление информационными ресурсами	100
3.5.2. Управление решателями задач.....	116
3.5.3. Управление пользовательским интерфейсом.....	123

3.6. Методы научных исследований по теории, технологии разработки и эксплуатации интеллектуальных систем.....	127
Контрольные вопросы.....	130
4. ПРИМЕНЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	
В ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ	131
4.1. Организация диалога между человеком и интеллектуальной системой.....	131
4.1.1. Диалоговые системы, основанные на распознавании рукописного текста	131
4.1.2. Диалоговые системы, основанные на распознавании речи	137
4.1.3. Системы с биологической обратной связью	140
4.1.4. Системы с семантическим резонансом. Компьютерные Ψ-технологии и интеллектуальный подсознательный интерфейс.....	144
4.1.5. Системы виртуальной реальности. Эффекты присутствия, деперсонализации и модификация сознания пользователя	147
4.1.6. Системы с дистанционным телекинетическим интерфейсом.....	157
4.2. Построение сложных предметно-ориентированных интеллектуальных систем на основе естественно-языкового интерфейса	158
4.2.1. Сравнительный анализ ЕЯ-интерфейсов и традиционных интерфейсов к структурированным источникам данных (СИД).....	158
4.2.2. Критерии качества ЕЯ-интерфейсов.....	164
4.2.3. Критерии стоимости построения и сопровождения ЕЯ-интерфейса	165

4.2.4. Вопросы портируемости.....	167
4.2.5. Основные составные части ЕЯ-интерфейсов	169
4.3. Создание и внедрение технических и экономических проектов при помощи современных интеллектуальных систем	172
4.3.1. Проектирование: принципы и методы создания ИС	172
4.3.2. Этапы создания информационных систем (ИС)	182
4.3.3. Реинжиниринг бизнес-процессов с помощью системы ReThink.....	185
4.3.4. Интеллектуальный анализ данных	192
4.3.5. Управление решателями задач.....	194
4.4. Работы с основными объектами, процессами и явлениями, связанными с интеллектуальными системами, и использование методов их научного исследования	195
4.4.1. Структура систем интеллектуального управления.....	196
4.4.2. Модели принятия решения в условиях конфликта.....	200
4.4.3. Определение оптимальной интеллектуальной системы принятия решения и управления в условиях конфликта	207
Видеоролики по главе.....	221
Контрольные вопросы.....	221
5. ОСОБЕННОСТИ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ЭКСПЕРТНЫХ СИСТЕМ	223
5.1. Назначение, классификация и принципы построения экспертных систем	223

5.1.1. Назначение экспертных систем.....	223
5.1.2. Классификация экспертных систем	228
5.1.3. Структура экспертных систем.....	231
5.2. Методология разработки экспертных систем	237
5.3. Этапы разработки экспертных систем.....	240
5.4. Взаимодействие инженера по знаниям (когнитолога) с экспертом	246
5.4.1. Представление знаний в экспертных системах	246
5.4.2. Уровни представления и уровни детальности	251
5.4.3. Организация знаний в рабочей системе.....	252
5.4.4. Организация знаний в базе данных	253
5.5. Методы поиска решений в экспертных системах....	257
5.6. Инструментальные средства проектирования и разработки экспертных систем.....	259
5.7. Трудности разработки экспертных систем	267
5.8. Перспективы развития экспертных систем	270
Видеоролики по главе.....	285
Контрольные вопросы.....	285
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	287

ВВЕДЕНИЕ

Искусственный интеллект (ИИ) — раздел информатики, изучающий принципы действия интеллектуальных машин.

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



Герберт Саймон (слева) и Аллен Ньюэлл (справа)

Эвристический метод решения задачи при этом рассматривался как свойственный человеческому мышлению "вообще", для которого характерно возникновение догадок о пути решения

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

На дальнейшие исследования в области ИИ большое влияние оказало появление метода резолюций Робинсона, основанного на доказательстве теорем в логике предикатов и являющегося исчерпывающим методом доказательства. При этом

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

Исследовательским полигоном для развития методов ИИ на первом этапе явились всевозможные игры, головоломки, математические задачи. Некоторые из этих задач стали классическими в литературе по ИИ (задачи об обезьяне и бананах, миссионерах и людоедах, Ханойской башне, игра в 15 и др.). Выбор таких задач обуславливался простотой и ясностью проблемной среды (среды, в которой разворачивается решение задачи), ее относительно малой громоздкостью, возможностью достаточно легкого подбора и даже искусственного конструирования "под метод". Основной расцвет такого рода исследований приходится на конец 60-х годов XX века, после чего стали делаться первые попытки применения разработанных методов для задач, решаемых не в искусственных, а в реальных проблемных средах. Необходимость исследования систем ИИ при их функцио-

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

К этому времени выяснилось, что даже простые на первый взгляд задачи, возникающие перед интегральным роботом при его функционировании в реальном времени, не могут быть решены методами, разработанными для экспериментальных задач в специально сформированных проблемных средах.

Стало ясно, что сочетание дополняющих друг друга возможностей человека и ЭВМ позволяет обойти острые углы путем перекадывания на человека тех функций, которые пока еще не доступны для ЭВМ. На первый план выдвигалась не разработка отдельных методов машинного решения задач, а разработка методов средств, обеспечивающих тесное взаимодействие человека и вычислительной системы в течение всего процесса решения задачи с возможностью оперативного внесения человеком изменений в ходе этого процесса.

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

1. СОСТАВ И СТРУКТУРА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

1.1. РОЛЬ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ В ПРОЦЕССЕ РЕШЕНИЯ ТРУДНОФОРМАЛИЗУЕМЫХ ЗАДАЧ

Интеллектуальная система (ИС) — это компьютерная модель интеллектуальных возможностей человека в целенаправленном поиске, анализе и синтезе текущей информации об окружающей действительности для получения о ней новых знаний и решения на этой основе различных жизненно важных задач.

Исследователи, работающие в этом направлении, надеются достичь такого понимания механизмов интеллекта, при котором можно будет составлять компьютерные программы с человеческим или более высоким уровнем интеллекта. Общий подход состоит в разработке методов решения задач, для которых отсутствуют формальные алгоритмы: понимание естественного языка, обучение, доказательство теорем, распознавание сложных образов и т. д. Теоретические исследования направлены на изучение интеллектуальных процессов и создание соответствующих математических моделей. Экспериментальные работы ведутся путем составления компьютерных программ и создания машин, решающих частные интеллектуальные задачи или разумно ведущих себя в заданной ситуации. Систематические исследования в области искусственного интеллекта начались лишь с появлением цифрового компьютера. Первая научная статья по искусственному интеллекту была опубликована в 1950 году А. Тьюрингом. Развитие интеллектуальных систем на современном этапе идет в соответствии с тремя направлениями исследований.

Первое направление объектом исследований рассматривает структуру и механизмы работы мозга человека, а конечной целью — раскрытие тайн мышления. Необходимым этапом исследований в этом направлении является построение моделей интеллектуальной деятельности на основе психофизиологических данных.

Второе направление в качестве объекта исследования рассматривает искусственную интеллектуальную систему. Здесь речь идет о моделировании интеллектуальной деятельности с помощью вычислительных машин. Целью работ в этом направлении является создание программного обеспечения, позволяющего решать некоторые виды интеллектуальных задач так же, как их решил бы человек.

Третье направление ориентировано на создание человеко-машинных или, как еще говорят, интерактивных, интеллектуальных систем. Важнейшей проблемой в этих исследованиях является организация **семантически безупречного диалога между человеком и такой системой**.

Типология ИС на современном этапе их развития представляется двумя крупными категориями.

К первой категории относятся ИС, получившие название **экспертных систем (ЭС)**. Основными компонентами ЭС являются **база знаний** и логическая машина (**универсальный решатель**), которая интерпретирует содержащиеся в базе знания. Главной особенностью этой категории ИС является то, что они основаны на применении к содержащимся в базе знаниям определенных наборов **правил** (содержащихся в решателе) и на соблюдении определенных **последовательностей применения этих правил**.

Ко второй категории относятся ИС, получившие название **искусственных нейронных сетей**. Искусственная нейронная сеть (ИНС) представляет собой структуру, состоящую из нескольких слоев электронных моделей нервных клеток человеческого мозга — нейронов: одного входного слоя, нескольких внутренних (скрытых) слоев и одного выходного слоя. Поток подлежащей обработке (распознаванию) информации поступает на входной слой, проходит через внутренние слои, и результаты обработки информации выдаются через выходной слой искусственных нейронов. Во внутренних слоях устанавливаются связи между входным и выходным сигналами нейронной сети. Изменчивость связей между входом и выходом такой сети обеспечивается за счет различия порогов чувствительности входного и выходного слоев, которые устанавливаются и корректируются в процессе обучения сети. Наиболее популярен метод, основанный на корректировке ее структуры с учетом результатов пробных решений. Суть его в том, что предварительно готовится достаточно представительная обучающая выборка — множество пар входных и выходных сигналов. Затем входные данные обучающей выборки последовательно вводятся в сеть для получения выходных данных сети, которые потом сравниваются с выходными данными обучающей выборки. Если они совпадают, то сеть считается обученной и никакой корректировки связей внутри сети не производится. В противном случае эти связи корректируются и процесс обучения повторяется до тех пор, пока не будет достигнута необходимая точность совпадения выходных данных сети с выходными данными обучающей выборки. Главной особенностью этих ИС является то, что они ориентированы на использование **примеров (преце-**

дентов) или образцов приемлемого исполнения целевой функции. При проектировании ИНС нет необходимости формализовать процесс решения задачи. Необходимо лишь подготовить достаточно представительную выборку обучающих примеров и провести на ее основе обучение системы.

Качество ИС определяется следующими аспектами:

- **коммуникативность**, трактуемая как многообразие доступных всем категориям пользователей способов общения с системой;
- **универсальность** по отношению к множеству задач, в пределах которого должна функционировать система;
- **умение обучаться** на основе приобретаемого опыта и знаний, приспособливаясь к изменению условий решения проблемы;
- **умение перестроиться** при изменении принципиальных положений (концепций) предметной и проблемной областей.

1.2. ЗАДАЧИ ПРЕДМЕТНОЙ ОБЛАСТИ И МЕТОДЫ ИХ РЕШЕНИЯ

Обобщенный перечень классов задач разработки ИС для практического применения выглядит следующим образом:

- разработка средств, обеспечивающих эффективное общение человека с машиной (автоматом) на естественном языке;
- автоматический перевод текстовой информации с естественного языка на внутримашинный и обратно;
- автоматический концептуальный анализ, поиск и интерпретация данных и знаний;

- разработка алгоритмов и методов поддержки принятия решений по целесообразному управлению различными системами, объектами и процессами в экономике, финансах, производстве, науке, образовании, медицине, экологии, природопользовании, вооружении и других отраслях хозяйства с учетом неопределенности в реализации факторов внешней среды и связанных с этим рисков;
- разработка алгоритмов и методов мониторинга и диагностики состояния систем, объектов и процессов;
- автоматическое проектирование систем и устройств с оптимальными свойствами;
- разработка алгоритмов логических выводов и доказательства теорем;
- разработка поведенческих алгоритмов в условиях неопределенности и риска;
- автоматическое распознавание образов различной природы;
- создание автоматически обучающихся систем.

В рамках каждого из трех направлений развития ИС существуют различные **подходы к построению ИС**.

Логический подход. Основой для данного подхода является применение различных правил логического вывода, изложенных с помощью аппарата математической логики (булевой алгебры и исчисления предикатов). ИС, основанная на логическом подходе, представляет собой машину для решения задач логических выводов и доказательства теорем. При этом исходные данные хранятся в базе знаний в виде аксиом и правил построения логического вывода путём задания отношений между

этими данными. К сожалению, выразительность математической логики недостаточна для реализации полных возможностей логического мышления человека, но зато неоспоримо, что **все, что можно реализовать на ЭВМ, принципиально реализуемо с помощью логики предикатов, и наоборот**. Для логических методов характерна большая трудоемкость, поскольку поиск доказательства может потребовать полного перебора всех возможных вариантов решений. Поэтому данный подход требует эффективной реализации вычислительного процесса и хорошо "работает" при сравнительно небольшом объеме базы знаний.

Физический подход объединяет методы моделирования интеллектуальных возможностей человека с помощью компьютера и различных физических устройств. Одной из первых таких попыток был перцептрон Фрэнка Розенблатта. Структурной единицей перцептрона (как и большинства других вариантов такого моделирования) является компьютерная модель нейрона — нервной клетки. Из них строятся "искусственные нейронные сети" (ИНС). Нейронные сети в настоящее время есть коммерческие аппаратно-программные продукты, представляющие собой законченные ИС, применяемые в различных областях деятельности. Их действие основано на перестройке параметров нейронной сети под воздействием предъявляемых примеров правильного решения задачи.

Широкое распространение получило **эволюционное моделирование**. Этот процесс начинается с создания начального варианта модели ИС для решения конкретной интеллектуальной задачи и набора правил, по которым эта модель может изменяться. Затем формулируется критерий правильного (или

приемлемого) решения этой задачи. Начальный вариант модели ИС обычно не отвечает установленному критерию качества решения задачи. С этого момента начинается пошаговое улучшение начального варианта модели ИС. Оно заключается в следующем. Описание начальной модели, правила вычисления критерия оценки её качества и правила (стратегии) её изменения вводятся в компьютер, который "проигрывает" работу начальной модели по каждой из возможных стратегий её развития и отбирает наилучший по установленному критерию результат эволюции в качестве новой начальной модели, которая вновь опробуется на допустимых правилами стратегиях работы. Такие циклы эволюции продолжаются, пока модель не придёт к требуемому уровню качества работы.

Имитационное моделирование. Оно связано с классическим для кибернетики базовым понятием — "черным ящиком" (ЧЯ). Так называют устройство, информация о внутренней структуре и содержании которого отсутствует полностью, но известна матрица обязательного соответствия сигналов на входе в него и сигналов на его выходе. Задача состоит в подборе правил соответствия сигналов на входе и выходе.

1.3. ОСНОВНЫЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Информация, с которой имеют дело ЭВМ, разделяется на *процедурную* и *декларативную*. Процедурная информация о веществе в *программах*, которые выполняются в процессе решения задач, декларативная информация — в *данных*, с которыми эти программы работают.

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

В ЭВМ *знания* так же, как и *данные*, отображаются в знаковой форме — в виде формул, текста, файлов, информационных массивов и т. п. Поэтому можно сказать, что знания — это особым образом организованные данные. Но это было бы слишком узкое понимание. А между тем в системах ИИ знания являются основным объектом формирования, обработки и исследования.

База знаний (БЗ; *англ.* knowledge base, KB) в информатике и исследованиях искусственного интеллекта — это особого рода база данных, разработанная для оперирования знаниями (метаданными). База знаний содержит структурированную информацию, покрывающую некоторую область знаний, для использования кибернетическим устройством (или человеком) с конкретной целью. Современные базы знаний работают совместно с системами поиска информации, имеют классификационную структуру и формат представления знаний.

Полноценные базы знаний содержат в себе не только фактическую информацию, но и правила вывода, допускающие автоматические умозаключения о вновь вводимых фактах и, как следствие, осмысленную обработку информации. Область наук об искусственном интеллекте, изучающая базы знаний и методы работы со знаниями, называется инженерией знаний.

Перечисленные ниже пять особенностей информационных единиц определяют ту грань, за которой данные превращаются в знания, а базы данных перерастают в *базы знаний* (БЗ).

Внутренняя интерпретируемость. Каждая информационная единица должна иметь уникальное имя, по которому ИС находит ее, а также отвечает на запросы, в которых это имя упомянуто. Когда данные, хранящиеся в памяти, были лишены имен, то отсутствовала возможность их идентификации системой. Данные могла идентифицировать лишь программа, извлекающая их из памяти по указанию программиста, написавшего программу. Что скрывается за тем или иным двоичным кодом машинного слова, системе было неизвестно.

Таблица 1.1. Сведения о сотрудниках учреждения

Фамилия	Год рождения	Специальность	Стаж, число лет
Попов	1965	Слесарь	5
Сидоров	1946	Токарь	20
Иванов	1925	Токарь	30
Петров	1937	Сантехник	25

Если, например, в память ЭВМ нужно было записать сведения о сотрудниках учреждения, представленные в таблице 1.1, то без внутренней интерпретации в память ЭВМ была бы занесена совокупность из четырех машинных слов, соответствующих строкам этой таблицы. При этом информация о том, какими группами двоичных разрядов в этих машинных словах закодированы сведения о специалистах, у системы отсутствует.

Они известны лишь программисту, который использует данные таблицы 1.1 для решения возникающих у него задач. Система не в состоянии ответить на вопросы типа "Что тебе известно о Петрове?" или "Есть ли среди специалистов сантехник?".

При переходе к знаниям в память ЭВМ вводится информация о некоторой *протоструктуре информационных единиц*. В рассматриваемом примере она представляет собой специальное машинное слово, в котором указано, в каких разрядах хранятся сведения о фамилиях, годах рождения, специальностях и стаже. При этом должны быть заданы специальные словари, в которых перечислены имеющиеся в памяти системы фамилии, года рождения, специальности и продолжительность стажа. Все эти *атрибуты* могут играть роль имен для тех машинных слов, которые соответствуют строкам таблицы. По ним можно осуществлять поиск нужной информации. Каждая строка таблицы будет экземпляром протоструктуры. В настоящее время СУБД обеспечивают реализацию внутренней интерпретируемости всех информационных единиц, хранящихся в базе данных.

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

Связность. В информационной базе между информационными единицами должна быть предусмотрена возможность установления связей различного типа. Прежде всего эти связи могут характеризовать отношения между информационными

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

Между информационными единицами могут устанавливаться и иные связи, например определяющие порядок выбора информационных единиц из памяти или указывающие на то, что две информационные единицы несовместимы друг с другом в одном описании.

Перечисленные три особенности знаний позволяют ввести общую модель представления знаний, которую можно назвать *семантической сетью*, представляющей собой иерархическую сеть, в вершинах которой находятся информационные единицы. Эти единицы снабжены индивидуальными именами. Дуги семантической сети соответствуют различным связям между информационными единицами.

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

Активность. Все процессы, протекающие в ЭВМ, инициируются командами, а данные используются этими командами лишь в случае необходимости. Для ИС эта ситуация неприемлема. Как и у человека, в ИС актуализации тех или иных действий способствуют знания, имеющиеся в системе. Таким образом, выполнение программ в ИС должно инициироваться текущим состоянием информационной базы. Появление в базе фактов или описаний событий, установление связей могут стать источником активности системы.

Совокупность средств, обеспечивающих работу со знаниями, образует *систему управления базой знаний* (СУБЗ). В настоящее время не существует баз знаний, в которых в полной мере были бы реализованы внутренние интерпретируемость, структуризация, связность, введена семантическая мера и обеспечена активность знаний.

1.4. МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Существуют два типа методов представления знаний (ПЗ):

- формальные модели ПЗ;

- неформальные (семантические, реляционные) модели ПЗ.

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

1.4.1. Неформальные (семантические) модели

Каждому из методов ПЗ соответствует свой способ описания знаний.

1. *Логические модели.* В основе моделей такого типа лежит *формальная система*, задаваемая четверкой вида $M = \langle T, P, A, B \rangle$. Множество T есть *множество базовых элементов* различной природы, например слов из некоторого ограниченного словаря, деталей детского конструктора, входящих в состав некоторого набора, и т. п. Важно, что для множества T существует некоторый способ определения принадлежности или непринадлежности произвольного элемента к этому множеству. Процедура такой проверки может быть любой, но за конечное

число шагов она должна давать положительный или отрицательный ответ на вопрос, является ли x элементом множества T . Обозначим эту процедуру $\Pi(T)$.

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

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

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

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

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

В зависимости от типов связей, используемых в модели, различают *классифицирующие сети*, *функциональные сети* и *сценарии*. В классифицирующих сетях используются отношения структуризации. Такие сети позволяют в базах знаний вводить разные иерархические отношения между информационными единицами. Функциональные сети характеризуются наличием функциональных отношений. Их часто называют *вычислительными моделями*, так как они позволяют описывать процедуры "вычислений" одних информационных единиц через другие. В сценариях используются каузальные отношения, а также отношения типов "средство — результат", "орудие — действие"

и т. п. Если в сетевой модели допускаются связи различного типа, то ее обычно называют семантической сетью.

3. *Продукционные модели.* В моделях этого типа используются некоторые элементы логических и сетевых моделей. Из логических моделей заимствована идея правил вывода, которые здесь называются *продукциями*, а из сетевых моделей — описание знаний в виде семантической сети. В результате применения правил вывода к фрагментам сетевого описания происходит трансформация семантической сети за счет смены ее фрагментов, наращивания сети и исключения из нее ненужных фрагментов. Таким образом, в продукционных моделях процедурная информация явно выделена и описывается иными средствами, чем декларативная информация. Вместо логического вывода, характерного для логических моделей, в продукционных моделях появляется *вывод на знаниях*.

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

(Имя фрейма:

Имя слота 1 (значение слота 1);

Имя слота 2 (значение слота 2);

.....

Имя слота K (значение слота K)).

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

уровня, что позволяет во фреймовых представлениях реализовать "принцип матрешки".

При конкретизации фрейма ему и слотам присваиваются конкретные имена и происходит заполнение слотов. Таким образом, из протофреймов получаются *фреймы* — *экземпляры*. Переход от исходного протофрейма к фрейму — экземпляру — может быть многоступенчатым за счет постепенного уточнения значений слотов.

Например, структура таблицы 1.1, записанная в виде протофрейма, имеет вид:

(Список работников:

Фамилия (значение слота 1);

Год рождения (значение слота 2);

Специальность (значение слота 3);

Стаж (значение слота 4)).

Если в качестве значений слотов использовать данные таблицы 1.1, то получится фрейм-экземпляр:

(Список работников:

Фамилия (Попов — Сидоров — Иванов — Петров);

Год рождения (1965 — 1946 — 1925 — 1937);

Специальность (слесарь — токарь — токарь — сантехник);

Стаж (5 — 20 — 30 — 25)).

Связи между фреймами задаются значениями специального слота с именем "Связь". Часть специалистов по ИС считает, что нет необходимости специально выделять фреймовые модели в представлении знаний, так как в них объединены все основные особенности моделей остальных типов.

1.4.2. Формальные модели представления знаний

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

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

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

Формальная теория должна, таким образом, удовлетворять следующему определению:

всякая формальная теория $F = (A, V, W, R)$, определяющая некоторую аксиоматическую систему, характеризуется:

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

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

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

1.5. АРХИТЕКТУРА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Архитектура интеллектуальных систем включает три комплекса вычислительных средств (рис. 1.1).

Первый комплекс представляет собой совокупность средств, выполняющих программы (исполнительную систему), спроектированных с позиций эффективного решения задач, имеет в ряде случаев проблемную ориентацию.

Второй комплекс — совокупность средств интеллектуального интерфейса, имеющих гибкую структуру, которая обеспечивает возможность адаптации в широком спектре интересов конечных пользователей.

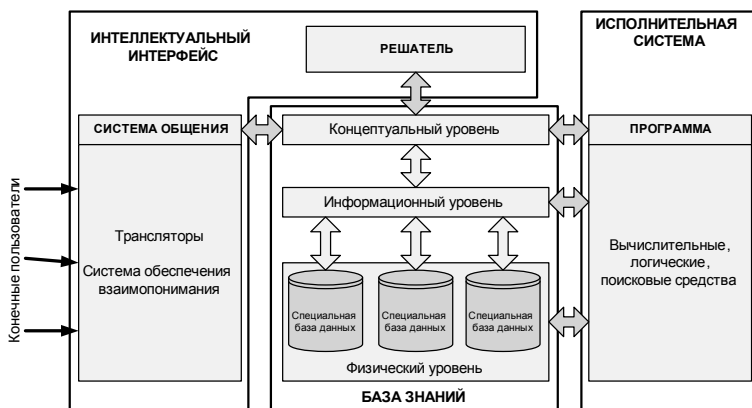


Рис. 1.1. Архитектура интеллектуальных систем

Третьим комплексом средств, с помощью которых организуется взаимодействие первых двух, является база знаний, обеспечивающая использование вычислительными средствами первых двух комплексов целостной и независимой от обрабатываемых программ системы знаний о проблемной среде. Исполнительная система объединяет всю совокупность средств,

обеспечивающих выполнение сформированной программы. Интеллектуальный интерфейс — система программных и аппаратных средств, обеспечивающих для конечного пользователя использование компьютера для решения задач, которые возникают в среде его профессиональной деятельности либо без посредников, либо с незначительной их помощью. База знаний (БЗ) занимает центральное положение по отношению к остальным компонентам вычислительной системы в целом, через БЗ осуществляется интеграция средств ВС, участвующих в решении задач.

1.6. ПЕРСПЕКТИВЫ ИНТЕЛЛЕКТУАЛИЗАЦИИ ИНФОРМАЦИОННЫХ СИСТЕМ

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. Это становится возможным 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.

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 разрабо-

ток, демонстрировавших пригодность систем искусственного интеллекта для практического использования. Начиная примерно с 1985 года, в первую очередь экспертные системы, а в последние годы и нейронные сети стали активно использоваться в реальных приложениях.

Одно из наиболее популярных направлений последних 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 прагматичный проект, который сейчас ведется под руководством профессора Генри Либермана в Медиа-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 документа-

ции (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.

Генетический алгоритм (*англ.* genetic algorithm) — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, напоминающих биологическую эволюцию. Является разновидностью эволюционных вычислений, с помощью которых решаются оптимизационные задачи с использованием методов естественной эволюции, таких как наследование, мутации, отбор и кроссинговер. Отличительной особенностью генетического алгоритма является акцент на использовании оператора "скрещивания", который производит операцию рекомбинации решений-кандидатов, роль которой аналогична роли скрещивания в живой природе.

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 сис-

темах 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 глаз животного может иметь значение "зеленый" и позицию 10).

В генетических алгоритмах роль основных строительных блоков играют строки фиксированной длины, тогда как в генетическом программировании эти строки разворачиваются в деревья, столь знакомые специалистам в области трансляции.

Ныне одним из лидеров в области генетического программирования является группа исследователей из Стэнфордского университета (Stanford University), работающая под руководством профессора Джона Коза. Генетическое программирование вдохнуло новую жизнь в хорошенько уже подзабытый язык LISP (List Processing), который создавался группой Джона Маккарти (того самого, кто в 60-е годы XX века ввел в 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.

Еще одно перспективное направление развития интеллектуальных систем — разработка **интеллектуальных роботов (ИР, рис. 1.2)**.

Общепринято, что интеллектуальный робот обладает так называемой моделью внешнего мира или внутренней средой, что позволяет ему действовать в условиях неопределенности информации. В том случае, если эта модель реализована в ви-

де базы знаний, то целесообразно, чтобы эта база знаний была динамической. При этом коррекция правил вывода в условиях меняющейся внешней среды естественным образом реализует механизмы самообучения и адаптации.

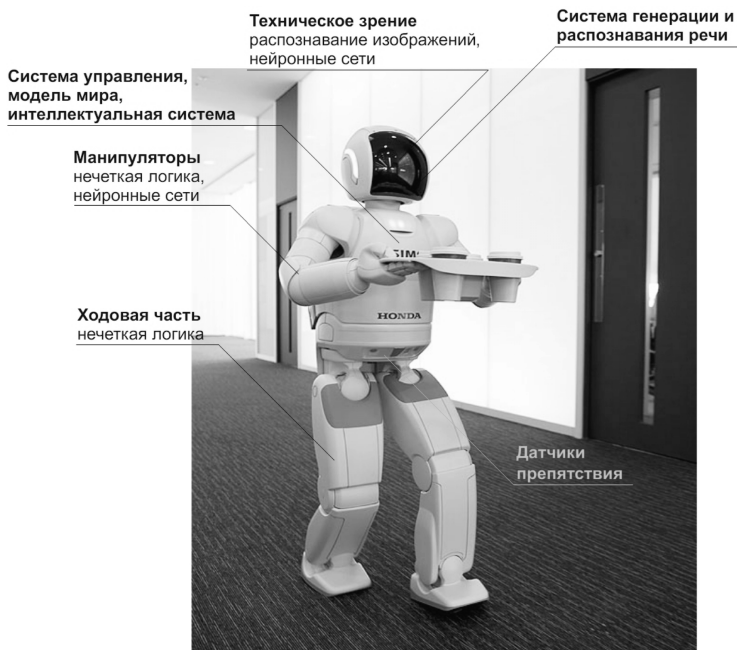


Рис. 1.2. Компоненты интеллектуального робота ASIMO, разработанного компанией HONDA

На сегодняшний день считается, что в состав интеллектуального робота должны входить:

1) **исполнительные органы** — манипуляторы, ходовая часть и другие устройства, с помощью которых робот может воздействовать на окружающие его предметы. Причем по своей структуре это сложные технические устройства, имеющие в

своем составе сервоприводы, мехатронные части, датчики, системы управления. По аналогии с живыми организмами это руки и ноги робота;

2) **датчики** — системы технического зрения, слуха, осязания, датчики расстояний, локаторы и другие устройства, которые позволяют получить информацию из окружающего мира;

3) **система управления** — мозг робота, который должен принимать информацию от датчиков и управлять исполнительными органами. Эта часть робота обычно реализуется программными средствами. В состав системы управления интеллектуального робота должны входить следующие компоненты.

Модель мира, которая отражает состояние окружающего робота мира в терминах, удобных для хранения и обработки. Модель мира выполняет функцию запоминания состояния объектов в мире и их свойств.

Система распознавания, в которую входят системы распознавания изображений, речи и т. п. Задачей системы распознавания является идентификация, т. е. "узнавание" окружающих робота предметов, их положения в пространстве. В результате работы компонентов системы распознавания строится модель мира.

Система планирования действий, которая осуществляет "виртуальное" преобразование модели мира с целью получения какого-либо действия. При этом обычно проверяется достижимость поставленной цели. Результатом работы планирования действий является построение планов, т. е. последовательностей элементарных действий;

4) **система выполнения действий**, которая пытается выполнить запланированные действия, подавая команды на ис-

полнительные устройства и контролируя при этом процесс выполнения. Если выполнение элементарного действия оказывается невозможным, то весь процесс прерывается и должно быть выполнено новое (или частично новое) планирование;

5) **система управления целями**, которая определяет иерархию, т. е. значимость и порядок достижения поставленных целей.

Важным свойством системы управления является способность к обучению и адаптации, т. е. способность генерировать последовательность действий для поставленной цели, а также подстраивать свое поведение под изменяющиеся условия окружающей среды для достижения поставленных целей.

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 интегрируемые 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.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Охарактеризуйте основные направления исследований, проводимых в области искусственного интеллекта.

2. Приведите известные вам примеры интеллектуальных систем.

3. Назовите основные функции, присущие ИС. На чем основана их реализация?

4. Сформулируйте основные отличия систем искусственного интеллекта от обычных программных средств.

5. Чем отличаются знания от данных? Приведите определение знаний.

6. Дайте характеристику основных признаков, по которым классифицируются знания (природа знаний, способ приобретения знаний, тип представления знаний).

7. Расскажите о логических способах представления знаний. Укажите преимущественную область применения логической модели.

2. ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

2.1. ПОНЯТИЕ ИНТЕЛЛЕКТУАЛЬНОЙ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ

Интеллектуальные информационные технологии (ИИТ) (*англ.* Intellectual information technology, IIT) — информационные технологии, помогающие человеку ускорить анализ политической, экономической, социальной и технической ситуации, а также синтез управленческих решений. При этом используемые методы не обязательно должны быть логически непротиворечивы или копировать процессы человеческого мышления.

Использование ИИТ на практике подразумевает учет специфики проблемной области, которая может характеризоваться следующим набором признаков:

- качество и оперативность принятия решений;
- нечеткость целей и институциональных границ;
- множественность субъектов, участвующих в решении проблемы;
- хаотичность, флюктуируемость и квантованность поведения среды;
- множественность взаимовлияющих друг на друга факторов;
- слабая формализуемость, уникальность, нестереотипность ситуаций;
- латентность, скрытость, неявность информации;
- девиантность реализации планов, значимость малых действий;
- парадоксальность логики решений и др.

ИИТ формируются при создании информационных систем и информационных технологий для повышения эффективности принятия решений в условиях, связанных с возникновением проблемных ситуаций. В этом случае любая жизненная или деловая ситуация — от выбора партнера по жизни до социального конфликта — описывается в виде некоторой познавательной модели (когнитивной схемы, архетипа, фрейма и пр.), которая впоследствии используется в качестве основания для построения и проведения моделирования, в том числе компьютерного.

Гносеологический фундамент ИИТ наиболее явно прослеживается в работах Канта, Гегеля, Гуссерля. Собственно же явную историю ИИТ удобно начать с середины XX века, когда появился термин "искусственный интеллект" (Artificial Intelligence). История ИИТ начинается с середины 1970-х годов и связывается с совместным практическим применением интеллектуальных информационных систем, систем искусственного интеллекта, систем поддержки решений и информационных систем. Эта история связана также с развитием трех научных направлений: компьютерной философии, компьютерной психологии и продвинутой компьютерной науки (*англ.* Advanced computer science). С организационно-технологической стороны ИИТ дополняются прогрессом в создании: ситуационных центров, информационно-аналитических систем. Программно-математическое обеспечение составляют эволюционные вычисления и генетические алгоритмы, системы поддержки общения человека с компьютером на естественном языке, когнитивное моделирование, системы автоматического тематического рубрицирования документов, системы стратегического планирования, инструментарий технического и фундаментального анализа финан-

совых рынков, системы менеджмента качества, системы управления интеллектуальной собственностью и др.

С середины 1940-х вплоть до начала 1970-х годов создание ИИТ рассматривалось преимущественно в рамках логического решения задач. Этот период развития ИИТ характеризуется сравнительно большой определенностью и низкой динамичностью объекта управления. Вместе с тем уже в 1943 году появились "продукции Поста" и методы решения некорректных (обратных) задач на метризуемых пространствах, а в 1947 году для моделирования сложных экономических ситуаций активно начали использоваться методы причинного нелогического вывода, которые позже легли в основу методов системной динамики, немонотонных вычислений, когнитивного моделирования. Создание центров управления полетами, организация штабных работ с применением средств визуализации и автоматизации, зарубежные публикации на тему создания специальных ситуационных центров вдохновили в 1970-е годы инженеров на создание ситуационных комнат для совершенствования управления крупными социальными и институциональными системами. В создании таких комнат и интеллектуальных технологий больше внимания стало придаваться средствам визуализации, диалоговым системам, помогающим использовать базы знаний и модели для решения плохо структурированных проблем. В середине 1970-х годов на основе ИИТ в корпоративном мире начинают развиваться системы поддержки решений для эффективного управления ресурсами, осуществления контроллинга. Ряд замечательных практических идей и результатов, например связанных с теорией нейронных сетей, многоагентных и активных систем, оптических и голографических процессоров,

появились именно в это время. Этот период можно отметить успехами в создании всеобъемлющих моделей ситуационного управления регионами в периоды кризисов. Его характеризует вера в практически неограниченные возможности искусственного интеллекта. Середина 1980-х годов характеризуется крахом иллюзий относительно неограниченных возможностей успешной формализации процессов мышления с помощью систем логической обработки естественного языка. Вместе с тем появились интеллектуальные технологии для ограниченной поддержки исследовательской и профессиональной деятельности лиц, принимающих решения. Практическое применение получили подходы, основанные на использовании достоверного и правдоподобного вывода, немонотонных логик и нечетких систем, лингвистических процессоров. Тогда же появилась явная потребность в оптических и квантовых вычислениях — для решения многомерных и слабо распараллеливаемых задач. Видимые успехи появились в сфере обработки текстов естественного языка, высококачественного поиска документов, слежения за динамичными объектами управления, решения задач распознавания образов, имитационного моделирования, статистической обработки данных, решения транспортных задач, построения нечетких контроллеров. В конце 1980-х годов внимание разработчиков ИИТ акцентировалось на исследовании адаптивных свойств информационных систем, учитывающих умственную активность человека при осуществлении речевых актов, дискурса и принятии решений. С начала 1990-х годов ИИТ все активнее используются в стратегическом менеджменте, управлении ресурсами, реинжиниринге, создании ситуационных центров. Все более заметно внедряются интеллектуальные ин-

формационные технологии аналитической обработки больших массивов информации, технологии поддержки решений. В 1990-х годах в совокупности и взаимосвязи развиваются: экспертные системы реального времени, интеллектуальные агенты, активные системы, достоверный и правдоподобный вывод, эволюционные и квантовые вычисления, когнитивные модели, ситуационные центры и пр. Эксклюзивное место в развитии ИИТ с середины 1990-х годов заняла разработка необходимых условий конвергентности (сходимости) процессов управления, поиска информации и синтеза управленческих решений, направленных на обеспечение необходимых условий устойчивой сходимости этих процессов к намечаемым целям. С 2000 года приобрел новое звучание процесс электронизации деятельности органов власти, бизнеса и населения. Концепция электронной демократии, предполагающая осуществление гражданского контроля, проведение выборов и референдумов, поддержку процессов самоорганизации населения, обеспечение возможности участия населения в принятии государственных решений, расширение технологической возможности обмена мнениями, также предусматривает расширение возможностей интеллектуальных информационных технологий. Концепция электронной коммерции, включающая маркетинг, управление корпоративными ресурсами, повышение качества продукции и услуг, расширение доступа к капиталу, электронные торги, развитие инноваций, поддержку процессов самоорганизации бизнеса, не могла не активизировать работу по дальнейшему развитию систем поддержки решений с помощью ИИТ.

2.2. КЛАССИФИКАЦИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

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

Для ИИС характерны следующие признаки [5]:

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

Каждому из перечисленных признаков условно соответствует свой класс ИИС. Различные системы могут обладать одним или несколькими признаками интеллектуальности с различной степенью проявления.

Средства ИИ могут использоваться для реализации различных функций, выполняемых ИИС.

На рисунке 2.1 приведена классификация ИИС, признаками которой являются следующие интеллектуальные функции:

- коммуникативные способности — способ взаимодействия конечного пользователя с системой;
- решение сложных плохо формализуемых задач, которые требуют построения оригинального алгоритма решения в зависимости от конкретной ситуации, характеризующейся неопределенностью и динамичностью исходных данных и знаний;

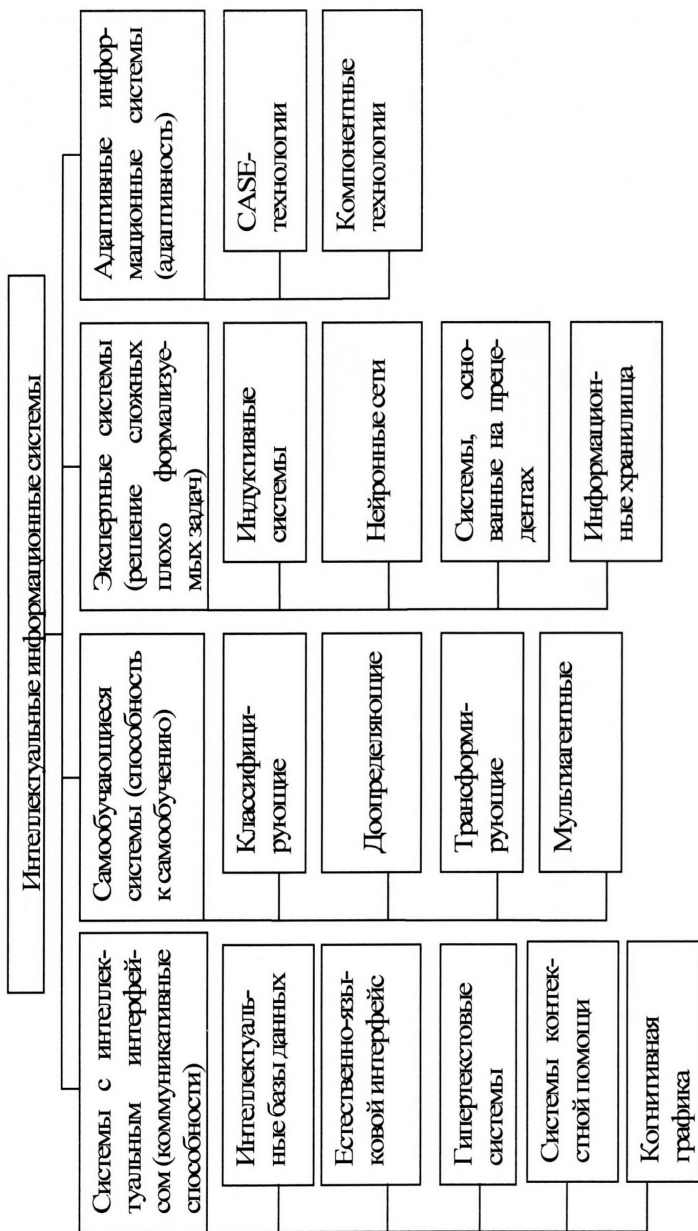


Рис. 2.1. Классификация интеллектуальных информационных систем

- способность к самообучению — умение системы автоматически извлекать знания из накопленного опыта и применять их для решения задач;
- адаптивность — способность системы к развитию в соответствии с объективными изменениями области знаний.

2.3. ПЕРСПЕКТИВНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ, СОЗДАНИЯ, АНАЛИЗА И СОПРОВОЖДЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

2.3.1. Системы с интеллектуальным интерфейсом

Применение ИИ для усиления коммуникативных способностей информационных систем привело к появлению систем с интеллектуальным интерфейсом, среди которых можно выделить следующие типы.

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

2. **Естественно-языковой интерфейс** применяется для доступа к интеллектуальным базам данных, контекстного поиска документальной текстовой информации, голосового ввода команд в системах управления, машинного перевода с иностранных языков. Для реализации естественно-языкового интерфейса (ЕЯ-интерфейса) необходимо решить проблемы морфологического, синтаксического и семантического анализа, а также задачу синтеза высказываний на естественном языке. При морфологическом анализе осуществляются распознавание и проверка правильности написания слов в словаре. Синтаксический

контроль предполагает разложение входных сообщений на отдельные компоненты, проверку соответствия грамматическим правилам внутреннего представления знаний и выявление недостающих частей. Семантический анализ обеспечивает установление смысловой правильности синтаксических конструкций. В отличие от анализа, синтез высказываний заключается в преобразовании цифрового представления информации в представление на естественном языке.

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

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

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

2.3.2. Экспертные системы

Экспертные системы как самостоятельное направление в искусственном интеллекте сформировалось в конце 1970-х годов. История ЭС началась с сообщения японского комитета по разработке ЭВМ пятого поколения, в котором основное внимание уделялось развитию "интеллектуальных способностей" компьютеров с тем, чтобы они могли оперировать не только данными, но и знаниями, как это делают специалисты (эксперты) при выработке умозаключений. Группа по экспертным системам при Комитете British Computer Society определила ЭС как "воплощение в ЭВМ компоненты опыта эксперта, основанной на знаниях, в такой форме, что машина может дать интеллектуальный совет или принять решение относительно обрабатываемой функции". Одним из важных свойств ЭС является спо-

способность объяснить ход своих рассуждений понятным для пользователя образом [5].

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

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

Обычно к ЭС относят системы, основанные на знаниях, т. е. системы, вычислительная возможность которых является в первую очередь следствием их наращиваемой базы знаний (БЗ) и только во вторую очередь определяется используемыми методами. Методы инженерии знаний (методы ЭС) в значительной степени инвариантны тому, в каких областях они могут приме-

няться. Области применения ЭС весьма разнообразны: военные приложения, медицина, электроника, вычислительная техника, геология, математика, космос, сельское хозяйство, управление, финансы, юриспруденция и т. д. Среди этих областей лидируют бизнес, производство, медицина, проектирование и системы управления. Более критичны методы инженерии знаний к типу решаемых задач. В настоящее время ЭС используются при решении задач следующих типов: принятие решений в условиях неопределенности (неполноты), интерпретация символов и сигналов, предсказание, диагностика, конструирование, планирование, управление, контроль и др.

Назначение и особенности ЭС. Знания, которыми обладает специалист в какой-либо области (дисциплине), можно разделить на формализованные (точные) и неформализованные (неточные). Формализованные знания формулируются в книгах и руководствах в виде общих и строгих суждений (законов, формул, моделей, алгоритмов и т. п.), отражающих универсальные знания. Неформализованные знания, как правило, не попадают в книги и руководства в связи с их конкретностью, субъективностью и приближенностью. Знания этого рода являются результатом обобщения многолетнего опыта работы и интуиции специалистов. Они обычно представляют собой многообразие эмпирических (эвристических) приемов и правил.

В зависимости от того, какие знания преобладают в той или иной области (дисциплине), ее относят к формализованным (если преобладают точные знания) или к неформализованным (если преобладают неточные знания) описательным областям. Задачи, решаемые на основе точных знаний, называют формализованными, а задачи, решаемые с помощью неточных зна-

ний, — неформализованными. Речь идет не о неформализуемых, а о неформализованных задачах, т. е. о задачах, которые, возможно, и формализуемы, но эта формализация пока неизвестна.

Как правило, неформализованные задачи обладают неполнотой, ошибочностью, неоднозначностью и/или противоречивостью знаний (как данных, так и используемых правил преобразования).

Экспертные системы не отвергают и не заменяют традиционного подхода к программированию, они отличаются от традиционных программ тем, что ориентированы на решение неформализованных задач и обладают следующими особенностями:

- алгоритм решений не известен заранее, а строится самой ЭС с помощью символических рассуждений, базирующихся на эвристических приемах;
- ясность полученных решений, т. е. система "осознает" в терминах пользователя, как она получила решение;
- способность анализа и объяснения своих действий и знаний;
- способность приобретения новых знаний от пользователя-эксперта, не знающего программирования, и изменения в соответствии с ними своего поведения;
- обеспечение "дружественного", как правило, естественно-языкового (ЕЯ) интерфейса с пользователем.

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

в качестве методов обработки информации применяются процедуры логического вывода и эвристического поиска решений.

Во многих случаях ЭС является инструментом, усиливающим интеллектуальные способности эксперта. Кроме того, ЭС может выступать в роли:

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

Для классификации ЭС используются следующие признаки:

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

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

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

По видам используемых данных и знаний различают ЭС с детерминированными и неопределенными знаниями. Под неоп-

ределенностью знаний и данных понимаются их неполнота, ненадежность, нечеткость.

ЭС могут создаваться с использованием одного или нескольких источников знаний.

В соответствии с перечисленными признаками можно выделить четыре основных класса ЭС (рис. 2.2): классифицирующие, доопределяющие, трансформирующие и мультиагентные.

	Анализ	Синтез	
Детерминированность знаний	Классифицирующие	Трансформирующие	Односточник знаний
Неопределенность знаний	Доопределяющие	Мультиагентные	Несколько источников знаний
	Статика	Динамика	

Рис. 2.2. Основные классы ЭС

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

Доопределяющие ЭС используются для решения задач с не полностью определенными данными и знаниями. В таких ЭС возникают задачи интерпретации нечетких знаний и выбора альтернативных направлений поиска в пространстве возможных решений. В качестве методов обработки неопределенных

знаний могут использоваться байесовский вероятностный подход, коэффициенты уверенности, нечеткая логика.

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

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

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

- реализация альтернативных рассуждений на основе использования различных источников знаний и механизма устранения противоречий;
- распределенное решение проблем, декомпозируемых на параллельно решаемые подзадачи с самостоятельными источниками знаний;
- применение различных стратегий вывода заключений в зависимости от типа решаемой проблемы;

- обработка больших массивов информации из баз данных;
- использование математических моделей и внешних процедур для имитации развития ситуаций.

2.3.3. Самообучающиеся системы

Самообучающиеся интеллектуальные системы основаны на методах автоматической классификации ситуаций из реальной практики или на методах обучения на примерах. Примеры реальных ситуаций составляют так называемую обучающую выборку, которая формируется в течение определенного исторического периода. Элементы обучающей выборки описываются множеством классификационных признаков.

Стратегия "обучения с учителем" предполагает задание специалистом для каждого примера значений признаков, показывающих его принадлежность к определенному классу ситуаций. При обучении "без учителя" система должна самостоятельно выделять классы ситуаций по степени близости значений классификационных признаков.

В процессе обучения проводится автоматическое построение обобщающих правил или функций, описывающих принадлежность ситуаций к классам, которыми система впоследствии будет пользоваться при интерпретации незнакомых ситуаций. Из обобщающих правил, в свою очередь, автоматически формируется база знаний, которая периодически корректируется по мере накопления информации об анализируемых ситуациях.

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

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

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

1. Выбор классификационного признака из множества заданных.

2. Разбиение множества примеров на подмножества по значению выбранного признака.

3. Проверка принадлежности каждого подмножества примеров одному из классов.

4. Проверка окончания процесса классификации. Если какое-либо подмножество примеров принадлежит одному подклассу, т. е. у всех примеров этого подмножества совпадает значение классификационного признака, то процесс классификации заканчивается.

5. Для подмножеств примеров с несовпадающими значениями классификационных признаков процесс распознавания продолжается, начиная с первого шага. При этом каждое подмножество примеров становится классифицируемым множеством.

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

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

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

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

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

Системы, основанные на прецедентах, применяются для распространения знаний и в системах контекстной помощи.

Информационные хранилища отличаются от интеллектуальных баз данных тем, что представляют собой хранилища значимой информации, регулярно извлекаемой из оперативных баз данных. Хранилище данных — это предметно-ориентированное, интегрированное, привязанное ко времени, неизменяемое собрание данных, применяемых для поддержки процессов принятия управленческих решений. Предметная ориентация означает, что данные объединены в категории и хранятся в соответствии с теми областями, которые они описывают, а не с приложениями, которые их используют. Привязанность данных ко времени выражает их "историчность", т. е. атрибут времени всегда явно присутствует в структурах хранилища данных. Неизменяемость означает, что, попав однажды в хранилище, данные уже не изменяются, в отличие от оперативных систем, где данные присутствуют только в последней версии, поэтому постоянно меняются. Технологии извлечения знаний из хранилищ данных основаны на методах статистического анализа и моделирования, ориентированных на поиск моделей и отношений, скрытых в совокупности данных.

Для извлечения значимой информации из хранилищ данных имеются специальные методы (OLAP-анализ, Data Mining или Knowledge Discovery), основанные на применении методов математической статистики, нейронных сетей, индуктивных методов построения деревьев решений и др. [5].

Технология OLAP (On-Line Analytical Processing) — оперативный анализ данных — предоставляет пользователю средства для формирования и проверки гипотез о свойствах данных или отношениях между ними на основе разнообразных запросов к базе данных. Они применяются на ранних стадиях процесса извлечения знаний, помогая аналитику сфокусировать внимание на важных переменных. Средства Data Mining отличаются от OLAP тем, что кроме проверки предполагаемых зависимостей они способны самостоятельно (без участия пользователя) генерировать гипотезы о закономерностях, существующих в данных, и строить модели, позволяющие количественно оценить степень взаимного влияния исследуемых факторов на основе имеющейся информации.

2.3.4. Адаптивные информационные системы

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

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

Адаптивные свойства информационных систем обеспечиваются за счет интеллектуализации их архитектуры. Ядром таких систем является постоянно развиваемая модель проблемной области, поддерживаемая в специальной базе знаний — репозитории. Ядро системы управляет процессами генерации или переконфигурирования программного обеспечения.

В процессе разработки адаптивных информационных систем применяется оригинальное или типовое проектирование. Оригинальное проектирование предполагает разработку информационной системы с "чистого листа" на основе сформулированных требований. Реализация этого подхода основана на использовании систем автоматизированного проектирования, или CASE-технологий.

При типовом проектировании осуществляется адаптация типовых разработок к особенностям проблемной области. Для реализации этого подхода применяются инструментальные средства компонентного (сборочного) проектирования информационных систем (R/3, BAAN IV, Prodis и др.).

Главное отличие подходов состоит в том, что при использовании CASE-технологии на основе репозитория при изменении проблемной области каждый раз выполняется генерация программного обеспечения, а при использовании сборочной технологии — конфигурирование программ и только в редких случаях — их переработка.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте краткую характеристику систем с интеллектуальным интерфейсом, экспертных систем, самообучающихся систем и адаптивных информационных систем.

2. Каким требованиям должны удовлетворять адаптивные системы?

3. Перечислите сферы применения экспертных систем.

4. Назовите основные классы экспертных систем.

5. Какие этапы составляют поиск решения на основе аналогий в системах, основанных на прецедентах?

3. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ В ОБЛАСТИ ПРИМЕНЕНИЯ И УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫМИ СИСТЕМАМИ

3.1. ПРОЦЕССЫ ПО РАЗВИТИЮ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ НА ВСЕХ СТАДИЯХ ИХ ЖИЗНЕННОГО ЦИКЛА

В настоящее время в технологиях разработки сложных систем наблюдается сдвиг от изолированных инструментов (даже адекватных и мощных) к интегрированным и сбалансированным инструментальным комплексам поддержки всего жизненного цикла разработки. При этом современные тенденции в данной области связываются с интеллектуализацией инструментов на основе использования методов и средств представления и обработки знаний, а также интеграцией таких интеллектуальных инструментов в общие технологические среды, использующие классические подходы, разработанные и апробированные в технологии программирования.

Дальнейшим развитием функциональных возможностей интеллектуальных систем является применение парадигмы мультиагентных систем (МАС), которая имеет уже более чем десятилетнюю историю активных исследований и разработок. Но одной из самых "горячих" точек этой бурно развивающейся области в настоящее время является инструментарий для создания МАС, так как проектирование и реализация таких систем — задача сложная, трудоемкая и предполагающая наличие у разработчиков значительного объема знаний и опыта в тематически разных плоскостях — от представления и обработки знаний до проектирования распределенных приложений и агентно-ориентированного программирования.

3.1.1. Инструментальные средства разработки МАС

В целом процесс создания МАС хорошо вписывается в рамки современных технологий разработки и реализации сложных программных систем. Ведь и здесь необходимы такие традиционные этапы, как анализ, проектирование и реализация, отладка и тестирование, интеграция и сопровождение. При этом агентно-ориентированное проектирование и программирование во многом схожи с объектно-ориентированным подходом к созданию сложных программных комплексов. Однако в данной области создатели МАС работают со значительно более сложными сущностями (агентами) и на более высоком уровне абстракции, чем это обычно делается в рамках объектно-ориентированной парадигмы. Поэтому инструментальные средства поддержки разработки МАС должны быть более "сообразительными" и обеспечивать разные активности в рамках интегрированного окружения.

Состояние исследований

Существует много публикаций и интернет-ресурсов, связанных с обсуждением средств проектирования и реализации агентов и мультиагентных систем. Впечатляющая коллекция ссылок на соответствующие проекты представлена на сайте Intelligent Information Integration [13, 2000]. Не имея места для сколько-нибудь подробного обсуждения таких проектов и инструментария, в них разрабатываемого, в данной работе мы остановимся на двух мощных инструментариях — AgentBuilder и ZEUS [AgentBuilder, 1999; Collis et al., 1999] и их кратком сравнении.

Философия инструментария AgentBuilder адресована разработчикам программного обеспечения, которым нужны мощ-

ные средства поддержки разработки агентно-ориентированных приложений. Цель проекта ZEUS — обеспечение быстрого прототипирования новых приложений за счет "внедрения" в инструментарий общих принципов проектирования агентных систем и компонент, лежащих в основе уже существующих MAC. Анализ архитектуры и функциональных возможностей этих инструментальных систем показывает, что AgentBuilder ближе к использованию методов и средств инженерии знаний, разработанных в рамках ИИ, а ZEUS — к подходу, связанному с повторным использованием и реинжинирингом компонент, характерному для современной технологии программирования.

AgentBuilder состоит из двух основных компонент — *Toolkit* и *Run-Time System*. Первая включает средства для управления процессом разработки, анализа предметной области, в которой должны функционировать агенты, проектирования среды коммуникации агентов, спецификации поведения отдельных агентов, а также отладки и тестирования созданного программного обеспечения. Вторая компонента инструментария AgentBuilder обеспечивает окружение периода исполнения. Реализован инструментарий на языке Java.

Инструментарий ZEUS состоит из множества компонент (также реализованных на языке Java), которые можно разбить на три функциональные группы (библиотеки): *Agent Component Library*, *Agent Building Tool* и *Suite of Utility Agents*. Детальный анализ состава библиотек ZEUS показывает, что здесь имеется более широкий набор полезных процедур (*utility tools*), чем в AgentBuilder. Различаются эти инструментальные системы и тем, что AgentBuilder ориентирован на интерпретацию специфици-

каций агентов, а ZEUS манифестирует использование режима компиляции.

AgentBuilder поддерживает расширение ментальной модели Шохам [Shoham, 1993], позволяющее описывать поведение агентов на основе специальных правил. Поведение агентов в ZEUS специфицируется на основе модели вложенных графов (nested graphs model). Оба инструментария используют KQML в качестве базиса коммуникации агентов. Язык представления знаний системы AgentBuilder — RADL (Reticular's Agent Definition Language) — это LISP-подобный по форме и продукционно-фреймовый по сути язык представления знаний (ЯПЗ). Представление знаний в системе Zeus кажется более простым и ограниченным. По существу, оно базируется на использовании композиций базовых действий (generic actions) и настройке их на конкретное приложение за счет параметров.

Анализ рассмотренных выше инструментальных средств построения MAC, а также сравнение многих других средств этого класса показывают, что в настоящее время нет ни одной системы, которая бы "покрывала" даже основные потребности разработчиков MAC и делала бы это на основе действительно интеллектуальных инструментов. Результаты сравнения инструментария AgentBuilder и ZEUS сведены в таблице 3.1 (дополнительная колонка таблицы фиксирует свойства инструментария AgSDK проекта AgentFlight).

Анализ состояния работ и тенденций в области инструментальной поддержки разработки мультиагентных систем позволяет выдвинуть следующие основные требования к MAC-инструментариям:

- эксплицитная поддержка методологии и технологии разработки и реализации MAC;
- ориентация на жизненный цикл MAC;
- поддержка пользователей разного уровня компетентности;
- ориентация на современные парадигмы визуального проектирования и программирования;
- разработка самого инструментария как системы, основанной на знаниях.

Таблица 3.1. Основные требования к MAC-инструментарю

	Agent Builder	ZEUS	Agent Flight
Управление проектом	+++	+++	+++
Инновации в разработке агентов	+++	++	++
Инновации в разработке MAC	++	+++	++++
Уровень использования знаний	+++	+	+++++
Набор вспомогательных процедур	+++	+++	++
Средства отладки	+++	++++	++
Технология реализации	+++	+++	++++

Перечисленные выше требования и положены в основу проектных решений при создании инструментария AgSDK, обсуждаемого в оставшейся части настоящей работы.

3.1.2. AgSDK: среда разработки и реализации MAC

Интеллектуальная среда разработки мультиагентных систем AgSDK, ориентированная на поддержку эксплицитного представления методологии и технологии создания таких сис-

тем, должна быть функционально замкнутой и платформенно независимой. Реализация программного обеспечения AgSDK осуществляется по контракту с компанией VERIDAN ERIM International Inc. Разработка инструментария осуществляется в среде JDK 1.2 и Swing 1.2, что позволяет использовать AgSDK на всех вычислительных платформах со стандартным окружением Sun Java, версия 1.2 и выше.

Семантически данная программная платформа делится на две основные части: средства визуального проектирования MAC (AgSDK) и объектную библиотеку AgentLib, которая обсуждается в отдельной статье [Матюшкин и др., 2000].

Методология и управление

В соответствии с общей концепцией инструментарий AgSDK архитектурно представлен интегрированным набором управляющих компонент, специализированных редакторов и специальных процедур, обеспечивающих разработчикам MAC интеллектуальное ассистирование в процессе спецификации мультиагентной среды в целом, поведения отдельных агентов, функционирующих в этой среде, а также генерацию собственно мультиагентных приложений. Технологическая схема проектирования и реализации MAC в среде AgSDK может быть описана потоковой диаграммой, представленной на рисунке 3.1.

Начинается разработка со стадии формирования проекта, которая поддерживается компонентами Project Management Tools, включающими два основных инструмента: Project Manager и Repository Manager. Первый обеспечивает идентификацию проекта, управление его выполнением и реинжиниринг уже использованных проектных решений, а второй — контроль доступа ко всей информации по проекту.

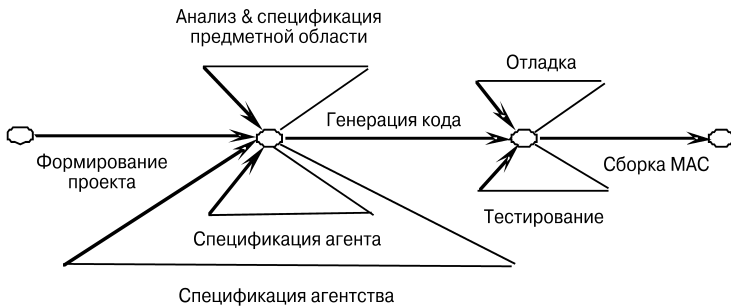
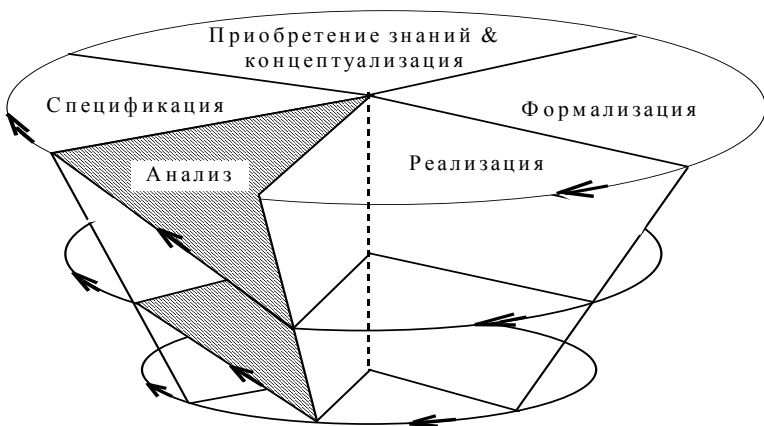


Рис. 3.1. Технологическая схема проектирования и реализации MAC на базе AgSDK

Онтологический инструментарий AgSDK

Обмен сообщениями между агентами MAC в рассматриваемом подходе, как и в большинстве современных разработок, осуществляется на основе KQML-протокола [Finin et al., 1997]. Но взаимодействие — это существенно более сложный процесс, предполагающий понимание смысла передаваемых сообщений и, как следствие, наличие общих и разделяемых знаний. На современном этапе такие знания, как правило, представляются в виде онтологий [Guarino, 1996]. Детальное обсуждение онтологического инструментария AgSDK — тема отдельной работы. Поэтому здесь мы лишь зафиксируем принципы, положенные в основу его проектирования и реализации.

Дуга "Анализ & спецификация предметной области" технологической схемы разработки MAC инкапсулирует жизненный цикл создания онтологий (рис. 3.2), который в рамках инструментария AgSDK базируется на интеграции модели METHONTOLOGY [Fernandez, et al., 1997] и спиральной модели Боэма [Boehm, 1986].



**Рис. 3.2. Жизненный цикл разработки онтологий
в инструментари AgSDK**

Специфика нашего подхода в том, что здесь каждый виток спирали представляет технологическую карту активностей (спецификация, концептуализация, формализация и реализация) и связей между ними, а переход с уровня на уровень — анализ полученных результатов. Таким образом процесс разработки онтологии трансформирует "начальный продукт" (потребность в создании определенной онтологии) в "конечный продукт" (документированную и закодированную в формальном языке онтологию).

Соответствующие активности поддерживаются на уровне AgSDK компонентой Domain Specification, которая представлена тремя инструментами: Ontology Manager, Ontology Editor и Ontology Specification Converter.

Ontology Manager обеспечивает разработчика средствами идентификации создаваемой онтологии и управления всем процессом. В качестве основных характеристик идентификации используется набор, аналогичный разработанному в рамках проекта (ONTO) [Vega et al., 1999].

Ontology Editor поддерживает следующие две стадии процесса разработки онтологии. При этом стадия приобретения знаний и концептуализации дает структурированную спецификацию всех концептов онтологии и их атрибутов, описание значимых отношений между концептами, а также требуемых аксиом и правил вывода, связанных с концептами и отношениями. Стадия формализации базируется на использовании специального языка представления знаний OntoPilot, который является развитием языка представления знаний (ЯПЗ) Pilot/2 [Khoroshevsky, 1994] с учетом результатов, полученных в проектах Ontobroker и SHOE [Benjamins et al., 1998; Heflin, et al., 1998]. Ontology Editor представляет собой интеллектуальный графический редактор, обеспечивающий разработчика средствами визуального проектирования онтологий.

Ontology Specification Converter реализует последнюю стадию создания онтологии — трансформацию OntoPilot-спецификации в соответствующие классы библиотеки AgLib и/или в XML-представление онтологии, интегрированные с машиной вывода OntoPilot.

Стадия анализа "замыкает" виток в спирали жизненного цикла онтологии и обеспечивает при необходимости переход к следующему витку этой спирали. А собственно спецификация MAC осуществляется компонентами Agency и Agent, обсуждаемыми ниже.

Для определенности дальнейшего изложения рассмотрим модельную задачу проектирования МАС.

Модельный пример

Пусть проектируется мультиагентная система, в архитектуре которой представлены несколько групп агентов, каждой из которых "руководит" свой менеджер. Предположим также, что в каждую группу могут входить агенты разных типов. Задача менеджеров групп состоит во взаимодействии с другими менеджерами с целью накопления информации о составе агентных групп и типах агентов, в них представленных, для того, чтобы в дальнейшем сформировать межгрупповые коллективы, куда бы входили лишь агенты одного и того же типа. После формирования таких коллективов взаимодействие между их членами должно осуществляться без вмешательства менеджеров групп.

Для определенности предположим, что в разрабатываемой МАС представлены две группы (группа А и группа В), "руководимые" менеджерами ManagerA и ManagerB соответственно. Пусть также типы агентов, функционирующих в каждой группе, ограничены следующими: "Танцоры" (Dancers), "Певцы" (Singers) и "Художники" (Painters). Тогда результатом работы МАС должно стать образование трех коллективов — "любителей танца", "любителей пения" и "любителей рисования", где общение заключается в реализации определенной схемы диалога [Хорошевский, 1999]. В общем случае такая схема включает приветствие, собственно взаимодействие и прощание. Предположим, что в ответ на предложение по установлению контакта каждый агент сформированного коллектива может реагировать случайным образом — либо принять предложение, либо отказаться от взаимодействия. В последнем случае агент-

инициатор общения "засыпает" на определенное время, после чего повторяет попытку установления контакта.

Проектирование агентной среды

Как отмечалось выше, одной из основных задач, возникающих перед разработчиками MAC, является проектирование мультиагентной среды, где должны функционировать агенты. В AgSDK решение этой задачи возлагается на интеллектуальную компоненту Agency Suite, обеспечивающую создание, редактирование и визуализацию объектов агенства и допустимых взаимодействий между ними на основе знаний об архитектуре создаваемой MAC. Экранная форма с результатами проектирования агенства для обсуждаемого ниже примера представлена на рисунке 3.3.

Для модельной задачи, рассмотренной выше, процесс проектирования агенства заключается в спецификации его состава (в нашем случае это менеджер MasterAgentA и агенты его группы DancerA, PainterA и SingerA, а также менеджер MasterAgentB и агенты его группы DancerB, PainterB и SingerB).

Для взаимодействия со средой в агенство включен и специальный объект Judge, основной задачей которого является инсталляция менеджеров групп. Кроме спецификации всех объектов агенства, в компоненте Agency Suite определяются и потоки сообщений, которыми данные объекты обмениваются между собой и со средой.

Для ассистирования в процессе спецификации компонента Agency Suite использует специальную фреймовую БЗ, основными прототипами в которой являются фреймы GenericMaster, GenericAgent и Environment. Результат проектирования погру-

жается в базу текущего проекта и используется в дальнейшем на этапе генерации исполнительного кода.

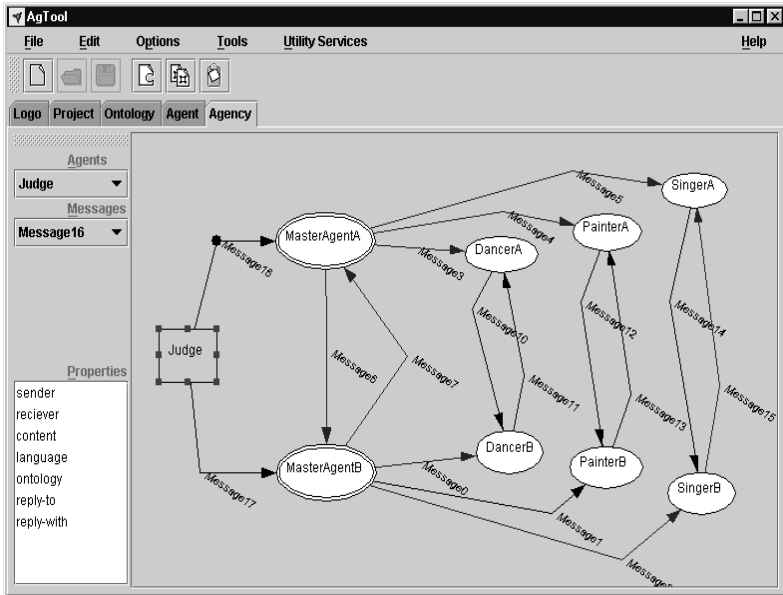


Рис. 3.3. Панель спецификации агентства инструментария AgSDK

Спецификация агентов

Второй инструментальной составляющей проектирования МАС является Agent Suite. Основная ее задача — спецификация поведения агентов и среды. Заметим, что у каждого из агентов может быть несколько "линий поведения", которые могут существовать одновременно и быть независимыми или связанными с друг другом. Так, например, для нашего модельного примера целесообразно считать, что генетическое поведение менеджеров содержит две линии — активную (DonorMasterBehaviour) и пассивную (RecipientMasterBehaviour). Первая из

них предполагает, что агент, ее реализующий, является инициатором общения, а вторая — что он лишь отвечает на активность других агентов. При этом обе линии поведения агента могут активироваться лишь в случае согласия от партнера по взаимодействию. В работе [Хорошевский, 1999] рассмотрена система потоковых диаграмм, реализующих специфицированное выше поведение. На рисунке 3.4 показана панель Agent Suite инструментария AgSDK, в рамках которой происходит спецификация поведения агентов.

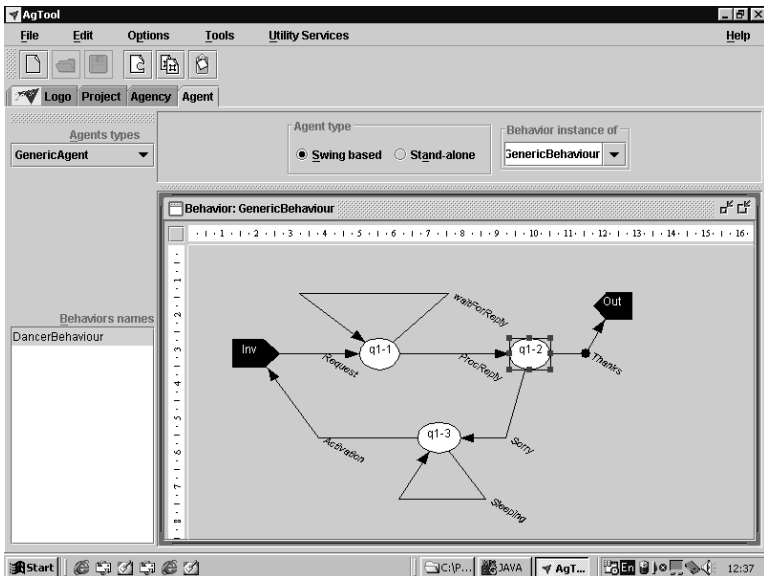


Рис. 3.4. Панель спецификации поведения агентов

Диаграмма Inv, представленная на рисунке 3.4, описывает компоненту "установление контакта". Перформативы, использованные в этой диаграмме, неоднородны по своему уровню и сложности. Так, например, Request, Thanks, Sorry имеют непо-

средственное отражение на соответствующие перформативы KQML, а waitForReply, ProcReply и некоторые другие представляются последовательностью действий, среди которых, в частности, присутствуют и перформативы KQML.

Описанный выше инструментарий расширяется за счет компоненты создания онтологий и более мощного генератора исполнительных программ.

3.2. ОСНОВНЫЕ ТЕНДЕНЦИИ РАЗВИТИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ, СВЯЗАННЫХ С ИЗМЕНЕНИЯМИ УСЛОВИЙ В ОБЛАСТИ ПРИМЕНЕНИЯ

Современный уровень развития информационных технологий позволяет сегодня воплощать на практике фундаментальные решения в области искусственного интеллекта (ИИ), разработанные в последние десятилетия прошлого века, создавая не просто корпоративные информационные системы, но интеллектуальные системы управления предприятием. В то же время повышение уровня "интеллектуальности" корпоративных информационных систем отвечает потребностям бизнеса — руководителю нужны интеллектуальные системы, позволяющие перенести отработанные многолетней практикой управленческие решения в область компьютерных технологий, высвободив интеллектуальный потенциал для стратегического мышления, определения направлений развития предприятия и решения нестандартных задач, требующих не искусственного, но естественного интеллекта.

Эволюция корпоративных информационных систем определяется также такими факторами изменения производственно-экономической системы, как интеграция видов бизнеса и развития бизнеса e-business on demand (бизнес по требованию, биз-

нес по запросу), необходимость адекватно реагировать на все изменения внешней среды, динамично перераспределяя ресурсы предприятия.

Корпоративную информационную систему (КИС) можно отнести к классу интеллектуальных, если она использует методы искусственного интеллекта при решении задач, возникающих в процессе управления. В частности, интеллектуальная система управления предприятием должна обеспечивать ситуационную поддержку принятия решений, автоматизировать процесс поиска управляющих решений на основе накопленных знаний о предметной области, обеспечивать принятие решений в условиях неопределенности.

Корпоративная интеллектуальная система управления (КИСУ) представляет собой новый уровень развития КИС, когда система автоматически определяет место возникновения несоответствия (противоречия) или отклонения, распознает ситуацию, сложившуюся на объекте и в среде управления, определяет множество возможных решений, а в определенных ситуациях реализует управляющее воздействие и автоматически осуществляет контроль его исполнения. Системы поддержки принятия решений, экспертные системы, информационно-аналитические системы, системы прогнозирования и моделирования должны быть интегрированы в корпоративную информационную систему управления предприятием.

Требования к корпоративной интеллектуальной системе управления

1. КИСУ должна поддерживать все задачи, возникающие в процессе управления: планирование — учет — контроль —

анализ — принятие решений, используя методы ИИ на каждом шаге цикла управления.

2. Архитектура системы, помимо реляционной базы данных, должна включать базу знаний, содержащую интенциональную составляющую описания предметной области (ПО), а также средства автоматизации пополнения базы знаний: извлечение закономерностей из множества фактов на основе методов индуктивного вывода, выявления неявных причинно-следственных связей, расширение описания ПО методом дедуктивного вывода, классификации понятий предметной области.

3. Система должна поддерживать интеллектуальные технологии, решающие задачи:

- прогнозирования развития ситуации и изменения состояний внутренней и внешней сред;
- моделирования описания предметной области и последствий принятия управляющих решений;
- распознавания объектов, их состояний и ситуаций, сложившихся на объекте и в среде управления.

4. Применение интеллектуальных технологий должно охватывать все:

- контуры управления (оперативный и регулярный менеджмент, стратегическое и ситуационное управление);
- сферы управления (управление финансовыми, трудовыми, производственными и материально-техническими ресурсами, управление клиентами и поставщиками и т. д.);
- бизнес-процессы предприятия (основные, обеспечивающие, управленческие, инвестиционные);

- функциональные подсистемы корпоративной системы менеджмента (система электронного документооборота, CRM, SRM, сбыт, материально-техническое обеспечение и т. д.);
- уровни управления — корпорация, дочернее предприятие, структурное подразделение;
- цели управления — снижение производственных, финансовых, экологических и тому подобных рисков, повышение надежности и эффективности работы производственной системы, совершенствование системы менеджмента качества и т. д.;
- роли и рабочие места (рабочее место руководителя, аналитика, администратора, владельца бизнес-процесса и т. д.).

Интеграция информационных и технологических систем

Повышение интеллектуального уровня корпоративных информационных систем обеспечивает также развитие геоинформационных (ГЕО) технологий, аэрокосмических и телекоммуникационных технологий и их интеграцию с информационными технологиями. Сегодня стало возможным применять интеллектуальные технологии в рамках интегрированной системы менеджмента, управляющей в рамках единого инфокоммуникационного пространства как производственными и техническими объектами, так и социально-экономической системой. XXI век — действительно время реализации фантазий специалистов в области ИИ XX века. В качестве примера можно привести КИСУ, интегрированную с системой управления интеллектуаль-

ным домом, когда все параметры здания обрабатываются информационной системой в общей системе показателей.

Ярким примером интеграции информационных и технологических систем является интеллектуальный аэропорт, где интегрируются достижения ГЕО, телекоммуникационных, информационных технологий и методов ИИ. В едином инженерно-телекоммуникационном и информационном пространстве автоматизированы процессы от видеонаблюдения и управления экологической безопасностью до управления клиентами. Комплексное управление работой аэропорта включает интеграцию и синхронизацию работы таких процессов, как:

- управление воздушным движением (взлет/посадка);
- управление наземными службами;
- формирование пассажиропотоков (бронирование и продажа билетов);
- формирование грузопотоков;
- обслуживание самолетного парка;
- управление расписанием;
- управление материально-техническими и трудовыми ресурсами;
- управление финансовыми ресурсами и т. д.;
- управление инфраструктурой (расчет арендных платежей, планирование схем автостоянок);
- управление процессом реконструкции (моделирование расположения объектов);
- визуализация объектов и процессов управления и т. д.

Интеграция ERP-систем и автоматизированных систем управления технологическими объектами позволяет использовать методы ИИ для мониторинга и интегрированного управления всеми видами ресурсов предприятия, включая производственные, материально-технические и финансовые ресурсы.

"Оркестр играет без дирижера: размышления об эволюции некоторых технических систем и управлении ими" — так называлась книга выдающихся российских ученых В. И. Варшавского и Д. А. Поспелова, идеи которых по-прежнему актуальны и активно используются.

Аккумуляция в хранилищах данных информации, отражающей жизненные циклы производственных, технологических, финансово-хозяйственных и социально-экономических процессов предприятия, позволяет сегодня использовать методы ИИ для поиска корреляций, тенденций, взаимосвязей и закономерностей между данными, расширить область использования систем поддержки принятия решений и переместить их за пределы сферы оперативного управления в область поддержки принятия стратегических решений.

Методы ИИ позволяют строить замкнутые контуры управления, поддерживать принятие решений в условиях неопределенности, распознавать ситуации, сложившиеся на объекте и в среде управления, поддерживать информационную целостность и безопасность баз данных и баз знаний, метазнаний, извлекать, обобщать и приобретать знания.

В таблице 3.2 показано место интеллектуальных технологий в системе управления предприятием.

Интеллектуальный документооборот

Рассмотрим применение интеллектуальных технологий на примере создания важной интеграционной компоненты корпоративной информационной системы управления — системы электронного документооборота.

Интеллектуальный документооборот является саморазвивающейся, самообучающейся и саморегулирующейся системой.

Таблица 3.2. Место интеллектуальных технологий в системе управления предприятием

Интеллектуальные технологии	Место в системе управления
Представление знаний: фреймовые модели, логико-лингвистические модели, семантические сети, семиотические системы, производственные модели	Нормативно-справочная информация, база знаний, мета-знания, хранилище данных, основные данные корпоративной информационной системы
Вывод на сетях, формальные системы	Целостность баз знаний, баз данных
Классификация, кластеризация понятий предметной области	Система электронного документооборота, нормативно-справочная информация
Накопление знаний, обучение, самоорганизация	Функциональные подсистемы
Прогнозирование ситуаций, автоматическое порождение гипотез, моделирование рассуждений	Системы поддержки принятия решений, рабочее место руководителя
Интеллектуальный анализ данных и обработка образной информации, обнаружение закономерностей и извлечение знаний	Работа с хранилищем данных (средства DATA MINING)
Распознавание ситуаций, объектов и их состояний на основе нейронных сетей	ЭЦП, функциональные блоки, в частности ФБ Управление финансами

Продолжение табл. 3.2

Интеллектуальные технологии	Место в системе управления
Нечеткие логики	Поддержка принятия решений в условиях неопределенности
Ситуационное, динамическое планирование	Системы бюджетирования и бизнес-планирования, управление инвестициями, стратегическое планирование
Моделирование	Обследование ПО, концептуальное проектирование
Саморегуляция, самообучение, самонастройка	Интеллектуальные бизнес-процессы, интеллектуальный документооборот, адаптивная стратегия

Для реализации подобных свойств система электронного документооборота должна включать механизмы генетического наследования и классификации, инструменты извлечения знаний из неструктурированной информации, механизмы накопления и обработки статистической информации, в частности информации, образующейся в процессе согласования документов или движения служебных записок.

Интегрированная модель представления знаний, положенная в основу системы интеллектуального документооборота, помимо знаний о формировании структуры и шаблонов документов, маршрутных схем движения документов, должна содержать лингвистические знания и знания о предметной области, включая знания об объекте, субъекте, задачах и процессах управления.

Потребность в модели предметной области порождает тот факт, что система интеллектуального документооборота наиболее эффективно работает, когда она полностью интегрирована с информационной системой управления.

При построении системы интеллектуального документооборота используются следующие интеллектуальные технологии:

- систематизация потоков документов — механизм кластеризации и классификации;
- оценка полноты, целостности и непротиворечивости множества документов — аппарат формальной логики;
- "умное" движение документов — производционные правила;
- генерация событий и новых документов — процедуры в структуре документов, инициируемые ситуациями;
- автоматизированный поиск ответов на запросы — генерация ответов за счет семантического анализа текста запроса;
- накопление знаний для поддержки принятия оптимальных решений — создание баз знаний.

Интеграция моделей представления знаний в системе интеллектуального документооборота включает:

- логические модели — аксиоматика ПО-оценки целостности, полноты и непротиворечивости;
- производционные модели — правила изменения предметной области — автоматизация формирования маршрутных схем;
- семантические сети — семантические структуры для выявления угроз, прогнозирования событий и фор-

мирования ответов на запросы при работе с документами;

- нейронные сети — задачи распознавания и классификации документов.

Инструментальные средства поддержки интеллектуальных технологий

В таблице 3.3 приведены примеры инструментальных средств поддержки интеллектуальных технологий, представляющих собой интегрированный комплекс совместимых продуктов.

Таблица 3.3. Инструментальные средства поддержки интеллектуальных технологий

Инструментальная среда	Назначение
Инструментальная среда ARIS (Architecture of Integrated Information System)	Развитая среда описания и моделирования ПО на стадии концептуального проектирования КИСУ
POWERSIM	Система динамического моделирования и прогнозирования
ERP SAP R3 (Enterprise Resource Planning)	Исполнительная система, поддерживающая процессный подход к управлению предприятием и позволяющая устанавливать точки контроля, анализа и принятия решений для выработки предупреждающих и корректирующих воздействий
Business Intelligence (BI my SAP BS)	Инструментальная среда, которая помимо средств поддержки хранилища данных и формирования многомерных отчетов включает инструментарий, позволяющий строить системы целей и показателей,

Продолжение табл. 3.3

Инструментальная среда	Назначение
Business Intelligence (BI my SAP BS)	деревья решений и ситуационные планы, фиксировать математические и ассоциативные связи между понятиями
SAP ERP 2005	Инструментальная среда для создания интеллектуальной системы управления, интегрированная с технологическими системами
Adobe LiveCycle Forms Designe	Среда разработки интеллектуальных документов в виде форм, бланков, отчетов

Специализация в области интеллектуальных технологий в управлении предприятием

Таким образом, одним из направлений подготовки специалистов в области ИИ может быть специализация, нацеленная на практическое применение технологий и методов ИИ для совершенствования и развития корпоративной информационной системы управления предприятием. С этой целью, помимо приобретения навыков и знаний в области технологий, методов и инструментальных средств ИИ, необходимо формирование целостного взгляда на предприятие как объект управления и корпоративную информационную систему как инструмент управления.

Подготовка специалистов должна охватывать такие области знаний, как:

- архитектура, базовые компоненты и функциональные подсистемы КИСУ;
- уровни и этапы проектирования и создания;

- взаимодействие компонент КИСУ, организационно-функциональное проектирование и развитие;
- технология проектирования и описания ПО — модели описания ПО (объектов, бизнес-процессов, логических, структурных и функциональных зависимостей);
- функциональный охват — знание таких функциональных подсистем, как система электронного документооборота, CRM, SRM, система материально-технического обеспечения и т. д.;
- инструментальные средства поддержки процесса проектирования и создания КИСУ и ее подсистем;
- стандарты, используемые в процессе проектирования и создания КИСУ и ее подсистем;
- вопросы интеграции с технологическими системами.

Специалист в области проектирования и создания КИСУ должен обладать знаниями предметной области, владеть такими понятиями, как счет-фактура, спецификации рабочего места, технологическая карта и т. д. Но главное, специалист должен понимать, в чем состоит управление предприятием, различать разные контуры управления — уровень оперативного (регулярного) менеджмента и уровень стратегического управления, задачи управления, точки контроля, анализа и принятия решений.

3.3. ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

3.3.1. Цели обеспечения информационной безопасности

Целями обеспечения информационной безопасности являются обеспечение сохранности и конфиденциальности информации, защита и гарантия доступности, достоверности и целостности информации, избежание утечки информации, минимизация ущерба от событий, несущих угрозу информационной безопасности.

Для обеспечения информационной безопасности должны быть достигнуты следующие цели:

- обеспечение конфиденциальности информации в соответствии с проведенной классификацией;
- обеспечение целостности информации на всех этапах связанных с ней процессов (создание, обработка, хранение, передача и уничтожение) при предоставлении публичных услуг;
- обеспечение своевременной доступности информации при предоставлении публичных услуг;
- обеспечение наблюдаемости, направленной на фиксирование любой деятельности пользователей и процессов;
- обеспечение аутентичности и неотказуемости от транзакций и действий, производимых участниками предоставления публичных услуг;
- учет всех процессов и событий, связанных с вводом, обработкой, хранением, предоставлением и уничтожением данных.

3.3.2. Задачи системы информационной безопасности

Для достижения целей обеспечения безопасности необходимо выполнение следующих задач:

- идентификация информационных ресурсов и систем и их классификация в зависимости от их ценности и важности в процессе предоставления публичных услуг (компоненты анализа должны включать данные, приложения, технологии, телекоммуникационные и аппаратные средства, помещения, человеческие ресурсы и т. д.);
- идентификация угроз информационной безопасности и определение возможных источников угрозы по каждому информационному ресурсу и системе;
- идентификация уязвимых мест по каждой идентифицированной угрозе;
- оценка рисков по идентифицированным информационным ресурсам и системам;
- выбор мероприятий по управлению информационной безопасностью на основе анализа соотношения стоимости их реализации к эффекту от снижения рисков и возможным убыткам в случае нарушения безопасности;
- разработка и внедрение механизма и мер оперативного реагирования на угрозы информационной безопасности и проявления негативных тенденций в функционировании информационных ресурсов и систем предоставления публичных услуг;

- разработка и внедрение системы контроля, позволяющей обеспечивать эффективное функционирование механизма и мер обеспечения безопасности;
- организация процесса эффективного пресечения посягательств на информационные ресурсы и системы;
- организация процесса обеспечения непрерывности предоставления услуг и создание условий для максимально возможного возмещения и локализации наносимого ущерба неправомерными действиями физических и юридических лиц, ослабление негативного влияния последствий нарушения информационной безопасности;
- организация процесса обеспечения безопасности при наличии поставщиков услуг, имеющих доступ к информационным ресурсам или системам при предоставлении публичных услуг.

3.3.3. Принципы обеспечения информационной безопасности

Система обеспечения информационной безопасности должна создаваться на следующих принципах:

- принцип равнопрочности — означает обеспечение защиты оборудования, программного обеспечения и системы управления от всех видов угроз;
- принцип непрерывности — предусматривает непрерывное обеспечение безопасности информационных ресурсов, ИС для непрерывного предоставления публичных услуг;

- принцип разумной достаточности — означает применение таких мер и средств защиты, которые являются разумными, рациональными и затраты на которые не превышают стоимость нарушения информационной безопасности;
- принцип комплексности — для обеспечения безопасности во всем многообразии структурных элементов, угроз и каналов несанкционированного доступа должны применяться все виды и формы защиты в полном объеме. Недопустимо применять отдельные формы или технические средства;
- принцип комплексной проверки — заключается в проведении специальных исследований и проверок, специального инженерного анализа оборудования, верификационных исследований программных средств. Должен осуществляться непрерывный мониторинг аварийных сообщений и параметров ошибок, постоянно должно выполняться тестирование аппаратного и программного оборудования, а также контроль целостности программных средств, как при загрузке программных средств, так и в процессе функционирования;
- принцип надежности — методы, средства и формы защиты должны надежно перекрывать все пути проникновения и возможные каналы утечки информации, для этого допускается дублирование средств и мер безопасности;
- принцип универсальности — меры безопасности должны перекрывать пути угроз независимо от места их возможного воздействия;

- принцип плановости — планирование должно осуществляться путем разработки детальных планов действий по обеспечению информационной защищенности всех компонент системы предоставления публичных услуг;
- принцип централизованного управления — в рамках определенной структуры должна обеспечиваться организовано-функциональная самостоятельность процесса обеспечения безопасности при предоставлении публичных услуг;
- принцип целенаправленности — необходимо защищать то, что должно защищаться в интересах конкретной цели;
- принцип активности — защитные меры обеспечения безопасности в работе процесса предоставления услуг должны претворяться в жизнь с достаточной степенью настойчивости;
- принцип квалификации обслуживающего персонала — обслуживание оборудования должно осуществляться сотрудниками, подготовленными не только в вопросах эксплуатации техники, но и в технических вопросах обеспечения безопасности информации;
- принцип ответственности — ответственность за обеспечение информационной безопасности должна быть ясно установлена, передана в ведение соответствующему персоналу и утверждена всеми участниками в рамках процесса обеспечения информационной безопасности.

3.4. УПРАВЛЯЕМОСТЬ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

В системах, управляющих программным средством, будем выделять четыре типа подсистем — *мониторинга*, т. е. регулярного сбора информации, необходимой для управления; *ручного (интерактивного) управления* для осуществления управляющих воздействий (изменения свойств программного средства) управляющим программным средством; *автоматического управления*, т. е. изменения свойств программного средства без участия управляющих; *автоматизированного управления*, реализующего сочетание ручного и автоматического управления (в частности, ручного управления параметрами средства автоматического управления). Разработку же программного средства можно рассматривать как управление им, начиная с состояния, когда этого программного средства еще нет, но уже есть система, управляющая им, аналогично тому, как сопровождение программного средства обычно рассматривается как его продолжающаяся разработка.

Программное средство как объект управления является многоцелевым объектом, поскольку предназначено для достижения нескольких заданных результатов функционирования (например, заданные результаты функционирования решателя задач и пользовательского интерфейса обычно слабо связаны между собой). Можно выделить как минимум три вида знаний, на основе которых принимаются решения о выработке управляющих воздействий при управлении программными средствами: априорные, явно или неявно представленные знания; знания о задачах, ранее решенных программным средством (о входном потоке данных, обработанных программным средством, и о том результате, который должен получиться после об-

работки); знания об особенностях пользователей этого программного средства. Поскольку эти три вида знаний в значительной степени независимы друг от друга, можно считать, что управление программным средством, т. е. выработка управляющих воздействий, осуществляется в "трехмерном" пространстве, "координатами" которого являются априорные знания, знания о входном потоке и о пользователях программного средства.

Большинство программных средств обладают тем свойством, что время их жизни разбивается на совокупность периодов, в одни из которых они не функционируют, но могут сопровождаться (будем называть их периодами сопровождения), а в другие они функционируют, но не сопровождаются (будем называть их периодами функционирования). Для такого программного средства можно говорить либо об управлении им в периоды сопровождения, когда управляющие воздействия вырабатываются и осуществляются в те периоды времени, когда программное средство не функционирует, либо об управлении им в периоды функционирования, когда выработка и осуществление управляющих воздействий происходят в периоды функционирования программного средства.

Для того чтобы программное средство было управляемым, необходимо, чтобы оно удовлетворяло определенным требованиям, предъявляемым управляющим объектом. Таким образом, средства управления и управляемое ими программное средство должны быть взаимно согласованы — управляющие воздействия, формируемые средствами управления, должны иметь форму, непосредственно воспринимаемую объектом управления.

Архитектуру интеллектуальных систем как класса программных средств, содержащих базу знаний, в первом приближении можно представлять себе как взаимосвязанную тройку, состоящую из базы знаний (и других информационных ресурсов — баз данных, онтологий, метаонтологий и т. п.), решателя задач и пользовательского интерфейса. Частные цели управления этими тремя компонентами интеллектуальной системы в общем случае различны и в значительной степени независимы. В соответствии с этим в архитектуре системы, управляющей интеллектуальной системой, будем выделять подсистемы, управляющие информационными ресурсами, решателем задач и пользовательским интерфейсом (рис. 3.5).

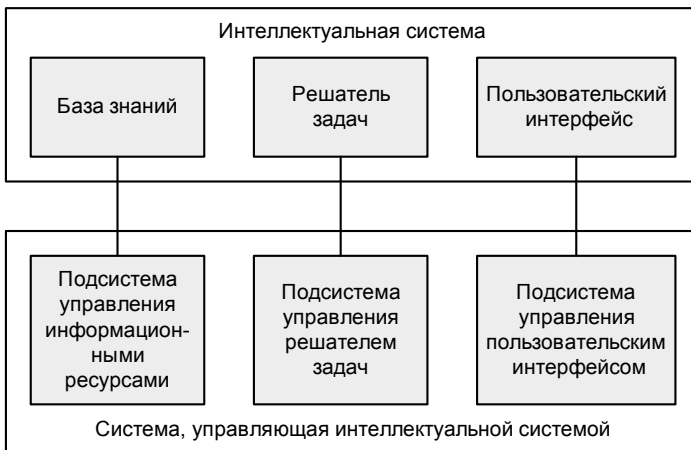


Рис. 3.5. Структура системы управления интеллектуальной системой

Исходя из того, что в архитектуре интеллектуальной системы как объекта управления можно выделить три самостоятельные компоненты (информационные ресурсы, в том числе

базу знаний, решатель задач и пользовательский интерфейс), а в системе, управляющей программным средством, — четыре типа подсистем (мониторинга, ручного управления, автоматического управления и автоматизированного управления), задачи управления интеллектуальными системами можно разделить на двенадцать классов (рис. 3.6).

Информационные ресурсы	Решатель задач	Пользовательский интерфейс
	М	о
	н	и
	т	о
	р	р
	и	н
	н	г
	Р	у
	ч	н
	н	о
	о	е
	у	п
	п	р
	р	а
	в	л
	е	н
	и	е
	А	в
	в	т
	о	м
	а	т
	и	ч
	е	с
	к	о
	о	н
	т	о
	ч	е
	с	к
	о	н
	н	о
	о	е
	у	п
	р	р
	а	в
	л	е
	н	и
	е	

Рис. 3.6. Классификация задач управления интеллектуальными системами

3.5. ЗАДАЧИ УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫМИ СИСТЕМАМИ

В настоящем параграфе будут рассмотрены некоторые общие и частные задачи управления информационными ресурсами, решателями задач и интерфейсами интеллектуальных систем.

При рассмотрении задач управления интеллектуальными системами будут приведены примеры подсистем управления двух основных типов: проблемно-зависимых и проблемно-

независимых. Примеры проблемно-зависимых подсистем управления будут продемонстрированы для интеллектуальной системы интерактивного доказательства теорем и экспертной системы медицинской диагностики с функцией диагностического тренажера. Приводимые в примерах подсистемы управления либо разрабатываются в настоящее время в коллективе, либо планируются к разработке, за исключением случаев, где сказано о существовании прототипа.

3.5.1. Управление информационными ресурсами

Целью управления информационными ресурсами является перманентное приведение их содержания в соответствие с постоянно меняющимися априорными знаниями управляющих и входным потоком данных интеллектуальной системы. В информационном ресурсе важно содержание информации и ее удобное представление для управляющих, а не способ ее компьютерного представления. Исходя из этого, выдвигаются следующие принципы управления информационными ресурсами. Информационные ресурсы должны:

- допускать создание и ручное управление носителями информации без посредников;
- управляться независимо от конкретных интеллектуальных систем;
- быть доступными для обработки решателями различных интеллектуальных систем.

Далее будут рассматриваться только задачи управления концептуальными информационными ресурсами, которые могут обрабатываться интеллектуальными системами и быть представленным в терминах некоторой онтологии, а их структура и

содержание понятны управляющим ими. Из вышесказанного следует, что концептуальные информационные ресурсы, в том числе и базы знаний, имеют чисто декларативное и структурное представление (такой информационный ресурс можно представлять себе в виде семантической сети, структура которой определяется некоторой онтологией, а каждый термин, вводимый в этой сети, относится к некоторому классу, определяемому этой онтологией). Примерами информационных ресурсов, не являющихся концептуальными, могут служить базы правил и системы фреймов. В них объединены два типа информации — о знаниях предметной области и о способе решения задачи. Если количество правил (фреймов) в такой базе велико, то ручное управление ею оказывается практически нереальным. Поскольку носителем знаний предметной области является эксперт, а носителем знаний о способе решения задачи — инженер знаний, то, чтобы обеспечить правильное функционирование интеллектуальной системы, инженер знаний должен взять на себя функции управления такой базой знаний, но при этом эксперт может потерять контроль над ее содержанием.

Поскольку концептуальный информационный ресурс представляется в терминах соответствующей онтологии, он является неразрывной парой, состоящей из этой онтологии и собственно содержания (контента) этого ресурса, которые находятся между собой в определенном соответствии. Онтология информационного ресурса формируется инженером знаний, а его содержание — носителем соответствующей информации. Таким образом, к концептуальной базе знаний должны иметь доступ и инженер знаний (к онтологии), и эксперт (к знаниям), при-

чем для ручного управления этим ресурсом им нужен разный интерфейс.

Теперь перейдем к обсуждению некоторых задач, возникающих при управлении информационными ресурсами.

Мониторинг информационных ресурсов направлен на получение неявной информации, содержащейся в них. В случае, когда мониторинг осуществляется в соответствии с априорными знаниями управляющих, этими знаниями могут быть, например, явные определения структурных моделей этих ресурсов, их структурных свойств в терминах этих моделей, дефектов, недостатков, неудобств и особенностей этих ресурсов в терминах этих свойств (рис. 3.6) [9, 18].

Задачами мониторинга информационных ресурсов в этом случае являются построение на основе этих знаний структурных моделей этих ресурсов, определение значений их структурных свойств, выявление дефектов, недостатков, неудобств и их особенностей (рис. 3.7).

Онтология концептуального информационного ресурса также является априорными знаниями, которые ограничивают управление этим информационным ресурсом — управление им не может нарушать ограничений целостности этого информационного ресурса, явно представленных в его онтологии. Однако информационный ресурс может быть формально неполон относительно своей онтологии — некоторые его элементы, наличие которых предписывается онтологией, на стадии формирования информационного ресурса или его изменения могут отсутствовать. Поэтому задачей мониторинга информационного ресурса в этом случае является получение информации о его формальной полноте относительно его онтологии.

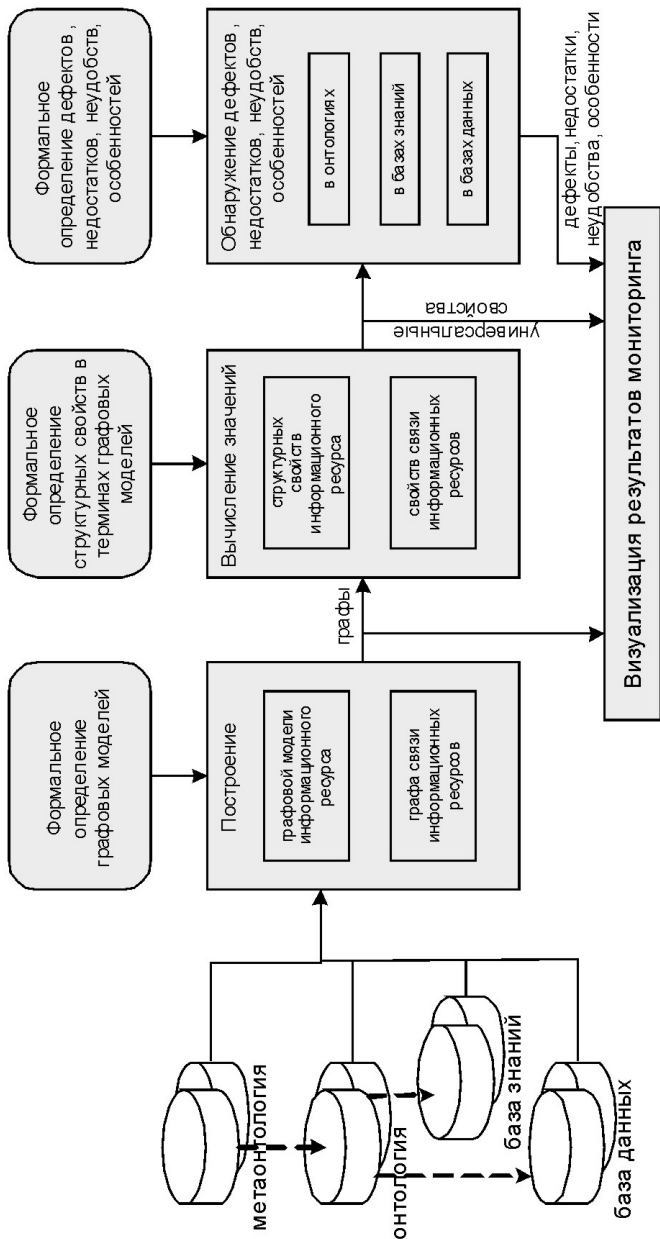


Рис. 3.7. Общая схема подсистемы мониторинга информационных ресурсов

Ручное управление информационными ресурсами может осуществляться на основе результатов их мониторинга. В этом случае задачей ручного управления информационными ресурсами является устранение выявленных в них дефектов, недостатков и неудобств с помощью интерактивных средств управления. Кроме того, ручное управление информационными ресурсами может быть направлено на приведение их содержания в соответствие с представлениями об этом содержании у лиц, управляющих этими информационными ресурсами, в частности, с учетом выявленных особенностей. Типичной задачей этого класса является ручное управление с помощью интерактивных средств управления: базой знаний, осуществляемое экспертом предметной области; базой данных, осуществляемое специалистами предметной области; онтологией (предметной области, базы знаний или базы данных), осуществляемое инженером знаний. На рисунке 3.8 приведена схема работы интерактивного средства ручного управления информационными ресурсами, которое по описанию языка представления онтологий, представленному на этом же языке, в режиме инженера знаний позволяет интерактивно формировать онтологию информационного ресурса на том же языке, а в режиме эксперта — управлять самим этим информационным ресурсом.

Автоматическое управление информационными ресурсами, как и ручное, также может осуществляться на основе результатов их мониторинга. В этом случае задачей автоматического управления информационными ресурсами является автоматическое устранение выявленных в них дефектов, недостатков, неудобств и особенностей.

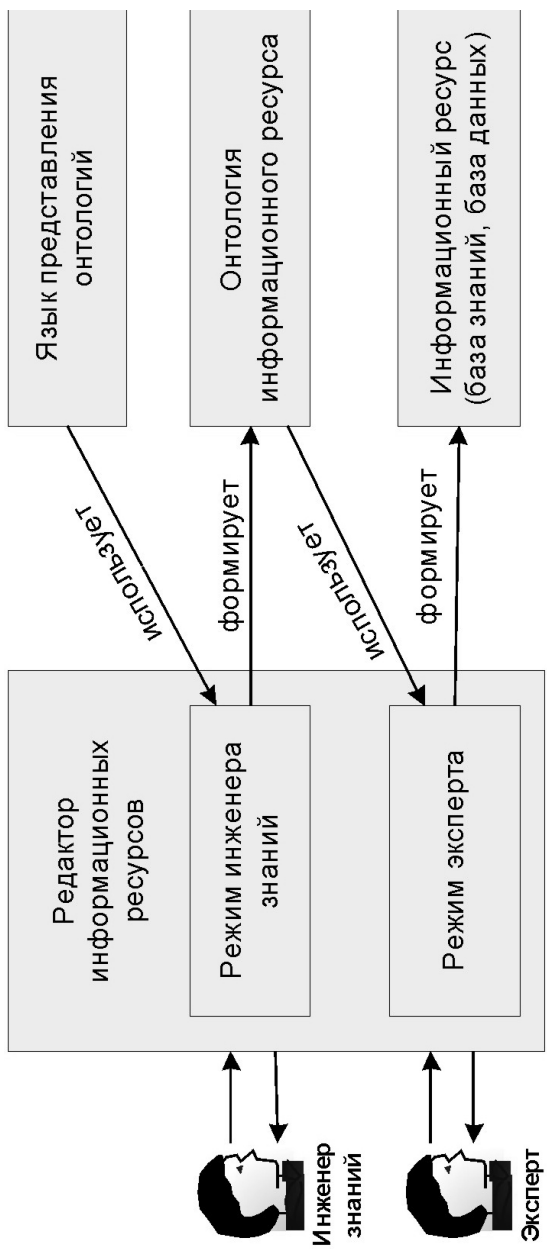


Рис. 3.8. Схема интерактивного средства ручного управления информационными ресурсами

Если же в качестве априорных знаний рассматривается онтология информационного ресурса, то одной из задач автоматического управления становится контроль соответствия между информационным ресурсом и его онтологией. Еще одним классом являются задачи управления взаимосвязанными информационными ресурсами.

Примером может служить управление базой знаний (например, базой заболеваний) и базой данных (например, архивом историй болезни), часть информации в которых представлена в терминах одной и той же онтологии (например, базы наблюдений) (рис. 3.9). Для автоматического контроля целостности взаимосвязей этих трех концептуальных информационных ресурсов (при их автономном ручном управлении) онтология каждого из них должна содержать еще и описание этих связей. Онтология связи базы заболеваний с базой наблюдений позволяет формировать базу заболеваний в терминах базы наблюдений и приводить базу заболеваний в соответствие с базой наблюдений при изменении последней; аналогичную роль играет и онтология связи архива историй болезни с базой наблюдений.

Если некоторая подсистема управления информационными ресурсами является интеллектуальной системой, т. е. системой, основанной на знаниях, то управление ее базой знаний может рассматриваться как частный случай автоматизированного управления информационными ресурсами — ручного управления параметрами этой подсистемы управления информационными ресурсами. Примером может служить задача ручного управления мониторингом информационных ресурсов, в которой инженеры знаний изменяют явно заданные описания структурных моделей информационных ресурсов, определений

их структурных свойств, дефектов, недостатков, неудобств и особенностей.

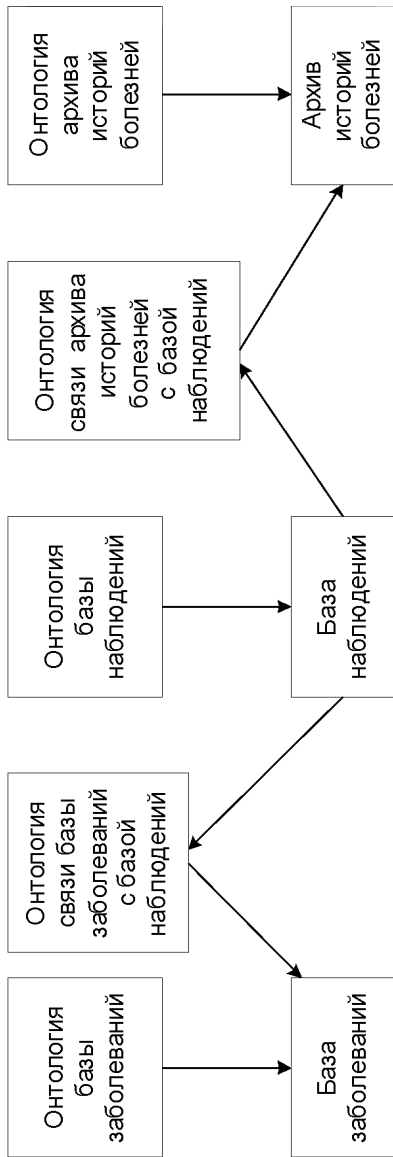


Рис. 3.9. Пример взаимосвязанных информационных ресурсов — базы заболеваний, базы наблюдений и архива историй болезней

Это, в свою очередь, влияет на результаты автоматического мониторинга информационных ресурсов (рис. 3.7). Если инженеры знаний посредством ручного управления изменяют онтологию информационного ресурса, то соответствие между этим ресурсом и его онтологией нарушается. При этом одной из задач автоматизированного управления становится задача косвенного управления этим информационным ресурсом, состоящая в том, чтобы через автоматическое управление восстановить соответствие между этим ресурсом и его изменившейся онтологией, пожертвовав его полнотой. Аналогичная (но более сложная) задача косвенного управления возникает и в случае такого ручного управления одним из взаимосвязанных информационных ресурсов, при котором нарушается целостность взаимосвязей между этими ресурсами (рис. 3.9).

Если знания экспертов не вполне отражают реальное состояние предметной области, ручное управление базой знаний, осуществляемое экспертом, может не дать требуемого эффекта. Такая ситуация имеет место, например, в медицине, где база знаний, сформированная отдельным экспертом, чаще всего имеет невысокие оценки на контрольной выборке историй болезни. В таких ответственных предметных областях, как медицина, базы знаний должны обладать свойством эталонности. Они должны быть понятны экспертам и другим специалистам предметной области, в том числе удовлетворять количественным ограничениям (например, в ряде случаев — известному психологическому критерию 7 ± 2). Базы знаний и онтологии, в терминах которых они представлены, не должны иметь заранее определенных внутренних дефектов; базы знаний должны иметь убедительные и понятные экспертам оценки на больших

выборках примеров и контрпримеров. Эталонные базы знаний должны допускать их использование специалистами в практической деятельности непосредственно и без интеллектуальной системы, как надежный справочник, а их оценки должны служить средством обеспечения доверия к ним. Для достижения эталонности базы знаний управление ею должно осуществляться в соответствии с входным потоком данных.

В этом случае мониторинг должен быть направлен на получение информации о соответствии базы знаний этому входному потоку. Это соответствие определяется онтологией предметной области, которая включает онтологию базы знаний, онтологию действительности (которой соответствуют данные из входного потока) и онтологические соглашения о взаимосвязи знаний и действительности (рис. 3.10).

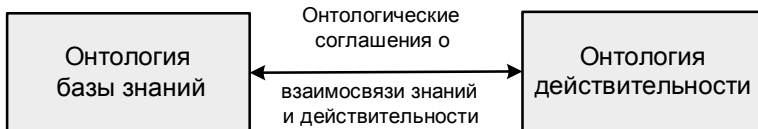


Рис. 3.10. Схема онтологии предметной области

Задачей такого мониторинга является получение информации о том, какие онтологические соглашения нарушены и на каких элементах базы знаний и входного потока (рис. 3.11).

Результаты этого мониторинга могут быть использованы для устранения некоторых ошибок в данных входного потока (ручного управления им), но в ряде случаев использование этих результатов для ручного управления базой знаний может представлять чрезмерную интеллектуальную сложность (например, для экспертной системы медицинской диагностики, где входной

поток состоит из большого числа историй болезни). В других случаях, когда объем базы знаний предполагается большим, а элементы базы знаний слабо связаны друг с другом, ручное управление оказывается возможным. Например, в интерактивной системе доказательства теорем, использующей технологию облачных вычислений (Cloud computing), где входной поток состоит из теорем, уже доказанных пользователями системы, интегратор общей базы знаний может включать в эту базу элементы входного потока из персональных баз знаний исследователей по своему усмотрению (рис. 3.12).

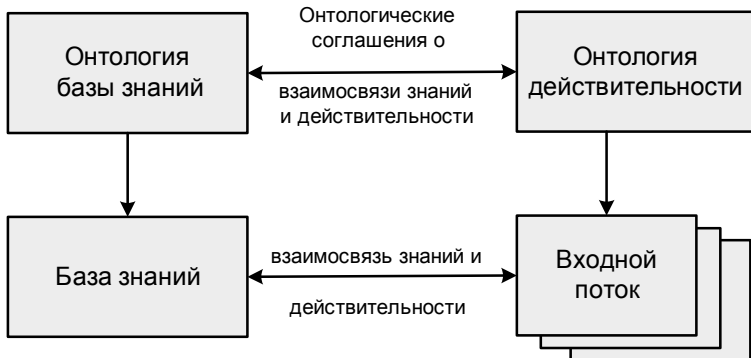


Рис. 3.11. Мониторинг соответствия базы знаний входному потоку

Если ручное управление базой знаний в соответствии с входным потоком оказывается чрезмерно сложным для управленцев, возникают задачи автоматического управления базой знаний — ее индуктивного формирования в соответствии с данными входного потока [10]. Итеративный алгоритм индуктивного формирования концептуальной базы знаний включает два асинхронных процесса — первый из них обрабатывает посту-

пающие один за другим примеры из обучающей выборки и после обработки некоторой порции новых примеров передает результат своей работы (множество возможных баз знаний) второму процессу для обновления эталонной базы знаний (рис. 3.13).

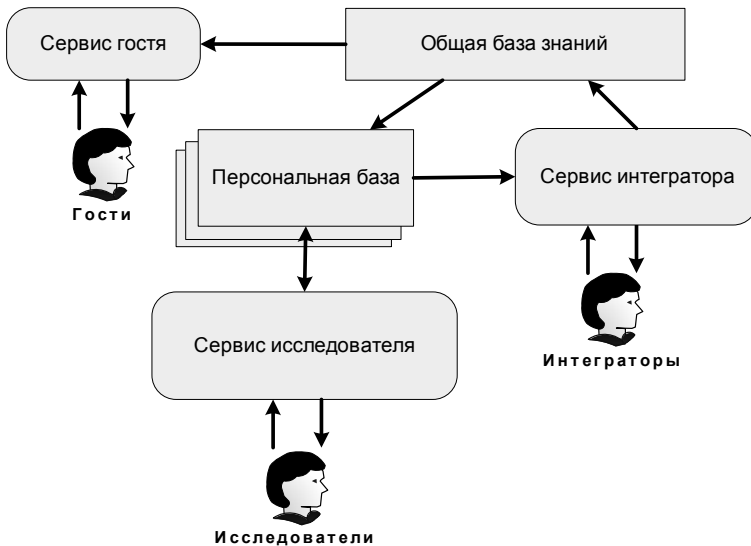


Рис. 3.12. Ручное управление базой знаний интерактивной системы доказательства теорем на основе результатов мониторинга ее входного потока

Процесс обработки новых примеров циклически обрабатывает один пример за другим, формируя варианты обобщения примеров (например, по времени, если предполагается, что данные зависят от времени); эти варианты обобщения используются затем для пересчета множества альтернативных баз знаний. После обработки заданного числа примеров варианты

базы знаний передаются процессу обновления эталонной базы знаний (который тем самым запускается), а процесс обработки новых примеров продолжается (рис. 3.14). Когда запускается процесс обновления эталонной базы знаний, по выборке контр-примеров вычисляются оценки альтернативных баз знаний, выбирается база знаний с наилучшей оценкой, которая и заменяет эталонную оценку, после чего процесс переходит в состояние ожидания нового множества возможных баз знаний (рис. 3.15).

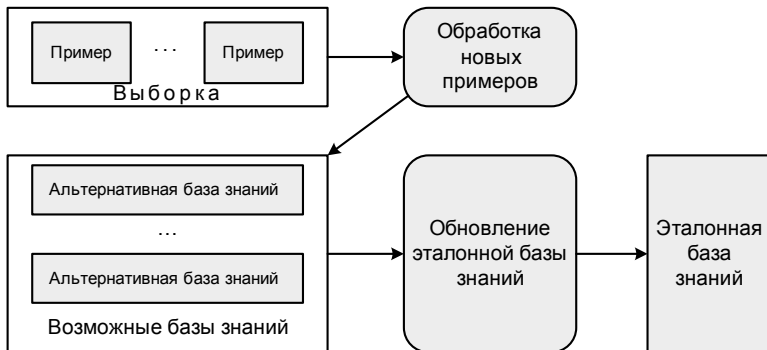


Рис. 3.13. Схема итеративного алгоритма индуктивного формирования концептуальной базы знаний

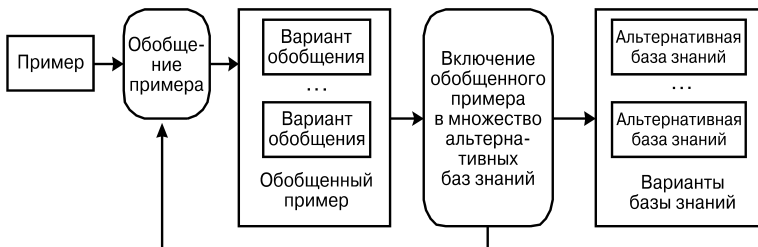


Рис. 3.14. Обработка новых примеров

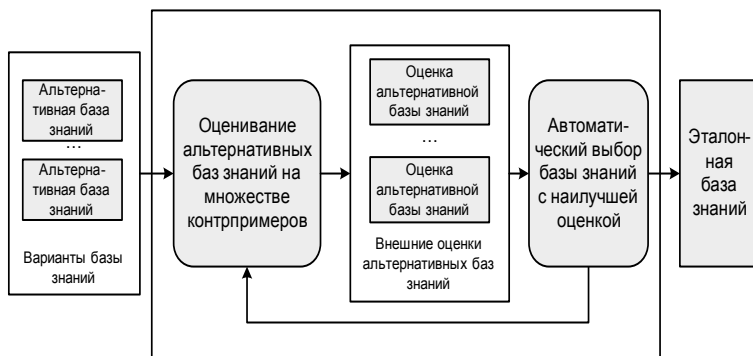


Рис. 3.15. Обновление эталонной базы знаний

Комбинирование возможностей мониторинга, ручного и автоматического управления базой знаний экспертной системы в соответствии с входным потоком данных приводит к задачам автоматизированного управления (человеко-машинной отладки базы знаний), в которых эксперту предоставляются результаты мониторинга базы знаний относительно входного потока. Целью управления является улучшение этих результатов, решение задач автоматического управления базой знаний на основе методов индуктивного формирования предоставляет эксперту возможные варианты изменения базы знаний, выбор из которых он делает по своему усмотрению (рис. 3.16).

Другим примером задачи автоматизированного управления базой знаний, где также осуществляется комбинирование возможностей мониторинга, автоматического и ручного управления в соответствии с входным потоком, является один из видов управления базой знаний интерактивной системы доказательства теорем. Задача мониторинга базы знаний состоит в обобщении каждого элемента входного потока (построенного средствами системы доказательства) до метадоказательства

[12]; задача автоматического управления базой знаний состоит в построении множества теорем, аналогичных ранее доказанным, на основе этих метадоказательств и базы знаний [12]; интегратор общей базы знаний, решая задачу ручного управления, включает некоторые из этих теорем в базу знаний по своему усмотрению (рис. 3.17).

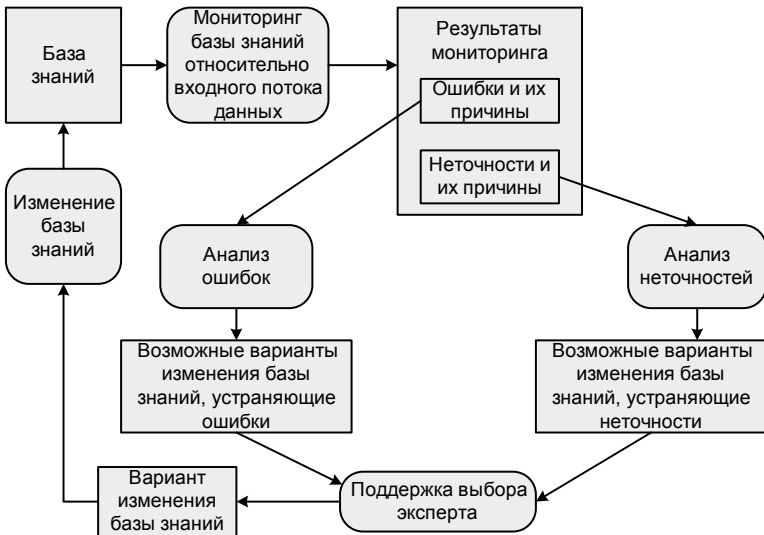


Рис. 3.16. Отладка базы знаний

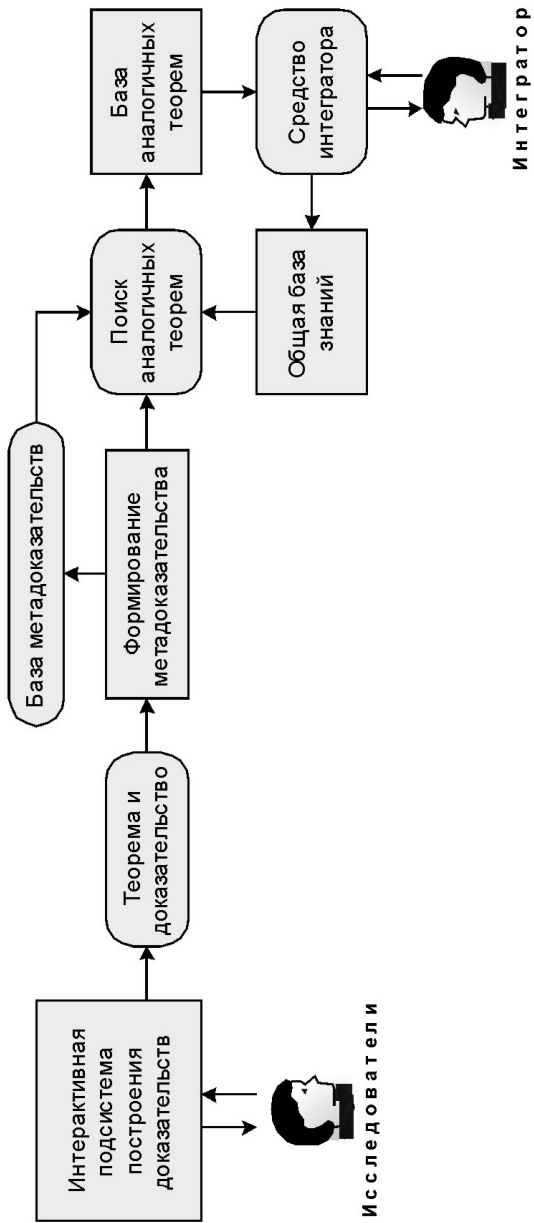


Рис. 3.17. Автоматизированное управление общей базой знаний интерактивной системы доказательства теорем

3.5.2. Управление решателями задач

Целью управления решателями задач является перманентное приведение их свойств в соответствие с постоянно изменяющимися текущими требованиями, условиями эксплуатации, знаниями предметной области, а также возможными изменениями онтологии предметной области.

Если база знаний интеллектуальной системы является концептуальной, т. е. имеет чисто декларативное представление, то использование при реализации этой интеллектуальной системы какой-либо универсальной машины вывода (оболочки) оказывается невозможным. В этом случае самостоятельным компонентом интеллектуальной системы становится решатель задач, т. е. программа, которая, используя базу знаний, решает задачи интеллектуальной системы. Решатель задач является специализированным интерпретатором концептуальных баз знаний, в котором зафиксированы онтология базы знаний (и других обрабатываемых им информационных ресурсов) и алгоритмы решения задач интеллектуальной системы. Для каждой новой интеллектуальной системы, для которой зафиксированы онтология обрабатываемых информационных ресурсов и алгоритмы решения задач, решатель задач должен разрабатываться заново.

В настоящее время разработка и сопровождение программ осуществляются на основе следующей парадигмы. Программа представляется в виде текста (строки символов) на некотором процедурном языке. Синтаксический и семантический анализаторы процессора этого языка формируют на основе этого текста совокупность промежуточных декларативных структур — дерево грамматического разбора программы в абстракт-

ном синтаксисе, ее управляющий и информационный графы и т. п. Генератор кода процессора этого языка на основе этих декларативных структур формирует объектный код программы, который и управляет процессом ее выполнения. Сопровождение программы, т. е. изменение ее свойств, требует изменения текста программы, после чего формирование нового объектного кода выполняется таким же способом по измененному тексту программы. Промежуточные декларативные структуры обычно недоступны для просмотра и тем более изменения. Их анализ и изменение с целью получения нового объектного кода, помимо всего прочего, затруднены тем, что они формируются автоматически, а их связь с текстом программы достаточно нетривиальна.

Управляемость программы в процессе ее жизненного цикла может быть повышена за счет инверсии указанной выше парадигмы, а именно за счет представления программы в виде декларативной структуры данных (семантической сети), с терминальными вершинами которой связаны модули, содержащие программный код, т. е. за счет разделения декларативного и процедурного компонентов. В этом случае декларативный компонент программы становится объектом мониторинга и ручного управления, а процедурный — объектом сопровождения и автоматического управления. Достижение управляемости программы в процессе ее жизненного цикла состоит в сведении к некоему разумному минимуму ее процедурного компонента за счет представления как можно большей информации в ее декларативном компоненте. В последнее десятилетие близкие идеи используются в области интеграции отдельных программ в

программные системы с использованием декларативного представления схемы интеграции [2, 14, 15].

Управляемость программы в процессе ее функционирования может быть повышена за счет подключения к решателю задач модулей, решающих задачи управления в процессе его функционирования. Таким способом может осуществляться мониторинг потока данных, т. е. извлечение из него неявной информации, и основанное на его результатах автоматическое управление решателем задач.

Мониторинг решателя задач, выполняемый в соответствии с априорными знаниями управленцев, направлен на получение неявной информации о декларативном компоненте этого решателя. Примерами таких задач мониторинга являются оценивание программ и поиск в них дефектов в соответствии с заданными метриками и определениями дефектов.

Примерами мониторинга решателя задач, выполняемого в соответствии с входным потоком данных, является получение информации о свойствах схемы его распараллеливания при работе некоторого решателя задачи на многопроцессорной вычислительной системе, а также перманентная коррекция списка уже доказанных утверждений или базы метадоказательств в процессе работы интерактивной системы доказательства теорем (рис. 3.17) [11, 12].

Ручное управление решателем задач направлено на изменение его декларативного компонента лицами, управляющими этим решателем, с учетом результатов его мониторинга. Примерами могут служить задачи устранения с помощью интерактивных средств выявленных дефектов в декларативной компоненте решателя, задачи изменения схемы распараллелива-

ния решателя задач с учетом информации о степени его реального параллелизма при работе на различных данных и т. п.

Автоматическое управление решателем задач направлено на автоматическое (без участия человека) изменение его свойств с учетом результатов мониторинга обрабатываемых им информационных ресурсов либо результатов мониторинга самого решателя задач (адаптацию к результатам мониторинга). Примером адаптации решателя задач к результатам мониторинга базы знаний экспертной системы медицинской диагностики является задача сужения множества гипотез о диагнозе на основе совокупности наблюдаемых значений признаков пациента без детального анализа их временной структуры. При этом для каждого значения признака вычисляется множество заболеваний, при которых возможно это значение (рис. 3.18).

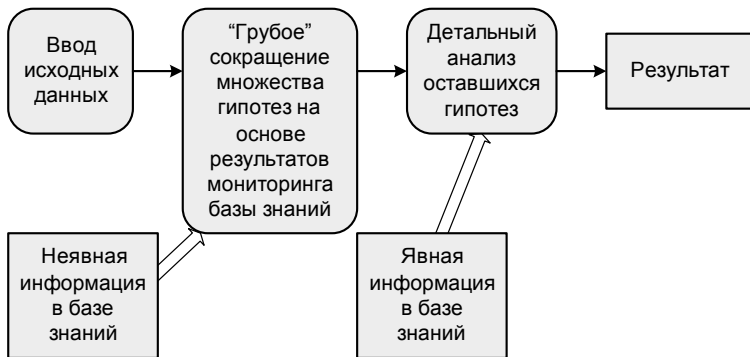


Рис. 3.18. Адаптация решателя задач к результатам мониторинга базы знаний экспертной системы

Примером адаптации решателя задач к результатам его потокового анализа является задача оптимизации программ. Примерами адаптации решателя задач к результатам его мони-

торинга, осуществляемого в соответствии с входным потоком данных, для интерактивной системы доказательства теорем является исключение повторного доказательства одних и тех же целей на основе анализа списка уже доказанных утверждений или исключение необходимости интерактивного построения доказательства цели в случае, если доказательство этой цели может быть получено по аналогии с использованием базы метадоказательств (рис. 3.19).

Если некоторая подсистема управления решателем задач является интеллектуальной системой, то управление ее базой знаний может рассматриваться как задача автоматизированного управления информационными ресурсами. Примером задач автоматизированного управления, ориентированного на априорные знания управленцев, является ручное управление базами знаний подсистем мониторинга решателя задач и оптимизации программ. Расширение функциональности решателя задач интерактивной системы доказательства теорем на случай построения интуитивных (неполных) доказательств требует включения в решатель механизмов автоматического управления, обеспечивающих проверку правильности таких доказательств и ориентированных на входной поток данных. Однако такие механизмы обладают свойством алгоритмической полуразрешимости. Естественным выходом из этой ситуации является переход к задаче автоматизированного управления решателем задач, когда неудачи механизма автоматического управления исправляются через механизм ручного управления (рис. 3.20).

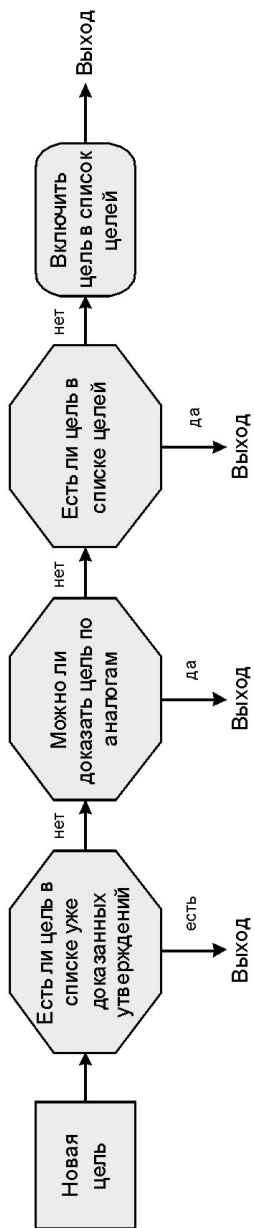


Рис. 3.19. Включение новой цели в список целей интерактивной системы доказательства теорем

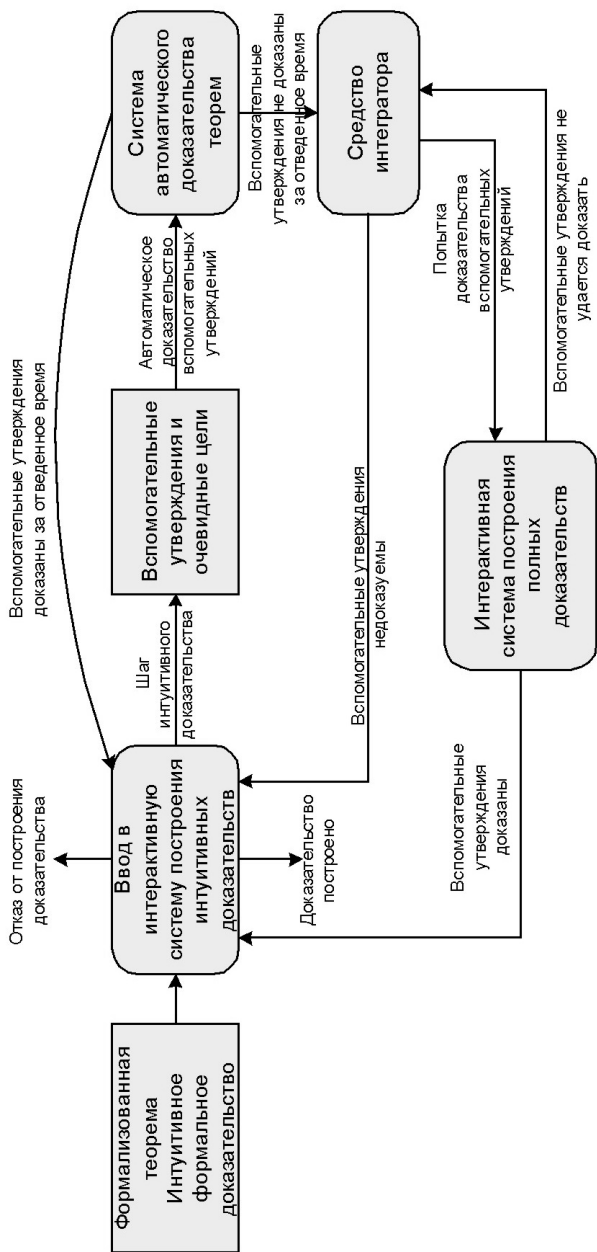


Рис. 3.20. Автоматизированное управление интерактивной системой построения правильных интуитивных доказательств

3.5.3. Управление пользовательским интерфейсом

Целью управления пользовательским интерфейсом является перманентное приведение его свойств в соответствие с постоянно изменяющимися онтологией предметной области, текущими требованиями и условиями эксплуатации.

Управляемость в процессе жизненного цикла пользовательского интерфейса может быть достигнута, если его проект представлять декларативно в виде концептуального информационного ресурса, а функционирующий интерфейс получать из этого представления проекта с помощью генератора кода или интерпретатора проекта [5, 6]. В этом случае проект становится объектом управления в процессе жизненного цикла. Управляемость в процессе функционирования пользовательского интерфейса может быть достигнута, если к коду интерфейса или к интерпретатору проекта подключать модули, решающие задачи управления в процессе функционирования. Таким способом может осуществляться мониторинг действий пользователя и основанное на его результатах автоматическое управление пользовательским интерфейсом.

Мониторинг пользовательского интерфейса направлен на получение неявной информации о нем (рис. 3.21).

Примерами задач мониторинга, выполняемого в соответствии с априорными знаниями управленцев, является выявление дефектов в проекте интерфейса в соответствии с метриками юзабилити. На рисунке 3.22 представлена архитектура прототипа системы управления мониторингом проекта интерфейса, разработанного в лаборатории [3, 4]. Примерами мониторинга, выполняемого в соответствии с входным потоком данных, является мониторинг данных, поступающих от решателя задач

пользовательскому интерфейсу и обратно (объем и характер данных). Примерами задач мониторинга, выполняемого в соответствии с моделью пользователя, является мониторинг действий пользователя при работе с интеллектуальной системой (время, затраченное на выполнение задачи, число ошибок, возвратов, обращений к справочной системе и т. п.).

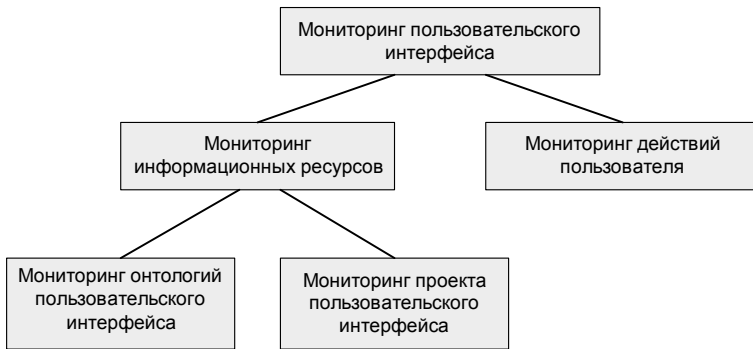


Рис. 3.21. Мониторинг пользовательского интерфейса

Ручное управление пользовательским интерфейсом направлено на изменение его свойств лицами, управляющими этим интерфейсом, с учетом результатов его мониторинга (рис. 3.23). Примерами могут служить интерактивные задачи устранения дефектов в проекте интерфейса в соответствии с метриками юзабилити или нарушений целостности проекта, изменения состава интерфейсных задач и отношений между ними, управления характеристиками пользователей, управления представлением информации в интерфейсе и управления системами понятий диалога.

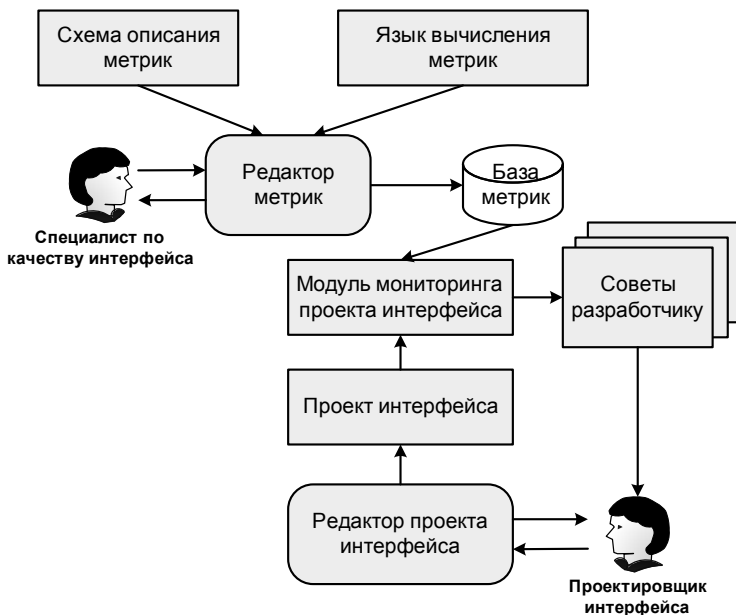


Рис. 3.22. Система управления мониторингом проекта интерфейса

Задачи автоматического управления пользовательским интерфейсом состоят в фиксированной адаптации интерфейса, когда задачи автоматического управления жестко связаны с типами пользователей; косметической адаптации, например опережающем вводе символов или ответов, использовании сокращений, умолчаний, многоуровневой помощи, многоязыковости и т. п.; адаптации, основанной на модели пользователя и учитывающей его статические и динамические характеристики, такие как возраст, количество ошибок, время ответа и т. п. Кроме того, если некоторые характеристики пользовательского интерфейса неизвестны на этапе его проектирования, то возникают задачи автоматического управления проектом интерфейса —

адаптации к объемам и структуре данных, циркулирующих между решателем задач и интерфейсом, формировании динамического сценария диалога на основе данных решателя (например, сценария запроса дополнительной информации о больном для дифференциальной диагностики) или выбора пользователя (сценария интерактивной системы доказательства теорем).

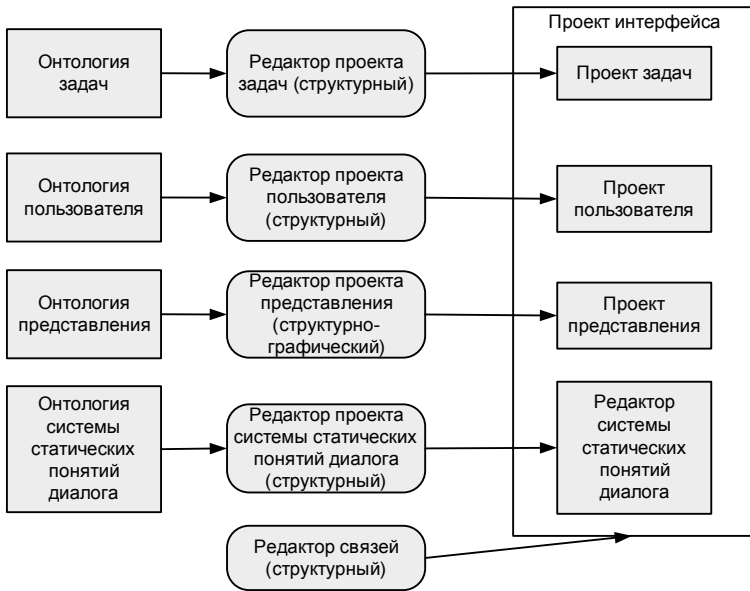


Рис. 3.23. Система ручного управления пользовательским интерфейсом

К задачам автоматизированного управления пользовательским интерфейсом относится управление его мониторингом, которое состоит в управлении базой метрик юзабилити, модификации советов разработчику по устранению дефектов, в задании групп метрик юзабилити, по которым мониторинг дол-

жен проводиться, а также в управлении параметрами внутренних механизмов адаптации к особенностям конкретных пользователей (автозаполнение текстовых полей, двухуровневые подсказки, сокрытие элементов главного меню и др.).

3.6. МЕТОДЫ НАУЧНЫХ ИССЛЕДОВАНИЙ ПО ТЕОРИИ, ТЕХНОЛОГИИ РАЗРАБОТКИ И ЭКСПЛУАТАЦИИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Искусственный интеллект в настоящее время применяется во многих областях. В последние годы современные информационные технологии совершили резкий скачок вперед в основном за счет повышения производительности массовых процессоров и удешевления памяти ЭВМ. Это привело к появлению приложений, в которых воплотились серьезные теоретические наработки по искусственному интеллекту.

Основной проблемой исследований в области искусственного интеллекта является построение машинной модели, которая бы производила сложные преобразования информации, осуществляемые человеческим мозгом, включая, в частности, зрительное распознавание пространственных сцен, общение на естественном языке, в том числе в форме речи, обучение на опыте, выработку новых понятий, открытие новых свойств и законов, постановку новых задач и нахождение алгоритмов их решения, разработку новых научных теорий и т. д.

Идея практического применения исследований в области искусственного интеллекта в виде экспертных систем заключается в следующем. Если пока не удастся заставить машину гибко реагировать на проблемные области, самой вырабатывать нужные методы поиска, находить существенно новые свойства и законы, вырабатывать новые знания, приобретать новый опыт

в изучаемой ею проблемной области, то можно воспользоваться накопленным человеческим опытом, готовыми знаниями, методами, навыками решения задач в некоторой предметной области и заложить их в машину (в ее базу знаний). Тем самым будет на время снята проблема накопления машиной опыта, открытия ею новых знаний, а останется проблема применения уже накопленного специалистами опыта для вывода знаний с помощью имеющихся средств.

Затем необходимо разработать программу применения этого опыта для решения тех задач, с которыми справляется специалист и при решении которых он не располагает строгими математическими алгоритмами в силу неформализованности соответствующих знаний, отсутствия точных математических моделей. Речь идет о том опыте, который специалист может выразить словами в терминах данной предметной области, в виде либо некоторых общих высказываний и правил, либо описания конкретных примеров, образцов решений и действий в различных конкретных ситуациях. Такие знания называются вербализуемыми. Но у человека вырабатывается и другой опыт, не описываемый терминами исследуемой предметной области. Этот опыт представляется в некоторой системе формирующихся у человека связей, образов, интуитивных предчувствий, предвидений, предпочтений, неосознаваемых реакций и т. п. Он не сформирован в четко осознаваемые человеком правила, связи, принципы, эмпирические законы.

По-видимому, описание подсознательного опыта следует проводить в другом языке — не в терминах внешнего поведения человека при обработке им информации, а в терминах нейронных структур человеческого мозга и их связей, обеспечивающих

самоорганизацию и специализацию поисковых механизмов. Поэтому предметная область для экспертных систем должна быть такой, чтобы опыт, который не удается вербализовать, не играл главенствующую роль при решении задач, как, например, в задачах оценки произведений искусства, в процессах художественного творчества, дегустации и т. п.

На этапе создания экспертных систем первого поколения в них применялись наиболее проработанные фрагменты еще далеких от завершения исследований в области искусственного интеллекта. При этом из-за недостаточности научных знаний о том, как заставить машину приобретать знания и опыт, использовался накопленный человечеством научный потенциал и практический опыт; из-за недостаточности научных знаний о том, как передать машине ту часть человеческого опыта, которая не поддается словесным описаниям, пришлось передавать машине только опыт, поддающийся вербализации. Наконец, из вербализуемых знаний использовались в основном только так называемые поверхностные, эмпирические знания, получаемые в результате обобщения внешнего поведения исследуемых объектов, без учета их внутренней природы, внутренних законов функционирования, глубоких причинно-следственных связей. Представление же глубинных знаний, а также приведение индуктивных выводов, обучение на опыте, открытие новых свойств, законов и другие сложные интеллектуальные действия включаются в разработку экспертных систем второго и последующих поколений. Тем не менее уже разработанные экспертные системы находят применение в самых разнообразных областях науки, техники, производства, культуры.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие принципиальные особенности отличают новую информационную технологию от существующей?
2. На какие типы подразделяются в настоящее время системы искусственного интеллекта, функционирующие на принципах новой информационной технологии?
3. Развитие каких основных проблем предопределяет дальнейший прогресс систем искусственного интеллекта и новой информационной технологии в связи с тем, что все системы искусственного интеллекта ориентированы на знания?
4. Какие основные пути использования психологических знаний в практике автоматизации умственного труда необходимо наметить в связи с развитием исследований и разработок систем ИИ?
5. Мультиагентные технологии. Понятие агента и его возможная реализация.

4. ПРИМЕНЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ В ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

4.1. ОРГАНИЗАЦИЯ ДИАЛОГА МЕЖДУ ЧЕЛОВЕКОМ И ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМОЙ

4.1.1. Диалоговые системы, основанные на распознавании рукописного текста

Рассмотрим, в чем заключается различие между двумя формами представления одного и того же текста: рукописной и печатной. При этом могут исследоваться и сравниваться как сам процесс формирования текста, так и его результаты, т. е. уже сформированные тексты.

При исследовании уже сформированных текстов обнаруживается, что главное отличие рукописного текста от печатного состоит в значительно большей степени вариабельности начертаний одной и той же буквы разными людьми и одним и тем же человеком в различных состояниях, чем при воспроизведении тех же букв на различных пишущих машинках и принтерах.

Почерком будем называть систему индивидуальных особенностей начертания и динамики воспроизведения букв, слов и предложений вручную различными людьми или на различных устройствах печати.

В рукописной форме начертание букв является индивидуальным для каждого человека и зависит также от его состояния, хотя, конечно, в начертаниях каждой конкретной буквы всеми людьми, безусловно, есть и нечто общее, что и позволяет идентифицировать ее именно как данную букву при чтении.

К индивидуальным особенностям рукописного начертания букв отнесены 13 шкал с десятками градаций в каждой.

На современных компьютерах основным устройством ввода текстовой информации является клавиатура. Результат ввода текста в компьютер с точки зрения начертания букв, слов и предложений не имеет особых индивидуальных особенностей (если не считать частот использования различных шрифтов, кеглей, жирностей, подчеркиваний и других эффектов, изменяющих вид текста). Поэтому необходимо ввести понятие ***клавиатурного почерка***, под которым будем понимать ***систему индивидуальных особенностей начертаний и динамики воспроизведения букв, слов и предложений на клавиатуре***.

Таким образом, ***любой текст содержит не только ту информацию, для передачи которой его собственно и создавали, но и информацию о самом авторе этого текста и о технических средствах и технологии его создания***.

Существует целая наука — ***психографология***, которая ставит своей задачей получение максимального количества информации об авторах текстов на основе изучения индивидуальных особенностей их почерка.

В настоящее время в России действует институт графологии. На сайте этого института <http://graphology.boom.ru> можно познакомиться с тем, что такое графология, с ее историей и задачами, которые она позволяет решать сегодня. Графологическое исследование имеет значительное преимущество перед простым тестированием или собеседованием, поскольку нет необходимости информировать человека, чей почерк подвергается изучению, о производимых исследованиях.

Но текст представляет собой не просто совокупность букв, а сложную иерархическую структуру, в которой буквы образуют лишь фундамент пирамиды, а на более высоких ее уровнях находятся слова, предложения и другие части текста разных размеров, обладающие относительной целостностью и самостоятельностью (абзацы, параграфы, главы, части, книги).

Понятие почерка акцентирует внимание именно на начертании и динамике воспроизведения **букв** и **слов**. При этом в понятие почерка не входят индивидуальные особенности текстов, обнаруживаемые на более высоких уровнях иерархической организации текстов, например частота употребления тех или иных слов и словосочетаний, средняя длина предложений и абзацев и т. п. Но именно эти индивидуальные особенности текстов исследуются и используются при **атрибуции** анонимных и псевдонимных текстов (определении их вероятного авторства) и датировки.

Соответственно и текст может представлять для читателя интерес по крайней мере с трех точек зрения:

- 1) как источник информации о том, о чем говорит автор, т. е. о предмете изложения;
- 2) как источник информации о самом авторе;
- 3) как источник информации о предмете изложения и об авторе.

В этом смысле читать Пушкина в рукописи может быть значительно интереснее, чем взяв томик с полки. Это объясняется просто: в томике есть лишь сам **результат** работы поэта и выхолощена вся информация о **процессе**, т. е. о самом поэте, содержащаяся в почерке, способе размещения текста на листе,

порядке и динамике его формирования, различных вариантах и ассоциациях, возникших в процессе создания произведения.

Таким образом, система, оснащенная интеллектуальным интерфейсом, может вести себя по-разному в зависимости от результатов идентификации пользователя, его профессионального уровня и текущего психофизиологического состояния.

Рассмотрим подробнее некоторые вопросы идентификации пользователей по клавиатурному почерку.

Проблемы идентификации и аутентификации пользователей компьютеров являются актуальными в связи со все большим распространением компьютерных преступлений. Использование для идентификации клавиатурного почерка является одним из направлений биометрических методов идентификации личности.

Подобные системы не обеспечивают такую же точность распознавания, как системы идентификации по отпечаткам пальцев или по рисунку радужной оболочки глаз, но имеют то преимущество, что система может быть полностью скрыта от пользователя, т. е. он может даже не подозревать о наличии такой системы контроля доступа.

Аутентификация — это проверка, действительно ли пользователь является тем, за кого себя выдает. При этом пользователь должен предварительно сообщить о себе идентификационную информацию: свое имя и пароль, соответствующий названному имени.

Идентификация — это установление его личности.

И идентификация, и аутентификация являются типичными задачами распознавания образов, которое может проводиться

по заранее определенной или произвольной последовательности нажатий клавиш.

При вводе информации пользователь последовательно нажимает и отпускает клавиши, соответствующие вводимому тексту. При этом для каждой нажимаемой клавиши можно фиксировать моменты нажатия и отпускания.

На IBM-совместимых персональных компьютерах на следующую клавишу можно нажимать до отпускания предыдущих, т. е. символ помещается в буфер клавиатуры только по нажатию клавиши, тогда как аппаратные прерывания от клавиатуры возникают и при нажатии, и при отпускании клавиши.

Основной характеристикой клавиатурного почерка следует считать временные интервалы между различными моментами ввода текста:

- между нажатиями клавиш;
- между отпусканиями клавиш;
- между нажатием и отпусканием одной клавиши;
- между отпусканием предыдущей и нажатием следующей клавиши.

Кроме того, могут учитываться производные от временных интервалов *вторичные показатели*, например такие, как скорость и ускорение ввода.

В литературе описаны четыре математических подхода к решению задачи распознавания клавиатурного почерка пользователя ЭВМ:

- статистический;
- вероятностно-статистический;
- на базе теории распознавания образов и нечеткой логики;
- на основе нейросетевых алгоритмов.

За более подробным анализом подходов к использованию компьютерного почерка для идентификации и аутентификации пользователей компьютеров отсылаем к вышеназванной статье.

В настоящее время Институтом психологии РАН, Институтом высшей нервной деятельности и нейрофизиологии РАН, Высшей школой экономики и Кубанским государственным аграрным университетом (Т. Н. Шукин, В. Б. Дорохов, А. Н. Лебедев, Е. В. Луценко) проводятся исследования, продемонстрировавшие принципиальную возможность прогнозирования ошибок оператора при работе с клавиатурой типа "ошибочное нажатие клавиши", "ошибочное ненажатие клавиши" и т. п. по изменениям в его электроэнцефалограмме (ЭЭГ). При этом для обработки информации успешно была применена система "Эйдос" [108, 224–226].

Эти работы в перспективе позволяют создать интеллектуальные **высоконадежные** интерфейсы, обеспечивающие решение этих и ряда других задач идентификации и прогнозирования состояния оператора **в режиме реального времени непосредственно в процессе его работы с системой**. При этом система в своей работе будет гибко учитывать текущее и прогнозируемое состояние оператора, что может проявляться в адаптации как алгоритмов работы, так и вида и содержания интерфейса.

Эти работы дополняют возможности **заблаговременного** отбора операторов, обладающих свойствами, необходимыми для высоко ответственных работ в экстремальных ситуациях.

4.1.2. Диалоговые системы, основанные на распознавании речи

Распознавание речи — процесс преобразования речевого сигнала в цифровую информацию (например, текстовые данные). Обратной задачей является синтез речи.

Можно выделить следующие области применения:

- голосовое управление;
- голосовой набор в различной технике (мобильники, компьютеры и пр.);
- голосовой ввод текстовых сообщений в смартфонах и прочих мобильных компьютерах;
- голосовой поиск;
- голосовая почта.

Первое устройство для распознавания речи появилось в 1952 году, оно могло распознавать произнесённые человеком цифры. В 1964 году на ярмарке компьютерных технологий в Нью-Йорке было представлено устройство IBM Shoebox.

Коммерческие программы по распознаванию речи появились в начале 1990-х годов. Обычно их используют люди, которые из-за травмы руки не в состоянии набирать большое количество текста. Эти программы (например, Dragon NaturallySpeaking, VoiceNavigator) переводят голос пользователя в текст, таким образом, разгружая его руки. Надёжность перевода у таких программ не очень высока, но с годами она постепенно улучшается.

Увеличение вычислительных мощностей мобильных устройств позволило и для них создать программы с функцией

распознавания речи. Среди таких программ стоит отметить приложение Microsoft Voice Command, которое позволяет работать со многими приложениями при помощи голоса. Например, можно включить воспроизведение музыки в плеере или создать новый документ.

Интеллектуальные речевые решения, позволяющие автоматически синтезировать и распознавать человеческую речь, являются следующей ступенью развития интерактивных голосовых систем (IVR). Использование интерактивного телефонного приложения в настоящее время не веяние моды, а жизненная необходимость. Снижение нагрузки на операторов контакт-центров и секретарей, сокращение расходов на оплату труда и повышение производительности систем обслуживания — вот только некоторые преимущества, доказывающие целесообразность подобных решений.

Прогресс, однако, не стоит на месте, и в последнее время в телефонных интерактивных приложениях все чаще стали использоваться системы автоматического распознавания и синтеза речи. В этом случае общение с голосовым порталом становится более естественным, так как выбор в нем может быть осуществлен не только с помощью тонового набора, но и с помощью голосовых команд. При этом системы распознавания являются независимыми от дикторов, т. е. распознают голос любого человека.

Следующим шагом технологий распознавания речи можно считать развитие так называемых Silent Speech Interfaces (SSI) (интерфейсов безмолвного доступа). Эти системы обработки речи базируются на получении и обработке речевых сигналов на ранней стадии артикулирования. Данный этап развития рас-

познавания речи вызван двумя существенными недостатками современных систем распознавания: чрезмерная чувствительность к шумам, а также необходимость четкой и ясной речи при обращении к системе распознавания. Подход, основанный на SSI, заключается в том, чтобы использовать новые сенсоры, не подверженные влиянию шумов в качестве дополнения к обработанным акустическим сигналам.

На сегодняшний день существуют два типа систем распознавания речи — работающие "на клиенте" (client-based) и по принципу "клиент — сервер" (client — server). При использовании клиент-серверной технологии речевая команда вводится на устройстве пользователя и через Интернет передается на удаленный сервер, где обрабатывается и возвращается на устройство в виде команды (Google Voice, Vlingo и пр.); ввиду большого количества пользователей сервера система распознавания получает большую базу для обучения. Первый вариант работает на иных математических алгоритмах и встречается редко (Speereo Software) — в этом случае команда вводится на устройстве пользователя и обрабатывается в нем же. Плюс обработки "на клиенте" — в мобильности, независимости от наличия связи и работы удаленного оборудования. Так, система, работающая "на клиенте", кажется надежнее, но порой ограничивается мощностью устройства на стороне пользователя.

Сейчас применяется также технология SIND (без привязки к голосу конкретного человека).

Широкое распространение получили системы распознавания голоса для мобильных телефонов: GOOGLE SEARCH для ANDROID и SIRI для iOS5.

Основным преимуществом голосовых систем является дружелюбность к пользователю — он избавляется от необходимости продирааться сквозь сложные и запутанные лабиринты голосовых меню. Теперь достаточно произнесения цели звонка, после чего голосовая система автоматически переместит звонящего в нужный пункт меню.

4.1.3. Системы с биологической обратной связью

Системами с биологической обратной связью (БОС) будем называть системы, поведение которых зависит от психофизиологического (биологического) состояния пользователя.

Это означает, что в состав систем с БОС в качестве подсистем входят информационно-измерительные системы и системы искусственного интеллекта.

Съем информации о состоянии пользователя осуществляется с помощью контактных и/или дистанционных датчиков в режиме реального времени с применением транспьютерных или обычных карт (плат) с аналого-цифровыми преобразователями (АЦП).

*При этом информация может сниматься по большому количеству каналов — показателей (количество которых обычно кратно степеням двойки), подавляющее большинство которых обычно являются **несознаваемыми** для пользователя.* Это является весьма существенным обстоятельством, так как означает, что системы БОС позволяют вывести на уровень сознания обычно ранее неосознаваемую информацию о состоянии своего организма, т. е. расширить область осознаваемого. А это значит, что у человека появляются условия,

обеспечивающие возможность сознательного управления своими состояниями, ранее неуправляемыми на сознательном уровне, что является важным эволюционным достижением технократической цивилизации.

Передача информации от блока съема информации к АЦП-карте может также осуществляться либо по проводной связи, либо дистанционно с использованием каналов инфракрасной или радиосвязи.

Приведем три примера применения подобных систем.

1. Мониторинг состояния сотрудников на конвейере с целью обеспечения высокого качества продукции.

2. Компьютерные тренажеры, основанные на БОС, для обучения больных с функциональными нарушениями управлению своим состоянием.

3. Компьютерные игры с БОС.

Известно, что одной из основных причин производственного брака является ухудшение состояния сотрудников. Но сотрудники не всегда могут вовремя заметить это ухудшение, так как самооценка (самочувствие) обычно запаздывает по времени за моментом объективного ухудшения состояния. Поэтому является актуальным своевременное обнаружение объективного ухудшения параметров и адекватное реагирование на него.

С помощью систем БОС это достигается тем, что:

1) каждому сотруднику одевается на руку браслет с компактным устройством диагностики ряда параметров, например таких, как:

- частота и наполнение пульса;
- кожно-гальваническая реакция;
- температура;

- давление;
- пототделение;

2) это же устройство и периодически передает значения данных параметров на компьютер по радиоканалу;

3) параметры от каждого сотрудника накапливаются в базе данных системы мониторинга на сервере, а также анализируются в режиме реального времени с учетом текущего состояния и динамики, в том числе вторичных (расчетных) показателей;

4) когда параметры выходят за пределы коридора "нормы" или по их совокупности может быть поставлен диагноз — сотрудник оперативно снимается с рабочего места и заменяется другим из резерва, а затем при наличии показаний направляется на лечение.

Некоторыми процессами в своем организме мы не можем управлять не потому, что у нас нет рычагов управления, а лишь потому, что мы их не знаем, не имеем навыков их использования и **не знаем результатов их применения**. Но ключевой проблемой, без решения которой невозможно управление, является отсутствие быстрого и надежного, адекватного по содержанию канала обратной связи.

Все эти проблемы снимаются системами БОС:

- на экран компьютера в наглядной и легко интерпретируемой форме в режиме реального времени выводится информация о состоянии какой-либо подсистемы организма, например об уровне рН (кислотности) в желудке;
- в качестве рычагов управления пациенту предлагается применить метод визуализации тех или иных образов, которые сообщаются врачом;

- когда пациент ярко зрительно представляет заданные образы, то при этом он обнаруживает, что кривая кислотности на экране начинает ползти вверх или вниз в прямом соответствии с тем, что именно он себе представляет.

Через пару недель подобных тренировок, проводимых по 15–20 минут через день пациент приобретает такой уровень навыков управления ранее неосознаваемыми процессами в своем организме, которых хатха-йоги добиваются за многие годы упорных тренировок под руководством профессиональных опытных и ответственных наставников (гуру). Причем скоро пациент начинает понимать, когда необходимо повысить или понизить кислотность и без компьютера с системой БОС и может делать это прямо в той обстановке, в которой возникла такая необходимость. Столь высокая эффективность метода БОС объясняется высокой скоростью, наглядностью и адекватностью обратной связи, что является одним из основных факторов, влияющих на эффективность формирования навыков управления своим состоянием.

Имеется информация, что такими методами могут лечиться или облегчаться многие заболевания, вплоть до диабета, причем не только на стадии функциональных нарушений, но даже и при наличии органических изменений.

В последнее время появляется все больше компьютерных игр, включающих элементы БОС. При этом от психофизиологического состояния игрока может зависеть, например, и развитие сценария, и точность прицеливания при использовании оптического прицела.

В этих играх часто моделируются ситуации, в которых человеку нужно быстро принимать и реализовывать решения, при

этом цена ошибки, а значит, и психическая напряженность, и волнение игрока постоянно увеличиваются. Этим самым создается экстремальная ситуация, напряженность которой все больше возрастает. В этих условиях лучших результатов достигает тот, у кого крепче нервы, кто лучше может управлять собой в экстремальных ситуациях.

Поэтому игры с элементами БОС можно считать своего рода тренажерами по формированию и совершенствованию навыков адекватного поведения в экстремальных ситуациях.

Здесь необходимо отметить один очень существенный момент. В обычной реальности развитие событий зависит не непосредственно от нашего психофизиологического состояния, а лишь от того, как оно проявляется в наших **действиях**. В случае же виртуальной реальности развитие сценария игры может зависеть **непосредственно** от состояния игрока. Таким образом, *в виртуальной реальности само сознательное (произвольное) или несознательное (непроизвольное) изменение нашего состояния по сути дела является действием*. Аналогичная ситуация в обычной реальности может иметь место при высших формах сознания и проявлении сверхспособностей.

4.1.4. Системы с семантическим резонансом. Компьютерные ПС-технологии и интеллектуальный подсознательный интерфейс

Системами с семантическим резонансом будем называть системы, поведение которых зависит от состояния сознания пользователя и его психологической реакции на смысловые стимулы.

Это означает, что в состав систем с семантическим резонансом, так же как и систем с БОС, в качестве подсистем входят

информационно-измерительные системы и системы искусственного интеллекта, аналогично осуществляется и съём информации о состоянии пользователя.

Различие между системами с БОС и с семантическим резонансом состоит в том, что в первом случае набор снимаемых параметров и методы их математической обработки определяются необходимостью идентификации биологического состояния пользователя, тогда как во втором — его реакции на смысловые стимулы (раздражители).

В частности, имеется возможность по наличию в электроэнцефалограмме так называемых вызванных потенциалов установить реакцию человека на стимул: заинтересовался он или нет.

Здесь принципиально важно, что вызванные потенциалы после предъявления стимула по времени возникают гораздо раньше, чем его осознание.

Из этого следует ряд важных выводов.

1. Если это осознание не наступает по каким-либо причинам, то вызванные потенциалы все равно с очень высокой достоверностью позволяют прогнозировать ту реакцию, которая была бы у человека, если бы информация о стимуле проникла в его сознание (причинами, по которым зрительный образ стимула может не успеть сформироваться и проникнуть в сознание пользователя, могут быть, например, его очень сильная зашумленность, фрагментарность или слишком короткое время его предъявления).

2. Реакция на стимул на уровне вызванных потенциалов не подвергается критическому анализу и корректировке на уровне сознания, т. е. является гораздо более "искренней" и

"откровенной", адекватной и достоверной, чем сознательные ответы на опросник с тем же самым стимульным материалом (сознательные ответы зависят от мотивации, конъюнктуры и массы других обстоятельств).

3. Для получения информации о подсознательной реакции пользователя на стимульный материал он может предъясняться в значительно более высоком темпе, чем при сознательном тестировании.

4. При подсознательном тестировании пользователь может даже не знать о том, что оно проводится.

Все это в совокупности означает, что системы с семантическим резонансом позволяют получить и вывести на уровень сознания обычно ранее неосознаваемую адекватную информацию о состоянии своего сознания, систем мотивации, целеполагания, ценностей и т. д., т. е. расширить область осознаваемого. Это позволяет создать качественно более благоприятные условия для управления состоянием сознания, чем ранее, что является важным эволюционным достижением технократической цивилизации.

Системы с семантическим резонансом могут эффективно использоваться в ряде направлений:

- психологическое и профессиональное тестирование, подбор персонала, в том числе для действий в специальных условиях и в измененных формах сознания;
- модификация сознания, систем мотиваций, целеполагания, ценностей и др. (компьютерное нейролингвистическое программирование: "компьютерные НЛП-технологии");
- компьютерные игры с системами семантической обратной связи.

4.1.5. Системы виртуальной реальности. Эффекты присутствия, деперсонализации и модификация сознания пользователя

Виртуальная реальность (ВР) — модельная трехмерная (3D) окружающая среда, создаваемая компьютерными средствами и реалистично реагирующая на взаимодействие с пользователями.

Технической базой систем виртуальной реальности являются современные мощные персональные компьютеры и программное обеспечение высококачественной трехмерной визуализации и анимации. В качестве устройств ввода-вывода информации в системах ВР применяются виртуальные шлемы с дисплеями (HMD), в частности шлемы со стереоскопическими очками, и устройства 3D-ввода, например мышь с пространственно управляемым курсором или "цифровые перчатки", которые обеспечивают тактильную обратную связь с пользователем.

Совершенствование систем виртуальной реальности приводит ко все большей изоляции пользователя от обычной реальности, так как все больше каналов взаимодействия пользователя с окружающей средой замыкаются не на обычную, а на виртуальную среду — виртуальную реальность, которая при этом становится все более и более функционально замкнутой и самодостаточной.

Создание систем ВР является закономерным следствием процесса совершенствования компьютерных систем отображения информации и интерфейса управления.

При обычной работе на компьютере монитор занимает не более 20% поля зрения пользователя. Системы ВР перекрывают **все** поле зрения.

Обычные мониторы не являются стереоскопическими, т. е. не создают объемного изображения. Правда, в последнее время появились разработки, которые позволяют преодолеть это ограничение (достаточно сделать поиск в yandex.ru по запросу "Стереоскопический монитор"). Системы ВР изначально были стереоскопическими.

Звуковое сопровождение, в том числе со стерео- и квадрослухом, сегодня уже стало стандартом. В системах ВР человек не слышит ничего, кроме звуков этой виртуальной реальности.

В некоторых моделях систем виртуальной реальности пользователи имеют возможность восприятия изменяющейся перспективы и видят объекты с разных точек наблюдения, как если бы они сами находились и перемещались внутри модели.

Если пользователь располагает более развитыми (**погруженными**) устройствами ввода, например такими, как цифровые перчатки и виртуальные шлемы, то модель может даже надлежащим образом реагировать на такие **действия** пользователя, как поворот головы или движение глаз.

Необходимо отметить, что в настоящее время системы виртуальной реальности развиваются очень быстрыми темпами и явно выражена тенденция проникновения технологий виртуальной реальности в стандартные компьютерные технологии широкого применения.

Развитие этих и других подобных средств привело к появлению качественно новых эффектов, которые ранее не наблюдались или наблюдались в очень малой степени:

- эффект присутствия пользователя в виртуальной реальности;
- эффект деперсонализации и модификации самосознания и сознания пользователя в виртуальной реальности.

Эффект присутствия — это создаваемая для пользователя иллюзия его **присутствия** в смоделированной компьютером среде, при этом создается полное впечатление "присутствия" в виртуальной среде, очень сходное с ощущением присутствия в обычном реальном мире.

При этом виртуальная среда начинает осознаваться как реальная, а о реальной среде пользователь на время как бы совершенно или почти полностью "забывает". При этом технические особенности интерфейса также вытесняются из сознания, т. е. мы не замечаем этот интерфейс примерно так же, как собственное физическое тело или глаза, когда смотрим на захватывающий сюжет. Таким образом, реальная среда **замещается** виртуальной средой.

Исследования показывают, что для возникновения и силы эффекта присутствия определяющую роль играет *реалистичность движения* различных объектов в виртуальной реальности, а также убедительность реагирования объектов виртуальной реальности при **взаимодействии** с ними виртуального тела пользователя или других виртуальных объектов. В то же время, как это ни странно, естественность вида объектов виртуальной среды играет сравнительно меньшую роль.

Системы виртуальной реальности уже в настоящее время широко применяются во многих сферах жизни.

Одним из первых технологии виртуальной реальности применило НАСА США для тренировки пилотов космических

челноков и военных самолетов при отработке приемов посадки, дозаправки в воздухе и т. п.

Самолет-невидка "Стелс" вообще управляется пилотом, практически находящимся в виртуальной реальности.

Из виртуальной реальности человек управляет роботом, выполняющим опасную или тонкую работу.

Технология Motion Capture позволяет дистанционно "снять" движения с человека и присвоить их его трехмерной модели, что широко применяется для создания компьютерных игр и анимации рисованных персонажей в фильмах.

Особенно эффективно применение виртуальной реальности в рекламе, особенно в интернет-рекламе на стадии информирования и убеждения.

С использованием виртуальной реальности можно показывать различные помещения, например совершить виртуальную экскурсию по музею, учебному заведению, дому, коттеджу или местности (прогулка по Парижу от туристической фирмы).

Во всех этих приложениях важно, что, в отличие от трехмерной графики, виртуальная реальность обеспечивает *эффект присутствия и личного участия пользователя в наблюдаемых им событиях*.

Сегодня уже для всех вполне очевидно, что виртуальная реальность может с успехом использоваться для развлечений, ведь *она помогает представить себя в другой роли и в другом обличье*. Однако в действительности этот эффект связан с модификацией "образа я", т. е. сознания и самосознания пользователя. Это значит, что последствия этого в действительности значительно серьезнее, чем обычно представляют, и выйдут за рамки собственно развлечений.

Как показано автором в ряде работ, приведенных на сайте <http://Lc.kubagro.ru>, форма сознания и самосознания человека определяются тем, как он осознает себя и окружающее, т. е. тем:

- что он осознает как объективное, субъективное и несуществующее;
- с чем он отождествляет себя и что осознает как объекты окружающей среды.

Очевидно, что разработчики новейших компьютерных технологий совершенно неожиданно вторглись в абсолютно новую для себя сферу исследования **измененных форм сознания**, и далеко идущие системные последствия этого ими, как и вообще научным сообществом, пока еще очень мало осознаны.

Еще в 1979–1981 годах автором и Л. А. Бакурадзе были оформлены заявки на изобретение компьютерной системы, выполняющей все трудовые функции физического тела, управляемой с использованием дистанционного мысленного воздействия, т. е. микротелекинеза. По мнению автора, телекинез представляет собой управление физическими объектами путем воздействия на них непосредственно с высших планов без использования физического тела, т. е. *тем же способом, с помощью которого любой человек, осознает он это или нет, управляет своим физическим телом*. Были предложены технические и программные решения и инженерно-психологические методики. Система предлагалась адаптивной, т. е. автоматически настраивающейся на индивидуальные особенности, "почерк" оператора и его состояние сознания, с плавным переключением на дистанционные каналы при повышении их надежности (которая измерялась автоматически) и могла одновременно

с выполнением основной работы выступать в качестве тренера. Человек, начиная работу с системой в обычной форме сознания с использованием традиционных каналов (интерфейса), имея мгновенную адекватную по форме и содержанию **обратную связь** об эффективности своего телекинетического воздействия, должен быстро переходить в форму сознания, оптимальную для использования телекинеза в качестве управляющего воздействия.

С учетом вышесказанного предлагается следующее определение виртуальной реальности.

Система ВР — это система, обеспечивающая:

1) **генерацию полиперцептивной модели реальности** в соответствии с математической моделью этой реальности, реализованной в программной системе;

2) **погружение пользователя в модель реальности** путем подачи на все или основные его перцептивные каналы — органы восприятия, программно-управляемых по величине и содержанию воздействий: зрительного, слухового, тактильного, термического, вкусового и обонятельного и др.;

3) **управление системой** путем использования виртуального **"образа я" пользователя** и виртуальных **органов управления** системой (интерфейса), на которые он воздействует, представляющие собой **зависящую от пользователя часть** модели реальности;

4) **реалистичную реакцию** моделируемой реальности на виртуальное воздействие и управление со стороны пользователя;

5) разрыв отождествления пользователя со своим **"образом я"** из обычной реальности (**деперсонализация**) и отожде-

ствление себя с "виртуальным образом я", генерируемым системой виртуальной реальности (**модификация сознания и самосознания пользователя**);

6) **эффект присутствия** пользователя в моделируемой реальности в своем "виртуальном образе я", т. е. эффект *личного участия пользователя в наблюдаемых виртуальных событиях*;

7) **положительные результаты применения критериев реальности**, т. е. функциональную замкнутость и самодостаточность виртуальной реальности, вследствие чего *никакими действиями внутри виртуальной реальности, осуществляемыми над ее объектами, в том числе объектами виртуального интерфейса, с помощью своего виртуального тела невозможно установить, "истинная" эта реальность или виртуальная*.

В этой связи вспоминается ставший уже классическим первый фильм "Матрица", в котором Морфей, обращаясь к Нео, произносит свою знаменитую фразу: *"Сейчас я покажу тебе, как выглядит **окончательная истинная реальность**"*. Эта фраза сразу вызвала у автора массу ассоциаций и вопросов, в частности:

1) а каковы критерии реальности?

2) а вдруг и эта реальность, которую Морфей назвал окончательной, истинной, в действительности является не более, чем симулятором следующего иерархического уровня, так сказать, более фундаментальным симулятором?

Здесь возникает сложный мировоззренческий вопрос о том, возможно ли **хотя бы в принципе**, находясь в виртуальной реальности, не выходя за ее пределы, установить, что ты

находишься именно в виртуальной, а не истинной реальности, или это возможно сделать только *задним числом*, после выхода из виртуальной реальности и перехода в истинную реальность?

Итак, каковы же критерии реальности?

*По нашему мнению, прежде всего это **самосогласованность** реальности, т. е. получение одной и той же информации качественно различными способами и по различным каналам связи (принцип наблюдаемости):*

- согласованность реальности самой с собой во **времени**;
- согласованность и взаимное подтверждение информации от различных органов восприятия, которые обычно реагируют на различные формы материи и часто являются парными (зрение, слух, обоняние) и расположенными в различных точках **пространства**.

*Например, мы не только что-то видим, но и слышим, и осязаем, и можем попробовать его на вкус и ощутить запах, и все эти восприятия **ОТ РАЗЛИЧНЫХ ОРГАНОВ ЧУВСТВ соответствуют друг другу и означают, что перед нами некий определенный объект, а не галлюцинация или визуализация. Согласованная и взаимно подтверждающая информация из различных органов чувств, в соответствии с принципом наблюдаемости, также может рассматриваться как повышающая достоверность и адекватность восприятия.***

*В современных компьютерных играх мы не только видим довольно качественную визуализацию, но и соответствующее реалистичное звуковое сопровождение. А в системах виртуальной реальности — визуализация стереоскопическая (то, что мы видим **РАЗНЫМИ** глазами как бы с разных точек в **ПРОСТРАНСТВЕ**, также взаимно подтверждается), а также*

появляется тактильный канал с обратной связью, который позволяет ощутить даже твердость, вес и температуру моделируемого в виртуальной реальности объекта. Все это вместе уже создает настолько высокую степень реалистичности, что может возникнуть эффект присутствия в виртуальной реальности, деперсонализация и отождествление с измененным "образом я", моделируемым в виртуальной реальности (переход в измененную форму сознания).

Представим, что эти сформулированные критерии реальности не выполняются, т. е. нарушается ее самосогласованность. Как и в чем это может проявляться?

По-видимому, как своего рода "сбои" и различные **нарушения физических законов** и несогласованности в виртуальной реальности:

- "заикливание" событий, как на заезженной пластинке, т. е. их многократное повторное осуществление без каких-либо изменений (пример: повторный проход черной кошки с характерной остановкой и поворотом головы в дверном проеме в "Матрице");
- прохождение сквозь стены;
- полеты и очень длинные прыжки, а также телепортация в своем "реальном" теле;
- действия в другом темпе времени, т. е. эффект замедления внешнего времени, соответствующий аналогичному ускорению внутреннего времени;
- действия в другом масштабе пространства, "увеличение" и "уменьшение" размеров, наблюдение мега- и микроструктуры материи;

- видение сквозь стены, видение на большие расстояния (в том числе с увеличением, как в телескоп), видение прошлого и будущего;
- телекинез, пирокинез, психосинтез, левитация и т. п.;
- одновременное нахождение в нескольких местах.

Нетрудно заметить, что все эти проявления весьма напоминают так называемые *паранормальные явления*, которые традиционно связывают со *сверхвозможностями* человека, т. е. с его возможностями при высших формах сознания.

Эти явления хотя и редко, но все же наблюдаются в нашем мире, что может указывать на то, что наша "истинная реальность" в определенной мере, возможно, является виртуальной, по крайней мере в большей степени, чем ранее предполагалось.

Вспомним известные в физике принципы относительности Галилея и Эйнштейна.

1. Никакими экспериментами внутри замкнутой системы невозможно отличить состояние покоя от состояния равномерного и прямолинейного движения (Галилей). Следовательно, покоящаяся система отсчета **физически эквивалентна** системе отсчета, движущейся равномерно и прямолинейно под действием сил инерции.

2. Никакими экспериментами внутри ограниченной по размерам замкнутой системы невозможно установить, движется она под действием сил гравитации или по инерции (Эйнштейн). Следовательно, система отсчета, движущаяся в поле сил тяготения, **физически эквивалентна** системе отсчета, движущейся под действием сил инерции.

Легко заметить, что формулировка 7-го пункта в определении системы виртуальной реальности весьма сходна с формулировками принципов относительности Галилея и Эйнштейна: *никакими действиями внутри виртуальной реальности, осуществляемыми над ее объектами, в том числе объектами виртуального интерфейса, с помощью своего виртуального тела, невозможно установить, "истинная" эта реальность или виртуальная.*

Следовательно, **виртуальная система отсчета, локализованная в полнофункциональной виртуальной реальности, полностью физически эквивалентна физической системе отсчета, локализованной в "истинной реальности"**. Учитывая эту аналогию, принцип, предложенный автором, назовем принципом относительности или *принципом эквивалентности виртуальной и истинной реальности.*

4.1.6. Системы с дистанционным телекинетическим интерфейсом

В 1981 году Л. А. Бакурадзе и Е. В. Луценко были оформлены заявки на изобретение компьютерной системы, выполняющей все трудовые функции физического тела, обеспечивающей управление с использованием дистанционного мысленного воздействия, т. е. микротелекинеза.

По мнению автора, телекинез представляет собой управление физическими объектами путем воздействия на них непосредственно с высших планов без использования физического тела, т. е. тем же способом, с помощью которого любой человек, осознает он это или нет, управляет своим физическим телом.

Были предложены технические и программные решения и инженерно-психологические методики. Система предлагалась адаптивной, т. е. автоматически настраивающейся на индивидуальные особенности, "почерк" оператора и его состояние сознания, **с плавным переключением на дистанционные каналы при повышении их надежности** (которая измерялась автоматически) и могла одновременно с выполнением основной работы выступать в качестве **тренажера для овладения высшими формами сознания**.

Человек, начиная работу с системой в обычной форме сознания с использованием традиционных каналов (интерфейса), имея мгновенную адекватную по форме и содержанию обратную связь об эффективности своего телекинетического воздействия, должен быстро переходить в одну из высших форм сознания, оптимальную для использования телекинеза в качестве управляющего воздействия.

4.2. ПОСТРОЕНИЕ СЛОЖНЫХ ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ НА ОСНОВЕ ЕСТЕСТВЕННО-ЯЗЫКОВОГО ИНТЕРФЕЙСА

Естественно-языковой интерфейс (ЕЯИ) — разновидность пользовательского интерфейса, который принимает запросы на естественном языке, а также, возможно, использует ЕЯ и для вывода информации (реакции системы на запрос пользователя).

4.2.1. Сравнительный анализ ЕЯ-интерфейсов и традиционных интерфейсов к структурированным источникам данных (СИД)

В противоположность ЕЯ-интерфейсам, нетрадиционным с точки зрения распространенности, существуют другие виды

пользовательских интерфейсов к СИД, которые можно назвать традиционными.

Среди них интерфейсы:

- с формальным языком запросов;
- с графическим построением запросов;
- основанные на заполнении форм запросов.

Предполагается, что рассматриваемые здесь виды интерфейсов по своему предназначению ограничены только получением информации из базы данных, это предположение сделано в силу ограничения на ЕЯ-интерфейсы областью запросов, поскольку область занесения данных и их модификации с помощью естественно-языковых оболочек является отдельной большой темой для рассмотрения.

В интерфейсах с *формальным языком запросов* пользователь для того, чтобы правильно задать запрос, должен, во-первых, знать синтаксис языка запросов (например, SQL), а во-вторых, представлять устройство конкретного структурированного источника данных (например, реляционную схему базы данных). При работе с этим типом интерфейсов пользователь должен обладать достаточно высокой квалификацией. Опыт показывает, что такой необходимой квалификацией обладают лишь специалисты, проектирующие и создающие информационные системы, и сам термин "пользователь" с учетом современных тенденций здесь не совсем адекватен. Очевидно, ЕЯ-интерфейсы обладают большей гибкостью — один и тот же запрос обычно можно формулировать различными способами. Что немаловажно, ЕЯ-интерфейсы, как правило, обладают системой понятий — описанием предметной области, которая на-

ходится выше логического уровня хранения данных. Это позволяет абстрагироваться от деталей устройства той или иной базы данных как на уровне структуры, так и на уровне содержимого.

Средства *графического построения запросов*, которыми снабжаются многие "настольные" СУБД (например, MS Access, MS FoxPro), безусловно, обладают большим удобством — по крайней мере пользователь не должен держать в голове названия таблиц, полей и конструкции языка. Однако для работы с такими средствами необходим опыт и представление некоторых понятий, относящихся скорее к математике (например, термин "связывание таблицы" в реляционной алгебре), а не к предметной области, и иногда достаточно утомительные действия по заполнению форм. Так, в базе данных Microsoft Access для того чтобы сформулировать выражение `AVG(PERSONNEL.SALARY)`, эквивалентное ЕЯ-фразе "средняя зарплата", требуется около 15 нажатий мышью. Неподготовленный пользователь обычно пасует перед системами, требующими сложных действий. Как и в случае интерфейсов с формальным языком, пользователь должен представлять устройство базы данных. По сути, эти средства позволяют графически создавать формальные запросы, и не случайно они обычно позволяют редактировать пользователю полученный формальный запрос.

Интерфейсы, основанные на *заполнении форм* запросов, являются более дружественными по сравнению с формальными языками. Сама метафора формы и ее заполнения подразумевает, что пользователь сразу видит набор критериев и параметров поиска, а иногда и список возможных значений полей формы, это сводит к минимуму ошибки при вводе запроса. От

предыдущего метода построения пользовательских интерфейсов данный отличается тем, что, как правило, все необходимые запросы уже написаны разработчиком интерфейса и пользователю, чтобы получить ответ, нужно просто вставить недостающие значения. Отличие заключается также в том, что, задавая значения формы, пользователь обычно не выбирает, какие атрибуты данного класса объектов будут в результате, а сам список доступных классов (в реляционной базе — таблиц) ограничен множеством построенных форм. Так работают многие современные коммерческие приложения, работающие с базами данных, — пользователю информация в системе доступна в виде нескольких типовых "срезов" информационного пространства. К недостатку систем, основанных на таком подходе, как и в предыдущем случае, также следует отнести необходимость наличия у пользователя опыта работы с подобными системами, а также необходимость создания форм, что требует дополнительных усилий программиста для создания интерфейса.

Преимущества ЕЯ-интерфейсов достаточно очевидны:

- минимальная предварительная подготовка пользователя. Естественный язык является наиболее привычным и удобным средством коммуникации, и именно в силу этого с ростом эффективности ЕЯ-систем он, безусловно, будет вытеснять другие виды интерфейсов к СИД, традиционные в данный момент;
- простота задания запросов на ЕЯ. Во многих случаях запрос на ЕЯ получается гораздо короче языка на формальном языке, поскольку ЕЯ-представление более емко, ведь в самой структуре языка содержится понятийная база, которую отра-

жает структура источника данных. Зачастую сложность этой структуры отражается на сложности запроса на формальном языке;

- большая скорость создания произвольного запроса (отсутствует стадия формального задания запроса). Как правило, пользователь сразу может сформулировать корректное ЕЯ-представление запроса, поскольку такое представление является самым естественным для человека, тогда как построение запроса на формальном языке даже с помощью вспомогательных средств таит множество ошибок, зачастую исправить которые можно, только проанализировав результат запроса;
- более высокий уровень модели предметной области. Традиционные интерфейсы обычно не обладают моделью предметной области как таковой и в лучшем случае скрывают от пользователя искусственные средства и особенности структуры, присущие конкретному типу СИД (такие как связи по идентификаторам между таблицами в реляционных базах данных или синтаксис XML).

Рассмотрим более подробно недостатки ЕИЯ по сравнению с другими типами интерфейсов.

- Неоднозначность естественного языка приводит к множественности смыслов. Специфика естественного языка такова, что часто запрос может иметь несколько смыслов, о которых пользователь в момент задания запроса не предполагает. Формальные же языки лишены проблемы неоднозначности. Это свойство ЕЯ приводит к усложнению ЕЯ-интерфейсов и

методов анализа, в противном случае ЕЯ-интерфейс получается слишком примитивным для реального использования.

- Недостаточная надежность анализаторов ЕЯ-запросов может привести к неправильному пониманию. Современные ЕЯ-интерфейсы далеко не всегда позволяют диагностировать причины неудач понимания. Причины этих неудач могут корениться как в лингвистической сфере, так и в концептуальной. Например, запрос к кадровой базе данных "Кто получает больше Иванова" может привести к непониманию, если ЕЯ-интерфейс не умеет распознавать вложенные запросы (а в данном случае надо сначала получить значение зарплаты Иванова, а затем сравнить с ней зарплаты сотрудников). Это случай лингвистической проблемы. Второй пример — "Как зовут жен сотрудников?" — может привести к неудаче понимания, если ЕЯ-интерфейс не поймет, что имя супруга/супруги — это реальный атрибут сотрудника, но отсутствующий в данной базе данных. В данном случае налицо будет концептуальная проблема — ЕЯ-интерфейс должен уметь отличать реальную предметную область, которую имеет в виду пользователь, задавая ЕЯ-запрос, от той ее части или трансформации, которая представлена в данном источнике данных.
- Пользователь может иметь завышенные или заниженные ожидания от ЕЯ-интерфейса. Сравнительный анализ типов пользовательских интерфейсов (основанных на формах, с формальным языком запросов, графических) показывает, что в целях построения ЕЯ-интерфейсов превалирует желание максимально приблизить интерфейс к потребностям непод-

готовленного пользователя. Это несколько поднимает планку требований к дружелюбности и надежности ЕЯ-интерфейсов, поскольку пользователь, впервые столкнувшись с системой, понимающей естественный язык, слабо представляет, насколько интеллектуальна система. При этом ожидание к степени понимания ЕЯ может отличаться от реальных способностей системы в обе стороны, т. е. пользователь может спрашивать систему о том, чего она "не знает", а может "по привычке" использовать простейшие шаблонные формулировки запросов. В других же типах интерфейсов к СИД рамки того, что пользователь может делать с помощью интерфейса, видны, как правило, сразу.

Поскольку характеристики ЕЯ-интерфейсов и систем для их построения могут существенно различаться, то преимущества и недостатки ЕЯ-интерфейсов по сравнению с другими типами интерфейсов к СИД можно выделить довольно схематично, только на качественном уровне. Для сравнения подходов к построению ЕЯ-интерфейсов введем метрику показателей, характеризующих качество ЕЯ-интерфейсов к структурированным источникам данных.

4.2.2. Критерии качества ЕЯ-интерфейсов

Для сравнительного анализа подходов к созданию ЕЯ-интерфейсов рассмотрим такую качественную интегральную характеристику, как надежность. Под *надежностью* здесь понимается способность ЕЯ-интерфейса правильно понимать намерения пользователя по получению информации из источника при условии, что пользователь корректно выразил потребности в виде ЕЯ-запроса. Надежность отражает правильность прин-

ципов, лежащих в основе методе ЕЯ-анализа, и правильность (корректность) построения ЕЯ-интерфейса к конкретному СИД.

Любой ЕЯ-интерфейс имеет некоторое пространство правильно понимаемых запросов. Чем больше это пространство, тем большей *полнотой* обладает ЕЯ-интерфейс. Полнота — характеристика, тесно связанная с *гибкостью* интерфейса. Поскольку пространство ЕЯ-запросов весьма неоднородно, следует говорить о различных типах запросов, т. е. групп запросов, имеющих сходное строение. Гибкость — показатель того, насколько разнообразные типы запросов может понимать ЕЯ-интерфейс. Речь в основном идет о так называемых "трудных" типах запросов, в числе которых — вложенные, эллипсис, анафорические.

Другой важной характеристикой является *дружественность* интерфейса, которую можно определить как меру того, насколько ЕЯ-интерфейс удобен в работе, насколько корректно он может сообщать о проблемах понимания, может ли он помогать в переформулировке неберущихся запросов и т. д.

4.2.3. Критерии стоимости построения и сопровождения ЕЯ-интерфейса

Вышеперечисленные характеристики входят в оценки качества ЕЯ-интерфейса. Важным критерием при сравнении ЕЯ-интерфейсов является также трудоемкость его создания, то есть необходимое количество усилий (времени), требуемых для его построения. Ранние ЕЯ-интерфейсы создавались для каждой базы данных отдельно, и, разумеется, их стоимость была очень большой. Все эти системы были экспериментальными. Усугубляло проблему также то, что до конца 1970-х годов не

было единого универсального формального языка запросов к базам данных. Ранние системы понимания ЕЯ-запросов к СУБД были непортируемыми на другие базы данных, и зачастую лингвистическое ядро не отделялось от предметно-ориентированных настроек.

Современные промышленные системы построения ЕЯ-интерфейсов [8] обладают достаточно высокой степенью портируемости, что, безусловно, снижает стоимость построения ЕЯ-интерфейса. Лингвистическое ядро является универсальным элементом, словарь содержит универсальную лексику, используемую во многих ЕЯ-интерфейсах, модели предметной области могут содержать шаблоны, общие для нескольких предметных областей, и т. д. Зачастую используется метафора "фабрики и изделия", изделием выступает ЕЯ-интерфейс, который собирается из готовых компонентов, которые настраиваются под конкретную базу данных.

Следует отметить, однако, что вопрос портирования на другие языки является открытым. Подавляющее большинство исследований проведено для английского языка, некоторые особенности которого изначально заложили в пути исследований мину замедленного действия — первоначально огромное количество усилий было потрачено на анализ синтаксиса. Сейчас можно сказать, что эти усилия не оправдали себя [41].

На трудоемкость создания ЕЯИ влияет также необходимая квалификация настройщика ЕЯ-интерфейса. Для систем, требующих навыков лингвиста, трудоемкость построения ЕЯИ больше, чем для систем, где для построения интерфейса требуется просто описать предметную область по некоторым предопределенным шаблонам и отобразить ее на схему базы данных,

и дело здесь не только в стоимости труда лингвиста и инженера знаний или специалиста в области баз данных. Системы, требующие подстроек на уровне лингвистического ядра, являются более гибкими, поскольку позволяют разрешать проблемы понимания ЕЯ-запросов написанием соответствующих "заплаток", однако работы по написанию таких "заплаток" являются настолько сложными, требуют такого уровня понимания принципов машинного анализа ЕЯ в целом, что настройка ЕЯИ на уровне лингвистического процессора зачастую возможна только авторами системы построения ЕЯИ. Впрочем, сложность подстройки ядра очень сильно зависит от принципов анализа, используемого при написании инструментария, открытости ядра и т. д.

4.2.4. Вопросы портируемости

Коснемся теперь характеристики, присущей системам построения ЕЯ-интерфейсов — портируемость компонентов анализа. В [1] вводится несколько видов портируемости на:

- 1) другую предметную область;
- 2) другой язык;
- 3) другую СУБД;
- 4) другие платформы и языки программирования.

Портируемость на другую ПО. Обычно, если ЕЯ-интерфейс портируется на другую предметную область, систему необходимо "научить" словам и концептам, используемым в новой предметной области. Также надо отобразить МПО и МБД. Портируемость на другую предметную область может проводиться различными специалистами, и механизмы портирования могут быть ориентированы на:

- программиста: в некоторых системах часть кода (обычно небольшая и четко определенная) должна быть переписана;
- инженера знаний: конфигурация системы без программирования, построением схем предметной области и/или введением и описанием новых концептов ПО;
- администратора СУБД: БД-центрическое описание предметной области, например описание некоторых важных для ПО ЕЯИ характеристик для каждой таблицы и поля БД;
- конечного пользователя: в предположении, что настройка ЕЯИ — постоянный процесс, ЕЯ снабжается средствами для настройки методом введения определений концептов на уровне ЕЯ и другими пригодными для конечного пользователя методами.

Портируемость на другой естественный язык. Подавляющее большинство ЕЯ-интерфейсов к БД были разработаны для английского языка. Переориентация ЕЯИ на другой язык является для большинства систем большой проблемой вследствие активного использования законов языка на уровне морфологии, синтаксиса, существенно отличающихся от языка к языку. Иногда настройка на язык возможна переписыванием синтаксических и семантических правил, а также новым наполнением словаря. В описываемой работе переносимость на другой язык является достаточно высокой, что обусловлено использованием преимущественно семантической информации при анализе.

Портируемость на другую СУБД. Система является портируемой на другую СУБД, если она позволяет перенос ЕЯ-интерфейса на другую СУБД. В случае если ЕЯИ генерирует запросы в распространенном языке (например, SQL), он может быть легко портирован на другую СУБД, если она поддерживает этот язык. Если язык запросов ЕЯИ не поддерживается новой СУБД, ЕЯИ может быть портирован переписыванием компонента генерации формального языка, однако это справедливо только для систем, в архитектуру которых включен промежуточный уровень запроса. В противном случае для портирования необходимо существенно переписать ЕЯИ.

Портируемость на другую платформу. Некоторые ЕЯИ могли исполняться только на дорогих исследовательских машинах, использующих экзотические языки программирования (например, Lisp-машины), что делает их практически неприменимыми в реальных приложениях. В последнее время появление недорогих мощных компьютеров и повсеместная доступность таких языков, ориентированных на приложения искусственного интеллекта, как Lisp и Prolog, делают доступными и ЕЯИ, написанные на этих языках.

4.2.5. Основные составные части ЕЯ-интерфейсов

Кратко рассмотрим основные части ЕЯ-интерфейсов и их взаимосвязи. Прежде всего следует выделить из интерфейса *анализатор ЕЯ* как компонент, реализующий тот или иной метод анализа естественного языка, и от принципов построения которого зависит архитектура системы и основные характеристики интерфейсов на основе данного компонента.

Работа анализатора заключается в построении *внутреннего представления* входного ЕЯ-текста либо запроса обычно в виде некоторой структуры, например синтаксического дерева, семантической сети, фреймовой структуры и т. д. Предшествующим этапом для процесса анализа является *лексический анализ (пред-анализ)*, который преобразует входной текст как последовательность символов в цепочку лексем, поступающую на вход анализатора.

Необходимым компонентом работы анализатора является *словарь*, который содержит слова и фразы обычно с привязкой к ним определенной информации, связанной с семантикой, морфологией и т. д. в зависимости от подхода анализа ЕЯ. Еще одним важным компонентом многих систем является *модель предметной области*, структура которой варьируется в очень больших пределах от системы к системе.

Для построения запроса на *формальном языке* источника данных используется *модель источника данных*, отражающая основную структуру СИД, ее части, существенные для данного ЕЯИ.

Для перевода запроса из внутреннего представления системы в формальный язык источника данных предназначен процесс *генерации формального запроса*. Некоторые системы имеют также модуль синтеза ЕЯ, который может применяться для генерации естественно-языкового представления запроса, например для верификации понимания запроса системой, а также для генерации уточняющих вопросов [8].

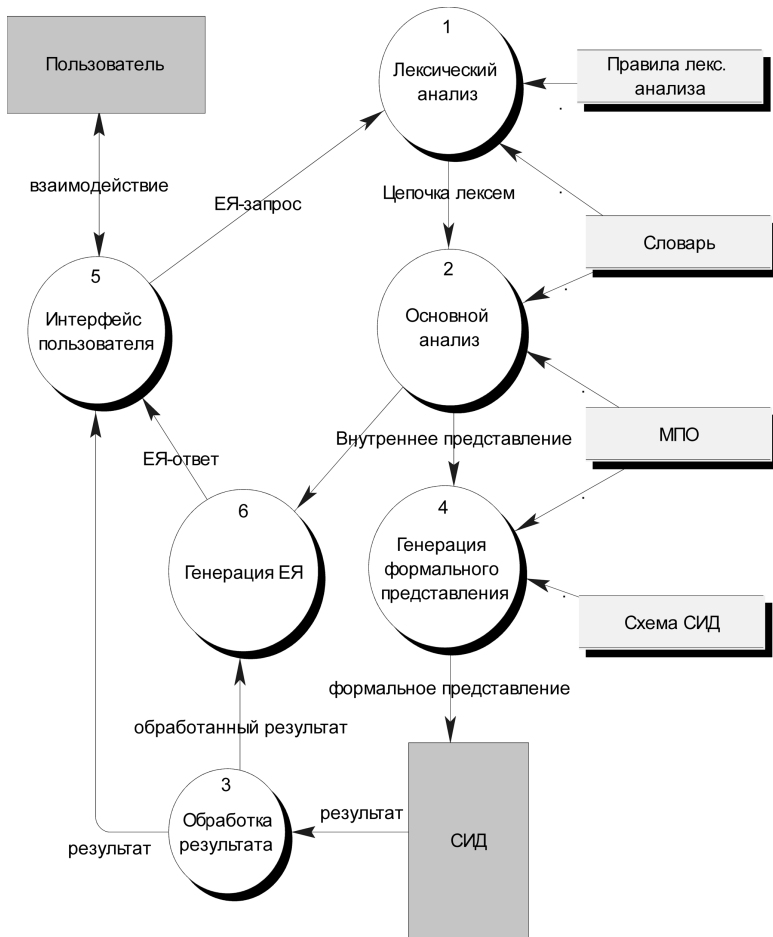


Рис. 4.1. Основные составляющие ЕЯ-интерфейсов и их взаимосвязи

Модель предметной области в некоторых системах (см. там же) дополняется *базой знаний* со средствами вывода новых знаний.

На рисунке 4.1 приведены основные составляющие ЕЯИ и взаимосвязи между ними, представленные потоками данных.

4.3. СОЗДАНИЕ И ВНЕДРЕНИЕ ТЕХНИЧЕСКИХ И ЭКОНОМИЧЕСКИХ ПРОЕКТОВ ПРИ ПОМОЩИ СОВРЕМЕННЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

4.3.1. Проектирование: принципы и методы создания ИС

Проектирование ИС — процесс создания и внедрения проектов комплексного решения экономических задач по новой технологии. Сюда включается: детальная разработка отдельных проектных решений, их анализ, апробация и внедрение.

Качественное проектирование и внедрение являются основной предпосылкой эффективного функционирования системы при постоянном совершенствовании ее обеспечивающих и функциональных составляющих. *Цель всех этих работ* состоит не только в компьютеризации информационных потоков, но и в совершенствовании самого управления и организации основной деятельности экономического объекта. Поэтому первый руководитель должен иметь представление об имеющихся на рынке технических и программных средствах, тенденциях в их развитии, основных принципах проектирования ИС. В каждом подразделении организации должен быть назначен сотрудник, ответственный за проектирование и внедрение ИС, который собирает нужную информацию, подбирает технику и программные средства, ведет обучение персонала, руководит внедрением и анализом функционирования информационных систем.

Современные предприятия относятся к классу больших динамических систем с характерной многопрофильной деятельностью и большим числом кооперативных связей с партне-

рами. При этом возрастает динамичность бизнес-процессов, связанных с изменяющимися потребностями и сильной конкуренцией. Управление бизнес-процессами предполагает рассмотрение всех материальных, финансовых, трудовых и информационных потоков с системных позиций, т. е. во взаимодействии. Достижения в области ИС и ИТ дают возможность проведения *инжиниринга и реинжиниринга бизнес-процессов*.

Реинжиниринг бизнес-процессов — это создание новых, более эффективных бизнес-процессов без учета предшествующего развития (все начинается заново, подвергается сомнению, проявляется творческое начало во всех действиях). Реинжиниринг выдвигает на первый план *новые цели и методы, способствующие*: глобализации бизнеса (работа с клиентом в режиме "24 часа 365 дней" в любой точке мирового пространства); снижению затрат и численности персонала; ускоренному продвижению новых технологий; росту мобильности персонала и ориентации деятельности на будущие потребности клиентуры; росту качества продукции и услуг.

Технология реинжиниринга основана на том, что в процессе управления пользователь активно использует современные информационные технологии для обучения, стратегического и тактического планирования, анализа возможных путей перестройки и улучшения бизнес-процессов, управления изменениями, реализации проектов и др.

Инжиниринг бизнес-процессов включает в себя реинжиниринг бизнес-процессов, проводимый с определенной периодичностью, например один раз в 5 лет, и последующее непрерывное улучшение.

Обратный инжиниринг предполагает исследование функционирующих на предприятии бизнес-процессов. Цель этапа заключается в проведении диагностики "узких мест" в организации существующих бизнес-процессов и формулировании направлений их реорганизации.

На этапе обратного инжиниринга постановка задач реорганизации бизнес-процессов (РБП) уточняется, сформулированные на этапе идентификации бизнес-процессов в общем виде цели РБП могут быть скорректированы по результатам исследования существующей системы организации бизнес-процессов.

Для оценки эффективности существующих бизнес-процессов используются прежде всего методы и средства для выявления:

- наиболее трудоемких и затратных функций;
- функций, не вносящих вклад в образование прибыли;
- функций с низким коэффициентом использования ресурсов.

Массовое проектирование ИС базируется на использовании нормативно-правовой базы (федеральных законах, ГОСТ и пр.) и семи основополагающих принципах.

1. *Принцип эффективности* заключается в том, что выгоды от новой автоматизированной системы должны быть равными или больше расходов на нее.

2. *Принцип контроля* требует, чтобы информационная система обладала механизмами для защиты имущества фирмы, ее данные были бы достаточно надежны для принятия управленческих решений.

3. *Принцип совместимости* предполагает, что проект системы будет учитывать организационные и человеческие факторы предприятия. Иными словами, система должна учитывать организационную структуру предприятия, а также интересы, квалификацию и отношение людей, выполняющих различные функции.

4. *Принцип гибкости* требует от системы возможности расширения без проведения больших изменений. Например, в новую автоматизированную систему учета можно легко ввести новые счета в план счетов, если он изменился, новые хозяйственные операции и др.

5. *Принципы системности* позволяют исследовать объект как единое целое во взаимосвязи всех его элементов. На базе системного подхода применяется и метод моделирования, позволяющий моделировать изучаемые процессы вначале для анализа, а затем и синтеза создаваемых систем.

6. *Принцип развития* заключается в непрерывном обновлении функциональных и обеспечивающих составляющих системы.

7. *Принцип стандартизации и унификации* предполагает использование уже накопленного опыта в проектировании и внедрении ИС посредством программирования типовых элементов, что позволяет сократить затраты на создание ИС. Под унификацией понимается реализация при разработке программ принципа единообразия в методах, средствах и содержании и формах представления информации. Под стандартизацией понимается обязательное соблюдение при разработке проектных решений, утвержденных государственным стандартом образцов форм представления и описания элементов проекта ИС.

К известным методикам и стандартам, касающимся организации жизненного цикла ИС, можно отнести: методику Oracle CDM (Custom Development Method) по разработке прикладных ИС под заказ; международный стандарт ISO/IEC 12207 по организации жизненного цикла продуктов программного обеспечения; отечественный стандарт ГОСТ 34.003-90.

Методы проектирования ИС, т. е. способы создания ИС: индивидуальный (оригинальный); типовое проектирование; автоматизированный проект (САПР).

1. *Индивидуальное проектирование* характеризуется тем, что все виды работ для различных объектов выполняются по индивидуальным проектам. В процессе индивидуального проектирования применяются свои оригинальные методики и средства проведения работ. Состав работ на всех этапах обследования, проектирования и внедрения создается для конкретного объекта по мере необходимости. Для этого метода проектирования характерны высокая трудоемкость, большие сроки проектирования, плохая модернизируемость, слабое сопровождение.

2. *Типовое проектирование* — разбиение системы на множество составных компонентов и создание для каждого из них законченного проектного решения, которое при внедрении привязывается к конкретным условиям объекта. В зависимости от декомпозиции различают: 1) *элементное*, 2) *подсистемное* и 3) *объектное проектирование*.

При *элементном методе проектирования* вся система разбивается на конечное множество элементов, каждый из которых является типовым. В качестве элементов могут выступать проектные решения по информационному, техническому, программному видам обеспечения.

Подсистемный метод проектирования характеризуется более высокой степенью интеграции элементов ИС. Декомпозиция системы осуществляется на уровне функциональных подсистем, иногда комплекса задач, каждая из выделенных подсистем представляется в законченном виде ППП.

Объектное проектирование — декомпозиция ИС не производится. Типовой объект создается в целом для некоторого обобщенного объекта, определенной группы.

В последнее время все большее число организаций, предприятий, фирм предпочитают покупать готовые пакеты и технологии, а если необходимо, добавлять к ним свое программное обеспечение, так как разработка собственных ИС и ИТ связана с высокими затратами и риском.

Рассмотрим первый из путей, т. е. возможности использования *типовых проектных решений (ТПР)*, включенных в пакеты прикладных программ (ППП). Наиболее эффективно информатизации поддаются следующие виды деятельности: бухгалтерский учет, включая управленческий и финансовый; справочное и информационное обслуживание экономической деятельности; организация труда руководителя; автоматизация документооборота; экономическая и финансовая деятельность, обучение.

Наибольшее число ППП создано для *бухгалтерского учета*: "1С: бухгалтерия", "Турбо-Бухгалтер", "Инфо-Бухгалтер", "Парус", "АВАСУС", "Бэмби+" и др.

Справочное и информационное обеспечение управленческой деятельности представлено следующими ППП: "ГАРАНТ" (налоги, бухучет, аудит, предпринимательство, банковское дело, валютное регулирование, таможенный контроль);

"КОНСУЛЬТАНТ+" (налоги, бухучет, аудит, предпринимательство, банковское дело, валютное регулирование, таможенный контроль).

Экономическая и финансовая деятельность представлена следующими ППП: а) "Экономический анализ и прогноз деятельности фирмы, организации" (фирма "ИНЕК"), реализующий функции: экономический анализ деятельности фирмы, предприятия; составление бизнес-планов; технико-экономическое обоснование возврата кредитов; анализ и отбор вариантов деятельности; прогноз баланса, потоков денежных средств и готовой продукции; б) многопользовательский сетевой комплекс полной автоматизации корпорации "Галактика" (АО "Новый атлант"), который включает такие важные аспекты управления, как планирование, оперативное управление, учет и контроль, анализ, а для принятия решений — позволяет в рамках СППР обеспечивать решение задач бизнес-планирования с использованием ППП Project-Expert.

В основе типового проектирования лежит первоначальная классификация или типизация экономических объектов по их важнейшим параметрам. Затем создаются типовые схемы их решения, внедрение которых в дальнейшем на конкретном предприятии сводится к привязке их в условиях данного предприятия. Декомпозиция функциональных компонентов ИС является основой технологии типового проектирования. Типовое проектирование предполагает разбиение ИС на отдельные составляющие и создание для каждого из них законченного проектного решения, которое затем с некоторыми модификациями будет использоваться при проектировании ИС.

ТПР в области ИС представляет комплект технической документации, содержащей проектное решение по части объекта проектирования и предназначенной для многократного использования в процессе разработки, внедрения и функционирования ИС с целью уменьшения трудоёмкости и разработки, затрат на создание ИС.

3. *Сущность технологии автоматизированного проектирования.* В области автоматизации проектирования ИС в последние годы сформировалось новое направление CASE-технологии (Computer Aided System / Software Engineering). Это совокупность методов анализа, проектирования, разработки и сопровождения ИС с максимальной автоматизацией процессов разработки и функционирования систем. Организационно CASE-индустрия включает компании трех типов:

- разработчиков средств анализа и проектирования;
- разработчиков специальных средств с ориентацией на узкие предметные области;
- обучающие, информирующие и консалтинговые фирмы, оказывающие соответствующие услуги при использовании CASE-пакетов.

Компании, предоставляющие такие услуги, получили название *системных интеграторов*. Следует отметить, что этот термин имеет два понятия. Согласно первому, под термином "системный интегратор" понимаются как компании, специализирующиеся на сетевых и телекоммуникационных решениях (сетевые интеграторы), имеющие, в свою очередь, сеть своих продавцов, так и компании — программные интеграторы. Существует и другая трактовка понятия "системный интегратор", которая закрепляет за компанией комплексное решение задач за-

казчика при проектировании ИС. При этом имеется в виду, что заказчик полностью доверяет детальную проработку и реализацию проекта системному интегратору, оставляя за собой лишь определение исходных данных и задач, которые должна решать реализуемая ИС.

Фирмы-интеграторы создают, как правило, дилерскую сеть представительств в ряде городов России и в странах СНГ. При этом компании осуществляют техническую и информационную поддержку своих дилеров, проводя совместные семинары и презентации, регулярно рассылая им информационно-рекламные материалы о новых продуктах и перспективных технологиях, осуществляют совместное участие в крупных региональных проектах.

Другим вариантом организации системной интеграции является выполнение проектов от консалтинга до создания прикладной системы, т. е. заказчику сдается готовая к эксплуатации информационная система "под ключ" и допускается привлечение организаций и квалифицированных специалистов в качестве партнеров для реализации.

CASE-технологии проектирования ИС ориентируются на архитектуру готовых программных изделий. Это обусловлено необходимостью быстрее создавать и внедрять ИС при меньших затратах; обеспечить единый простой интерфейс; сократить усилия на обслуживание существующих приложений при их адаптации к постоянным изменениям в программно-технической среде. CASE-технология включает вопросы определения требований к системе и создание проекта на глобальном уровне, так чтобы он наиболее полно отвечал требованиям с учетом заданных экономических и технологических ограничений. CASE-технология со-

держит средства поддержки на всех основных этапах проектирования и внедрения ИС, при этом на этапе анализа целей создания системы обычно используется концепция диаграмм потоков данных. Причем особенно уделяется внимание связям между данными. В результате между входными и выходными данными устанавливаются парные связи. CASE-технология обеспечивает: последовательную декомпозицию сложной задачи на более простые компоненты; уменьшение времени и стоимости создания системы по сравнению с неавтоматизированными технологиями; контроль за взаимосвязями и полнотой представления отдельных компонент проекта; одновременное внесение нескольких изменений в проект.

Ядром системы является база данных проекта — *репозиторий* (словарь данных). Он представляет собой специализированную базу данных, предназначенную для отображения состояния проектируемой ЭИС в каждый момент времени.

Репозиторий содержит информацию об объектах проектируемой ЭИС и взаимосвязях между ними, все подсистемы обмениваются данными с ним. В репозитории хранятся описания следующих объектов:

- проектировщиков и их прав доступа к различным компонентам системы;
- организационных структур;
- диаграмм и пр.

Преимущества CASE-технологии по сравнению с традиционной технологией оригинального проектирования сводятся к следующему:

- улучшение качества разрабатываемого программного приложения за счет средств автоматического контроля и генерации;
- возможность повторного использования компонентов разработки;
- поддержание адаптивности и сопровождения ЭИС;
- снижение времени создания системы, что позволяет на ранних стадиях проектирования получить прототип будущей системы и оценить его;
- освобождение разработчиков от рутинной работы по документированию проекта, так как при этом используется встроенный документатор;
- возможность коллективной разработки ЭИС в режиме реального времени.

4.3.2. Этапы создания информационных систем (ИС)

Выделяется несколько этапов создания ИС.

1. Предпроектная стадия.
 - 1.1. Обследование объекта и обоснование необходимости создания ИС.
 - 1.2. Формирование требований пользователя к ИС.
 - 1.3. Оформление отчета о выполненной работе и заявки на разработку ИС.
 - 1.4. Разработка и утверждение технического задания ИС.
2. Проектная часть.
 - 2.1. Разработка проектных решений по системе и ее частям.
 - 2.2. Разработка документации на ИС.

- 2.3. Разработка и оформление документации на поставку изделий для комплектования ИС.
- 2.4. Разработка рабочей документации на систему или ее части.
- 2.5. Разработка или адаптация программ.
- 3. Стадия внедрения.
 - 3.1. Подготовка объекта автоматизации к вводу в действие.
 - 3.2. Подготовка персонала, проводится обучение персонала.
 - 3.3. Строительно-монтажные работы, в том случае, если строится специализированное здание.
 - 3.4. Проведение предварительных испытаний.
 - 3.5. Проведение опытной эксплуатации.
 - 3.6. Проведение опытных испытаний.
 - 3.7. Введение в промышленную эксплуатацию.
- 4. Анализ функционирования.
 - 4.1. Гарантийное и послегарантийное обслуживание.
 - 4.2. Внесение изменений в проектные решения.

Основными участниками процесса создания ИС являются предприятие-заказчик, для которого она создается, и предприятие-разработчик, выполняющее работы по проектированию ИС. Юридические и организационные взаимоотношения конкретно заказчиков и разработчиков регулируются заключенными между ними договорами.

Заказчик обязан заключить договор на создание ИС, приобрести технические средства, подготовить задание на строительство или реконструкцию помещения, если необходимо совместно с разработчиком выполнить работы предпроектной

стадии, в необходимые сроки подготовить помещение, приобрести и установить технические средства, разработать и осуществить мероприятия по совершенствованию организации управления и производства. На стадии проектирования необходимо обеспечить обучение персонала, обеспечить запись необходимой информации на машинные носители и ее контроль, обеспечить уточнение исходных данных по составу и структуре информационной базы, завершить ее формирование, подготовить контрольные примеры, организовать поэтапную приемку рабочих программ с проверкой на контрольных примерах. При подготовке объекта к внедрению заказчик выполняет следующие работы: внедряет локальные и общегосударственные классификаторы, унифицированные формы документов, проводит в намеченные сроки мероприятия по подготовке объекта к внедрению ИС. При вводе системы в действие заказчик завершает ввод в эксплуатацию технических средств, завершает опытную эксплуатацию комплекса задач и принимает в промышленную эксплуатацию. Разрабатывает и согласовывает с разработчиком программу приема сдаточных испытаний и организует работу приемочной комиссии по проведению испытаний системы.

Основная цель разработчика — создание ИС. На предпроектной стадии проводит обследование объекта, обрабатывает материалы обследования, определяет задачи, комплексы задач, подлежащие автоматизации, определяет экономическую эффективность. На стадии ТП разрабатывает документацию, в соответствии с утвержденным ТЗ осуществляет методическое руководство работами по созданию классификаторов, внедрению унифицированных систем документации, разрабатывает

структуру информационной базы, принимает участие в обучении персонала заказчика. На стадии рабочей документации осуществляет разработку программного обеспечения, генерацию рабочих программ, участвует в разработке должностных инструкций управленческого персонала, технологических инструкций пользователя. При вводе системы в действие разработчик осуществляет методическое руководство, вносит коррективы в проекты, принимает участие в сдаче задач и комплексов задач в промышленную эксплуатацию и участвует в работе комиссии по приемке системы в промышленную эксплуатацию.

4.3.3. Реинжиниринг бизнес-процессов с помощью системы ReThink

С интегрированным подходом к поддержке реинжиниринга на основе интеллектуальных технологий можно ознакомиться на примере одного из перспективных инструментальных средств BPR — системы ReThink, разработанной фирмой Gen-sym (США). В этой системе объединены возможности ключевых современных информационных технологий: графический объектно-ориентированный язык для описания моделей и проектов, средства анимации и имитационного моделирования реконструируемых процессов, методы искусственного интеллекта для полного и адекватного представления экспертных знаний о процессах. Все это открыло доступ к непосредственному моделированию и реконструированию бизнес-процессов новой группе пользователей — менеджерам. Сочетание прозрачных средств интерактивной графики с возможностями моделирования процессов в реальном времени позволяет им самостоятельно, без

помощи программистов, воплощать свои идеи в виде работающих моделей процессов.

Система ReThink построена на базе инструментального комплекса G2 и является проблемно-ориентированным приложением, позволяющим разработчикам использовать не только специализированные средства моделирования бизнес-процессов, но и универсальные средства комплекса по созданию интеллектуальных объектно-ориентированных систем управления в реальном времени.

Фирма Gensym, основанная в 1986 году, специализируется на создании объектно-ориентированных интеллектуальных систем, функционирующих в реальном режиме времени и использующих средства имитационного моделирования. Основной программный продукт фирмы — инструментальный комплекс G2, представляющий собой универсальную среду разработки приложений, функционирующих в реальном масштабе времени. Комплекс G2 нашел применение практически во всех отраслях промышленности, а также в космических приложениях (NASA), телекоммуникациях, безопасности офисов и др. Разработчики, использующие комплекс G2, отмечают такие его достоинства, как существенное сокращение сроков создания приложений по сравнению со средствами традиционного программирования; удобные средства интерактивной графики и анимации, обеспечивающие прозрачность и наглядность моделей и приложений в целом; средства вывода решений, позволяющие параллельно обрабатывать тысячи параметров в реальном режиме времени; архитектура "клиент — сервер"; разнообразие аппаратно-программных платформ.

Тесная связь фирмы Gensym с техническими университетами обеспечивает быстрое внедрение новейших достижений в области информационных технологий в коммерческие программные продукты фирмы. На базе комплекса G2 разработан ряд проблемно-ориентированных инструментальных средств для решения задач диагностики и мониторинга процессов, решения задач составления расписаний, решения задач пространственного планирования, обучения на основе методов нейронных сетей и ряд других.

Для представления моделей бизнес-процессов используются диаграммы, состоящие из блоков и соединений. Блоки представляют задачи в бизнес-процессах, а соединения — потоки сущностей: документов, информации, а также предметов, фигурирующих в бизнесе, например запасных частей или упаковок с отпускаемой продукцией. В системе реализован ряд стандартных блоков, которые могут быть использованы в качестве сборочных элементов для построения работающих моделей практически любых процессов, например: источник заявок, принятие решения, обработка задания. Свойства и поведение блоков могут описываться как точными, так и случайными величинами. В случае необходимости разработчик переопределяет поведение блоков или задает новые их классы с помощью встроенных базовых средств.

Объектная ориентация системы ReThink позволяет создавать понятные и довольно наглядные модели бизнес-процессов, что упрощает освоение и использование системы непрограммирующими пользователями. Объекты, построенные в результате моделирования бизнес-процессов, становятся естественной основой для проектирования информационных сис-

тем поддержки этих процессов. В этом смысле средства системы ReThink могут рассматриваться как развитие CASE-средств.

ReThink поддерживает анимацию потоков работ в ходе моделирования деятельности компании. Благодаря этому менеджер имеет возможность непосредственно наблюдать функционирование моделей, что повышает степень его доверия к результатам моделирования. Данная система обеспечивает создание иерархических моделей, позволяющих описывать процессы с различной степенью детализации. Это гарантирует простоту и естественность при создании сложных моделей крупных компаний (рис. 4.2).



Рис. 4.2. Пример иерархической модели

Все элементы моделей, включая ресурсы процессов, могут модифицироваться непосредственно во время исполнения, результаты изменений можно увидеть сразу же после их введения.

Система ReThink позволяет формировать стоимостные и временные характеристики различных проектов для их объективного сравнения, а также проверять гипотезы типа "что, если". Для анализа работы моделей предусмотрен целый набор инструментов: блоки-датчики для сбора данных, блоки-установщики значений атрибутов сущностей, графики для наглядного отображения результатов моделирования, всевозможные просмотревые табло из стандартных средств комплекса G2. С помощью датчиков можно снимать такие показатели, как длительность цикла обработки сущности на том или ином этапе, стоимость обработки, а также другие свойства, определенные разработчиком модели. Для отсева шумов и выявления тенденций можно использовать специальные блок-фильтры.

Для проверки гипотез "что, если" в системе реализован механизм сценариев. Сценарии позволяют исследовать зависимость поведения одной и той же модели от поведения внешнего мира (например, частота поступления заявок, их сложность и т. д.), а также от каких-либо параметров этой модели (например, количества транспортных средств или численности служащих, занятых оформлением заказов). Варьируемые параметры и измеряемые показатели выносятся на отдельное окно сценария, после чего в результате прогона модели автоматически формируется отчет. Кроме этого, система позволяет использовать сценарии для объективного сравнения альтернативных проектов: один и тот же сценарий, описывающий некоторое заранее заданное поведение внешнего мира, может применяться для прогона различных моделей. Результаты, вынесенные в отчет, являются основой для сопоставления и оценки этих моделей.

Основа данной системы — поддержка коллективной работы с приложениями на основе архитектуры "клиент — сервер" с помощью системы Telewindows комплекса G2. Telewindows обеспечивает множественный доступ к централизованному приложению на сервере с других рабочих станций или с ПК. Коллективная разработка и использование приложений имеют принципиальное значение при проведении глобального реинжиниринга крупной компании или их объединения в рамках отрасли.

Средства стандартных интерфейсов с внешними приложениями (GSI — G2 StandardInterface) комплекса G2 позволяют использовать в моделях реальные данные. Это обеспечивает надежность при тестировании, а кроме того, превращает графические модели, создаваемые средствами системы ReThink, в идеальную основу для мониторинга бизнес-процессов, управления информационными потоками, непрерывного инкрементального реконструирования текущих процессов, а также поддержки принятия решений в оперативном управлении. Полученные приложения будут естественным образом стыковаться с технологическими приложениями диагностики и мониторинга производственных процессов, разработанными на базе комплекса G2.

С помощью интерфейсов GSI возможна стыковка системы ReThink с внешними приложениями. В частности, реализованы стандартные средства сопряжения с СУБД Oracle, Sybase, Rdb и Informix и с системами сбора данных в реальном режиме времени. Имеются средства сопряжения с Visual Basic. Как и инструментальный комплекс G2, система ReThink функционирует на

Unix-рабочих станциях, системах OPEN VMS, а также на ПК в среде Windows.

При создании системы ReThink фирма Gensum не ставила своей целью предложить какую-либо конкретную методологию реинжиниринга. Ее задача — создание удобного универсального средства для реализации различных методологий. Система адресована в первую очередь консалтинговым фирмам и информационным подразделениям крупных компаний для воплощения их оригинальных идей в области реинжиниринга. Предоставлена возможность развития системы средствами инструментального комплекса G2 — вплоть до реализации новых нестандартных средств моделирования и анализа.

Сегодня ReThink используется в ряде компаний, среди которых патентное ведомство США и компания Xerox, осуществившая реинжиниринг отделения по закупке сопутствующих материалов с годовым оборотом в 3 млрд долл. В компании Xerox при проведении реинжиниринга сначала использовался пакет ABC FlowCharter, а построенная модель работы отделения включала 17 процессов и 314 рабочих процедур. Анализ модели показал, что 70% процедур оказались непроизводительными. Затем была разработана новая модель процессов закупки, включающая всего 42 рабочие процедуры. Столкнувшись с таким существенным сокращением количества процедур, руководство компании поставило вопрос о работоспособности новой организации: не возникнут ли перед компанией серьезные непредвиденные проблемы после того, как она сделает основные капиталовложения в реконструкцию отделения? Чтобы обосновать предложенный проект, было решено использовать систему ReThink, с помощью которой предполагалось исследовать ими-

тационную модель планируемой организации работы отделения. В результате несколько процессов пришлось снова перепроектировать, что привело к выигрышу в качестве проекта, а следовательно, снизило риск неудачи при проведении реинжиниринга.

4.3.4. Интеллектуальный анализ данных

Интеллектуальный анализ данных (ИАД) (Data Mining) — это процесс поддержки принятия решений, основанный на поиске в данных скрытых закономерностей (шаблонов информации). При этом накопленные сведения автоматически обобщаются до информации, которая может быть охарактеризована как знания.

В общем случае процесс ИАД состоит из трёх стадий [19]:

- 1) выявление закономерностей (свободный поиск);
- 2) использование выявленных закономерностей для предсказания неизвестных значений (прогностическое моделирование);
- 3) анализ исключений, предназначенный для выявления и толкования аномалий в найденных закономерностях.

Иногда в явном виде выделяют промежуточную стадию проверки достоверности найденных закономерностей между их нахождением и использованием (стадия валидации).

Все методы ИАД подразделяются на две большие группы по принципу работы с исходными обучающими данными.

В первом случае исходные данные могут храниться в явном детализированном виде и непосредственно использоваться для прогностического моделирования и/или анализа исключений; это так называемые методы рассуждений на основе анали-

за прецедентов. Главной проблемой этой группы методов является затрудненность их использования на больших объемах данных, хотя именно при анализе больших хранилищ данных методы ИАД приносят наибольшую пользу.

Во втором случае информация вначале извлекается из первичных данных и преобразуется в некоторые формальные конструкции (их вид зависит от конкретного метода). Согласно предыдущей классификации, этот этап выполняется на стадии свободного поиска, которая у методов первой группы в принципе отсутствует. Таким образом, для прогностического моделирования и анализа исключений используются результаты этой стадии, которые гораздо более компактны, чем сами массивы исходных данных. При этом полученные конструкции могут быть либо "прозрачными" (интерпретируемыми), либо "черными ящиками" (нетрактруемыми).

Оперативная аналитическая обработка и интеллектуальный анализ данных — две составные части процесса поддержки принятия решений. Но сегодня большинство систем OLAP заостряет внимание только на обеспечении доступа к многомерным данным, а большинство средств ИАД, работающих в сфере закономерностей, имеют дело с одномерными перспективами данных. Эти два вида анализа должны быть тесно объединены, то есть системы OLAP должны фокусироваться не только на доступе, но и на поиске закономерностей. Как заметил N. Raden, "многие компании создали прекрасные хранилища данных, идеально разложив по полочкам горы неиспользуемой информации, которая сама по себе не обеспечивает ни быстрой, ни достаточно грамотной реакции на рыночные события".

К. Parsaye вводит составной термин "OLAP Data Mining" (многомерный интеллектуальный анализ) для обозначения такого объединения (рис. 4.3). J. Han предлагает еще более простое название — "OLAP Mining" — и несколько вариантов интеграции двух технологий.

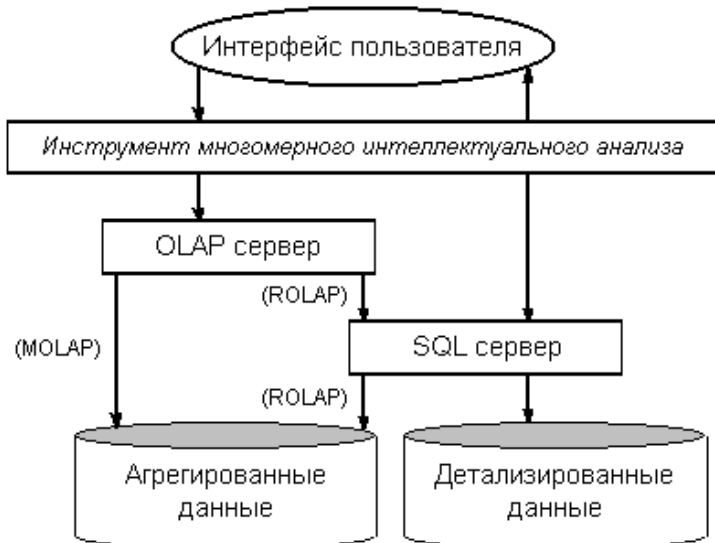


Рис. 4.3. Архитектура системы многомерного интеллектуального анализа данных

4.3.5. Управление решателями задач

Целью управления решателями задач является постоянное приведение их свойств в соответствие с постоянно изменяющимися текущими требованиями, условиями эксплуатации, знаниями предметной области, а также возможными изменениями онтологии предметной области.

Гибкий способ интеграции "Cubing while mining" позволяет автоматически активизировать однотипные механизмы интеллектуальной обработки над результатом каждого шага многомерного анализа (перехода между уровнями обобщения, извлечения нового фрагмента гиперкуба и т. д.).

К сожалению, очень немногие производители предоставляют сегодня достаточно мощные средства интеллектуального анализа многомерных данных в рамках систем OLAP. Проблема также заключается в том, что некоторые методы ИАД (байесовские сети, метод k -ближайшего соседа) неприменимы для задач многомерного интеллектуального анализа, так как основаны на определении сходства детализированных примеров и не способны работать с агрегированными данными.

4.4. РАБОТЫ С ОСНОВНЫМИ ОБЪЕКТАМИ, ПРОЦЕССАМИ И ЯВЛЕНИЯМИ, СВЯЗАННЫМИ С ИНТЕЛЛЕКТУАЛЬНЫМИ СИСТЕМАМИ, И ИСПОЛЬЗОВАНИЕ МЕТОДОВ ИХ НАУЧНОГО ИССЛЕДОВАНИЯ

В настоящее время научные исследования направлены на изучение и построение сложных, больших и слабоформализуемых технических, экологических, экономических, политических и социальных проблем, порождаемых процессом развития цивилизации. По мнению ряда ученых, только использование всего потенциала знаний, накопленных человеком и создаваемых его интеллектом, позволяет успешно решать возникающие проблемы и находить пути адаптации человека к новым условиям его жизни при развитии цивилизации.

Интеллектуальные системы и носители интеллекта традиционно находили применение в различных системах управления, включая ручное и полуавтоматическое управление, но новым этапом развития интеллектуальных систем стало появление электронных вычислительных машин (ЭВМ) и телекоммуникационных сетей с элементами искусственного интеллекта [4]. С момента начала исследований по искусственному интеллекту понималось создание вычислительных систем, обладающих свойствами имитации творческих процессов, логических выводов, восприятие естественно-язычных запросов и команд, аккумуляции знаний в ЭВМ. В качестве начальных научных направлений исследований новой информационной технологии можно выделить работы по интеллектуальным информационно-поисковым системам, обеспечивающим в процессе диалога человека с ЭВМ, пользователей-непрограммистов с базами данных и знаний на профессиональных языках пользователей, близких к естественному языку.

4.4.1. Структура систем интеллектуального управления

Построение структуры системы интеллектуального управления связано в первую очередь с построением модели системы, в которой должны быть определены как традиционные элементы системы управления, так и модели обработки знаний, реализуемые интеллектуальной системой. В интеллектуальной системе управления новыми элементами по сравнению с традиционной системой управления являются все интеллектуальные преобразования или элементы управления знаниями, которые связаны с реализацией искусственного интеллекта, т. е. с

использованием технологий экспертных систем, базы знаний, принятия решений, ассоциативной памяти, нечеткой логики, семиотических сетей, управления структурной динамикой и т. п.

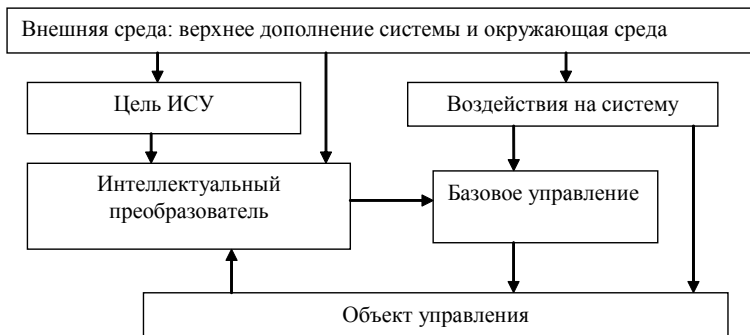


Рис. 4.4. Обобщенная схема системы интеллектуального управления

Анализируя принятые структуры систем управления с решающими устройствами [22], можно и для обобщенной интеллектуальной системы использовать аналогичную структуру (рис. 4.4), которая взаимодействует с внешней средой и в процессе получения от нее необходимой информации формирует цель действия и анализирует воздействия на систему (физические и информационные). Определяющими элементами системы управления в этом случае являются: интеллектуальный преобразователь и базовая система управления.

Можно заметить, что использование интеллекта человека строится на основе рассмотренной структуры, когда человек участвует в управлении в качестве интеллектуального преоб-

разователя, согласованного с внешней средой через специализированные датчики, и реализует воздействие на систему управления через ручку управления или интерфейс взаимодействия с компьютером. В практике управления подвижными объектами такие системы получили специальное название. Системы управления подвижными объектами (в авиации, в космической технике, автомобиле и других транспортных средствах) получили название систем полуавтоматического управления, когда используется способность человека наблюдать и оценивать ситуации, возникающие при движении объектов, и формировать непрерывное управление ими. В системах ручного управления на человека-оператора возлагаются дополнительные функции по управлению движением подвижным объектом, связанные с тем фактом, что он полностью осуществляет функции базового управления и воздействует на органы управления. В ручных системах управления человек-оператор рассматривается как звено управления, формирующее закон и программу управления. В автоматизированных системах управления (автоматизированном управлении космическим кораблем, атомной электростанцией и другими подобными системами управления) человек-оператор оказывает воздействие на базовую автоматическую систему через вычислительные системы, что адекватно может быть описано предлагаемой структурой.

В случае использования в системе управления искусственного интеллекта в качестве интеллектуального преобразователя реализуются [18]:

- экспертные системы [11, 15];
- ситуационное управление [12, 15];

- управление структурной динамикой сложных технологических [3] и других интеллектуальных систем и их элементы.

Интересным примером использования интеллектуального преобразователя в системе управления является использование динамической экспертной системы.

Математическая модель интеллектуальной системы управления состоит из трех частей:

- интеллектуального преобразователя (экспертной системы, включающей базы данных и знаний);
- объекта управления;
- управляющих устройств системы (вычислительных и преобразующих и исполнительных устройств).

Интеллектуальный преобразователь представляет собой логико-преобразующее устройство, которое преобразовывает информацию о внешней среде и объекте управления и трансформирует в сигналы Y — сигналы воздействия на управляющие устройства системы [18]. Математическая модель интеллектуального преобразователя оператором вида

$$Y = F(x, u, w, p, z), \quad (4.1)$$

где $F(\cdot)$ — некоторый оператор интеллектуального преобразования, характеризующий структуру или работу интеллектуального преобразователя; x — вектор состояния системы управления; u — вектор управления; w — вектор воздействий внешней среды; p — вектор сигналов цели; z — вектор параметров объекта.

Объект управления в достаточно общем случае описывается уравнениями вида

$$\dot{x} = f(x, u, w, z, t), \quad y = C(x), \quad x(t_0) = x_0, \quad t \geq t_0, \quad (4.2)$$

где $f(.)$ — вектор-функция, описывающая объект управления; $C(.)$ — заданная функция выходных сигналов; t — координата времени; y — вектор выхода или измерений.

Управляющие устройства системы (вычислительных и преобразующих и исполнительных устройств) формируют управляющие воздействия на объект управления u из множества его возможных значений в соответствии с решаемой задачей для достижения сформированной интеллектуальным преобразователем цели.

Для формирования воздействий на систему управления объектом в интеллектуальном преобразователе используется блок принятия решения, который может быть рассмотрен как самостоятельный элемент. Блок принятия решений формируется на основе теории принятия решений [25–27].

4.4.2. Модели принятия решения в условиях конфликта

Основные этапы процесса принятия решения согласно теории принятия решения декомпозируются на следующие этапы [27]:

- определение цели решения возникшей проблемы;
- выбор наиболее предпочтительного варианта действий, ведущего к достижению цели;
- реализация решения (выбранного варианта действия).

Определение цели решения возникшей проблемы реализуется в блоке интеллектуального преобразователя, получающего и обрабатывающего информацию о внешней среде с системы датчиков. В условиях конфликта цель может зависеть от имеющихся ресурсов и факторов, которые образуют *проблемную ситуацию*, т. е. ситуацию принятия решения в условиях

конфликта. Способ действия для управления объектом в процессе принятия решения называют *стратегиями*, а результат, к которому может привести выбранная стратегия, — *исходом*. Условия конфликта порождают факторы, воздействующие на стратегию и соответственно на управление, реализуемое интеллектуальной системой. С точки зрения наличия информации об условиях конфликта факторы разделяются на две группы:

- *определенные* (фиксированные) факторы, значения которых известны;
- *неопределенные* факторы, о которых априорно не известно, какое значение они примут.

В зависимости от происхождения неопределенные факторы делятся на *случайные* и *неопределенные нестохастического характера*, состоящие из *природных* и *стратегических*. Математическая модель принятия решений формируется с учетом всех факторов и имеющейся о них информации. Упрощенная модель принятия решения в этом случае может быть описана следующей системой:

$$D_0 = \langle Y, G, U, L, J, \Omega \rangle, \quad (4.3)$$

где Y — множество исходов (результатов); G — модель предпочтений исходов (принимаемых решений); U — множество стратегий принятия решений; L — множество возможных значений неопределенных факторов; J — функция, определяющая взаимосвязь неопределенного фактора и исход, получаемый в результате принятого решения; Ω — вся остальная информация о принимаемом решении в формализованном виде (сведения о конфликте, предпочтения других лиц, участвующих в конфликте, и др.).

Удобство использования модели (4.3) в условиях конфликта определяется тем положением, что она позволяет просто и наглядно связать значения неопределенных факторов и стратегий с управлением, реализуемым интеллектуальной системой. Множества Y, G, U, L и функция J формально задают компоненты принимаемого решения и определяют связь с системой управления через понятия критерия и показателей эффективности системы. В теории управления наиболее часто отношения предпочтения описываются с помощью специальных функций: показателей качества и критериев. Под *показателем* W качества или эффективности системы управления понимается мера степени соответствия реального результата управления Y требуемому Y^{mp} для достижения цели и получения оценок или измерений интенсивности исходов. Под *критерием* K понимается правило, позволяющее сопоставлять принимаемые решения и стратегии с точки зрения выбранных показателей оценок исходов. Критерии вводятся на основе определенной концепции рационального поведения интеллектуальной системы [25]: пригодности, оптимизации и адаптивности.

Более общей моделью принятия решения в условиях конфликта по сравнению с (4.3) является модель динамической системы интеллектуального управления на основе моделей теории игр [28, 29] и современного аппарата функционального анализа [30].

Теория игр как раздел математики в настоящее время стала теорией математических моделей [29]. При этом под *конфликтом* понимается явление, которое связано с ответом на вопросы:

Кто и как в этом явлении участвует?

Каковы возможные исходы конфликта?

Кто в этих конфликтах заинтересован?

В чем состоит заинтересованность участников конфликта?

Для описания конфликта вводятся понятия:

- *коалиций действия* R_d , объединяющих множество *игроков* (участников конфликта) по их действиям;

- *коалиций интересов* R_i , объединяющих игроков по интересам;

- *стратегий*, характеризующих решения коалиций K (действий R_d и коалиций интересов R_i , которые в зависимости от типа игры могут иметь одних и тех же игроков или образовываться из разных игроков);

- *отношений предпочтения* G как абстрактного бинарного отношения на множестве всех стратегий (нередко отношения предпочтения задаются функцией выигрыша W_k , и тогда коалицию K , если она предпочитает ситуацию x ситуации y , обозначают через отношение предпочтения в виде $x G_k y$ или при использовании функции выигрыша, если $W_k(x) > W_k(y)$, в виде $x W_k y$).

Тогда формальное описание конфликта состоит в задании системы

$$\Gamma = \langle R_d, S_{R_d}, S, R_i, G_{R_i} \rangle, \quad (4.4)$$

где S_{R_d} — множество стратегий коалиции действия; G_{R_i} — множество отношений коалиции интересов.

В работе [30], как и в ряде других исследований по общей системной теории управления, предложено рассматривать динамические системы управления, состоящие из объекта управления, характеризуемого некоторым множеством состояний, и регулятора, под которым понимается математическая модель,

состоящая из элемента, обеспечивающего оценку состояния объекта, и элемента, формирующего управление.

Для описания динамической системы используется математическая модель совокупности элементов, удовлетворяющих следующим основным аксиомам.

А. Заданы множества моментов времени T , множество состояний системы X . Множество значений входных воздействий U , непустое множество их допустимых значений:

$$\Omega = \{\omega: T \rightarrow U\}, \quad (4.5)$$

множество значений выходных величин Y и множество их допустимых значений:

$$Y = \{\gamma: T \times X \rightarrow Y\}. \quad (4.6)$$

В. Множество T есть некоторое упорядоченное подмножество множества вещественных чисел (направление времени).

С. Существует *переходная функция*:

$$\Phi = \{\varphi: T \times T \times X \times \Omega \rightarrow X\}, \quad (4.7)$$

где состояния системы $x(t) = \varphi[t, t_0, x(t_0), \omega] \in X$.

Д. Задано входное отображение:

$$Y = \{\gamma: T \times X \rightarrow Y\}. \quad (4.8)$$

Приведенная модель в наиболее общем виде динамического объекта или системы изображена на рисунке 4.5.

Для описания динамической системы используются дополнительные термины. Так, состояние системы x в момент времени t или пара элементов множества $T \times X$ называется *событием* (или *фазой*) динамической системы. Множество $T \times X$ называется *пространством событий*, или *фазовым пространством*. В том случае, когда множество выходных воздействий используется для управления, оно называется управлением.

Управление переводит систему из одного состояние в некоторое другое. При этом система находится в *движении*, описывая в пространстве состояния *траекторию*. В рассмотренной формализации динамическая система рассматривается как объект управления.

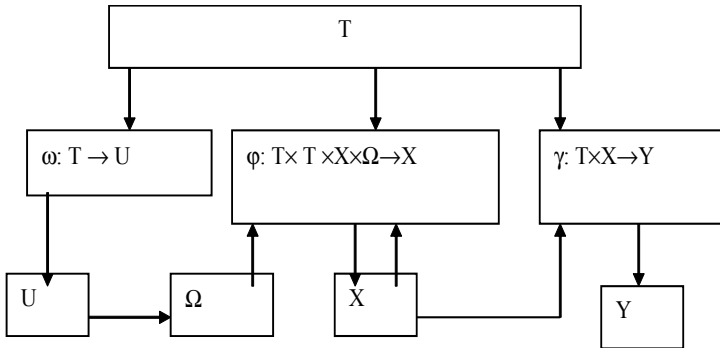


Рис. 4.5. Множественная схема динамической системы

Для реализации системы управления в классической теории систем выделяют объект управления и регулятор управления, состоящий в простейшем случае из элемента оценки состояния системы и элемента, формирующего закон управления. Так как объект управления нами уже рассмотрен как динамическая система, то введем понятия *закона управления* и *оценки состояния* системы.

Законом управления принято называть отображение

$$k : T \times X \rightarrow U, \tag{4.9}$$

где $u(t) = k[t; x(t)]$ — величины управления, принадлежащие множеству U .

Для реализации управления необходимо знание переменных состояния системы $\mathbf{x}(t)$, что требует операции определения обратного отображения:

$$\gamma^{-1} : Y \rightarrow X, \quad (4.10)$$

и тогда координаты системы определяются из условия

$$\mathbf{x}(t) = \gamma^{-1}[\mathbf{y}(t)]. \quad (4.11)$$

Кроме того, для оценки состояния системы необходимо оценивать точность определения $\mathbf{x}(t)$, т. е. получать оценку состояния $\mathbf{x}_0(t)$.

Структурно-логическая схема системы управления, состоящая из объекта управления и регулятора, показана на рисунке 4.6.

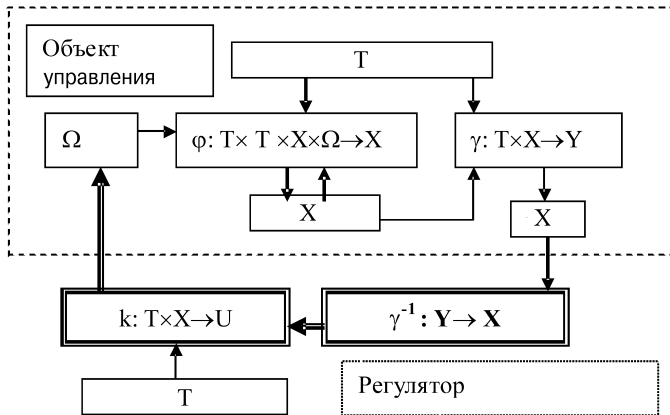


Рис. 4.6. Структурно-логическая схема блока основного управления системы

Для учета факторов воздействия внешней среды и конфликта необходимо провести учет воздействий этих объектов. При этом факторы внешней среды и условий конфликта необ-

ходимо рассматривать и описывать с позиций динамической системы.

При принятии решения в системе интеллектуального управления возникает необходимость объединения интеллектуального преобразователя, включающего блок принятия решения, и динамической системы управления.

4.4.3. Определение оптимальной интеллектуальной системы принятия решения и управления в условиях конфликта

Понятие оптимальности получило широкое распространение в теории управления и теории принятия решений. Под оптимальной системой автоматического управления понимают "наилучшую" систему, которую выбирают из множества систем по принятому показателю качества системы или эффективности ее функционирования. При этом система является оптимальной, если она обеспечивает экстремум принятого показателя. В зависимости от конкретного вида показателя понятие оптимальности связывается с критерием, т. е. с минимумом или максимумом показателя. Если, например, в качестве показателя используется средняя квадратическая ошибка системы, то оптимальной системой считают ту, которая реализует критерий — минимум средней квадратической ошибки. Если же в качестве показателя используется вероятность невыхода ошибки системы из заданных допусков, то оптимальной считают ту систему, которая реализует критерий — максимум невыхода ошибки системы из заданных допусков.

Принятие решений в условиях определенности (фиксированности) факторов и в случае, когда неопределенность является случайной и о факторе имеется полная априорная инфор-

мация, то оптимальность решения определяется аналогично тому, как это делается в системах автоматического управления.

В условиях стратегической (поведенческой) неопределенности, которая появляется в условиях конфликта, понятие оптимальности принимаемого решения значительно труднее поддается формализации. Теория математических моделей принятия оптимальных решений составляет значительную часть науки, которая получила название *исследования операций* [1]. Особое место в исследовании операций занимает раздел, занимающийся теорией математических моделей принятия оптимальных решений в условиях конфликта, который получил название *теории игр* [29]. Теория игр, как теория математических моделей, базируется на использовании формальных, знаковых моделей для описания конфликта, а также использует формальные средства их анализа. В теории игр были успешно реализованы определения понятия конфликта и принятия решения. В настоящее время понятие оптимальности удалось реализовать только для части игр. Поэтому при анализе систем принятия решений и управления мы ограничимся рассмотрением только тех конфликтов, которые описываются моделями бескоалиционных (множества коалиций действия и интересов совпадают, определены функции выигрыша) и антагонистических (число игроков равно двум, а значения их функций выигрыша в любой ситуации равны и противоположны по знаку) игр. Для бескоалиционных и антагонистических игр условия оптимальности основываются на понятии равновесия.

Вопрос об оптимальности управления и принятия решений в интеллектуальных системах усложняется и тем фактом, что для реализации системы используются экспертные знания, ко-

торые не всегда удается реализовать. Эта трудность преодолевается за счет допущения, что в системе реализуется интеллектуальный датчик, который однозначно определяет информацию о действиях другой стороны за счет физических измерений действия противоположных игроков, воздействующих на систему управления и принятия решения в условиях конфликта. Фактически в интеллектуальной системе реализуется рациональное управление и принятие решения. Достоверность и степень приближения к точному оптимальному управлению и принятию решения определяется той информацией, которую получает, обрабатывает и выдает интеллектуальная система или интеллектуальный датчик. При этом в результате ошибок интеллектуального датчика, связанных с использованием экспертных знаний, появляются риски, связанные с ошибочным формированием цели принятия решения и управления, определением показателей и критериев оценки эффективности работы системы, выбором лучшей системы. Для уменьшения рисков в работе интеллектуальной системы используются методы обучения и самообучения, которые в настоящее время начинают разрабатываться в интеллектуальных системах и которые всегда присутствовали в биологическом интеллекте [17–21].

Решение задач синтеза систем оптимального управления и принятия решения в условиях конфликта традиционно связано с математическими методами отыскания экстремумов функций и функционалов, нахождения равновесных состояний, что позволяет решать задачи синтеза систем управления и принятия решений в практически важных случаях.

Рассмотрение структуры систем интеллектуального принятия решения и управления в условиях конфликта (рис. 4.7)

показывает, что система состоит из трех основных элементов (подсистем), каждый из которых имеет свои модели и специфические методы исследования:

- интеллектуальная подсистема;
- подсистема принятия решения;
- подсистема управления объектом и объект управления.

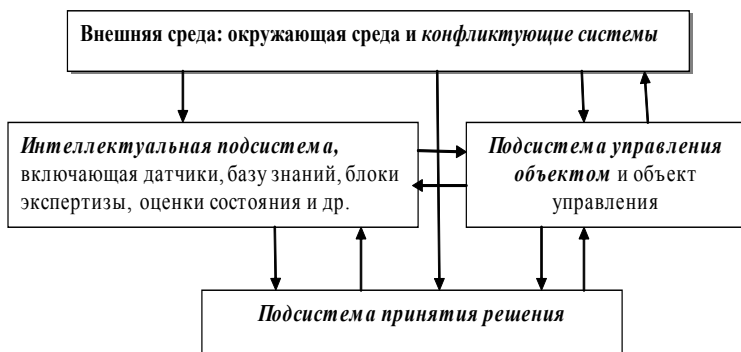


Рис. 4.7. Структура интеллектуальной системы принятия решения и управления в условиях конфликта, порождаемого конфликтующими системами

Несмотря на сложность и методологические особенности элементов интеллектуальных систем принятия решения и управления в условиях конфликта при определении оптимальной системы, удается выделить следующие основные задачи:

- построение модели условий конфликтного взаимодействия оптимизируемой системы с другими системами, изучение и описание информации о воздействиях на систему;

- формирование интеллектуальной подсистемы, включающей датчики, базу знаний, блоки экспертизы, оценки состояния, формирования целей, выбор показателей и критериев оптимизации системы и других элементов интеллектуальной деятельности;
- принятие решений, обеспечивающих оптимальное противодействие одной или нескольким конфликтующим системам;
- определение и математическое описание классов допустимых систем управления объектом управления;
- отыскание экстремума критерия или точек равновесия в игровых задачах, а также соответствующих им характеристик.

Формирование модели условий конфликтного взаимодействия оптимизируемой системы с другими системами описывается в простейшем случае с позиции теории игр как конфликт двух игроков (**A**, **B**). При этом конфликт рассматривается как операция, в которой игроки имеют различные цели, и реализуют свою деятельность и выбирают свои стратегии в соответствии со своими целями (рис. 4.8) [1, 24–26].

Исследование операции всегда проводится с точки зрения одного игрока, а в рассматриваемом случае — с позиции интеллектуальной системы принятия решения и управления (для определенности примем, что в конфликте это игрок **A**).

Будем считать, что эффект достижения цели определяется векторными показателем $\mathbf{W} = (\mathbf{W}_A, \mathbf{W}_B)$ и критерием $\mathbf{K} = (\mathbf{K}_A, \mathbf{K}_B)$, связанными с целями поведения сторон. При этом в зависимости от наличия датчиков получения информации о целях действия игроков предполагается, что идентификация целей

позволяет интеллектуальной системе определять критерии и показатели, используемые игроками для идентификации показателей и критериев игроков.

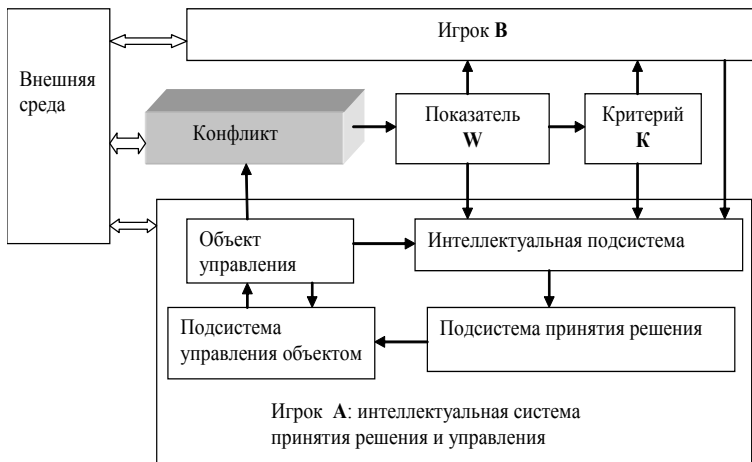


Рис. 4.8. Общая схема конфликта как операции двух систем с разными целями

Вопросы формирования интеллектуальной подсистемы, включающей датчики, базу знаний, блоки экспертизы, оценки состояния, формирования целей, выбор показателей и критериев оптимизации системы и других элементов интеллектуальной деятельности были рассмотрены ранее, когда анализировались интеллектуальные системы и их элементы.

Принятие решений, обеспечивающих оптимальное противодействие одной или нескольким конфликтующим системам, основывается в условиях конфликта на использовании моде-

лей, методов и алгоритмов принятия оптимальных решений в теории игр. В рассматриваемом нами случае мы ограничиваемся матричными антагонистическими (игры двух лиц с нулевой суммой), которые имеют развитый инструментарий решения прикладных задач.

Определение и математическое описание классов допустимых систем управления объектом управления основывается на теории систем управления [1, 31]. В рассматриваемой задаче оптимизации системы управления объектом используется подход, основанный на использовании корреляционной теории статистической оптимизации систем, разработанной профессором Н. И. Андреевым [22, 32]. В его работах разработаны методы исследования сложных линейных и нелинейных динамических систем, подверженных случайным воздействиям. При постановке задач определения оптимальных динамических систем по статистическим критериям (по минимуму средней квадратической ошибки системы, максимуму вероятности невыхода ошибки системы из заданных допусков и др.) им разработана методология учета ограничений типа неравенств, накладываемых на функции управления и вектора. Большое внимание он уделял исследованию динамических систем с заданной структурой, так как для прикладных задач эти системы, с одной стороны, играют важную роль, а с другой — приводят к значительному усложнению математических задач оптимизации этого класса систем. В его работах исследован ряд новых задач, относящихся к оценке параметров системы, фильтру Калмана, синтезу нелинейных динамических систем, построению адаптивных систем оценивания и управления, развитию методов принятия ста-

тистических решений, определению оптимальных систем управления с решающими устройствами.

Проиллюстрируем проведенный анализ постановки задачи определения оптимальной интеллектуальной системы принятия решения и управления в условиях конфликта на простом примере, который был рассмотрен в работе [33] в иной постановке задачи оптимизации.

Пример. Рассмотрим задачу определения интеллектуальной системы принятия решения и управления, на вход которой поступает сигнал, представляющий сумму полезного сигнала $G(t)$ и помехи $Z(t)$. При этом будем предполагать, что слежение за полезным сигналом осуществляет интеллектуальная система управления (игрок **A**), а помеху формирует противоположная сторона (игрок **B**), которая выбирает характеристики такими, чтобы ошибки следящей системы были большими. В результате возникает конфликт, который часто рассматривается в теории игр. Для наглядности и простоты изложения будем полагать, что каждая из сторон имеет две стратегии управления и по этой причине можно ограничить решение задачи на основе матричной и биматричной игр.

Пусть помеха представляет собой нормально распределенный белый шум с интенсивностями α_1 , α_2 , выбором величины которых управляет игрок **A**. Полезный сигнал $\mathbf{G} = \mathbf{G}_0$ — числовая случайная величина, принимающая значения β_0 и $-\beta_0$ с вероятностью 0,5, величина которых формируется игроком **A** и может принимать значения β_{01} , β_{02} .

Желаемая выходная величина формируемой системы слежения — полезный сигнал. Система выбирается на классе ста-

ционарных систем, время регулирования которых не превосходит время T .

В качестве показателя качества системы принимается суммарная ошибка системы D_{Σ} , которая определяется биномиальным распределением полезного сигнала D_1 и случайным характером помехи D_2 . Следуя работе [32], запишем ошибки системы в следующем виде:

$$D_1 = \int_0^T \int_0^T \delta(\tau_1 - \tau_2) \omega(\tau_1) \omega(\tau_2) d\tau_1 d\tau_2,$$

где $\delta(\tau_1 - \tau_2)$ — дельта-функция, принимающая нулевое значение, если $(\tau_1 - \tau_2) \neq 0$, и равная ∞ , если $(\tau_1 - \tau_2) = 0$; $\omega(\tau_1)$ — весовая функция при переменной τ_1 ; $\omega(\tau_2)$ — весовая функция при переменной τ_2 ;

$$D_2 = (\beta_0)^2 \int_0^T \omega(\tau) d\tau - 1)^2;$$

$$D_{\Sigma} = D_1 + D_2.$$

Базовая подсистема управления будет описываться весовой функцией, параметры которой зависят от решения игровой задачи и выбора подсистемой интеллектуального управления цели системы и соответствующих показателей и критериев оптимизации.

Тогда для иллюстрационного примера примем $T = 1$. Выражение для весовой функции можно записать в виде

$$\omega_{ij}(\tau) = \beta^2_{0j} (\alpha_i + \beta^2_{0j})^{-1},$$

где управление реализуется в виде весовых функций в зависимости от стратегий игроков (\mathbf{B} максимизирует дисперсию ошибки, а \mathbf{A} минимизирует ошибку слежения).

В этом случае дисперсии для принятой весовой функции записываются в следующем виде, учитывающем стратегии игроков:

$$D_1^{IJ} = \alpha_i (\alpha_i \beta_{0j}^{-2} + 1)^{-2},$$
$$D_2^{IJ} = \alpha_i^2 \beta_{0j}^2 (\alpha_i + \beta_{0j}^2)^{-2}.$$

Интеллектуальная подсистема **B**, наблюдая с помощью своих информационных и физических датчиков за игроком **A**, определяет его действия и формирует свою стратегию движения в каждой конкретной ситуации. Возникает задача ситуационного управления движением, когда решения и управления должны осуществляться в соответствии с той ситуацией, которая складывается в текущий момент конфликта.

Одной из сложных задач в этом случае является задача определения типа антагонизма между игроками и определения целей действия систем, выяснения их интересов. В практике конфликтных ситуаций о действиях другой стороны или игрока, как правило, используется информация, получаемая через информационные сообщения, телекоммуникационные сети и путем сбора различными способами конфиденциальной информации. При реализации интеллектуальной экспертной системы используются эксперты и их оценки, но возможен и аналитический подход к решению идентификации намерений и цели противоположной стороны на основе анализа конфликта. Если реализуется интеллектуальная подсистема на основе ситуационного управления, то возможна декомпозиция общей задачи на две части, когда интеллектуальная подсистема идентифицирует конкретную ситуацию, а управление в конкретной ситуации формируется на основе решения игровых задач для определения неопределенных стратегий или неопределенных пара-

метров противодействующей стороны. Вместе с тем интеллектуальная подсистема должна определять тип модели игры в конфликте. Возникает вопрос: как это может реализовать интеллектуальная подсистема? Существует ли такая возможность?

Покажем возможность оценки действий противника на основе обработки информации по решению задачи с помощью теории игр. Для этого проанализируем результаты принятия решения при реализации матричной антагонистической игры. Вначале рассмотрим влияние конфликта на управление на примере единичной ситуации, когда время управления T фиксировано и единственное, а затем рассмотрим условную модель ситуационного управления на основе сценария, в котором достижение цели обеспечивается несколькими ситуациями управления в разные моменты времени T_v .

Примем, что игроки могут реализовать по две стратегии, т. е. конфликт описывается матрицей размерности 2×2 , где $\beta^2_{01} = 1$, $\beta^2_{02} = 2$ — значения стратегий игрока B , $\alpha_1 = 1$, $\alpha_2 = 2$ — значения стратегий игрока A . В этом случае при принятых исходных данных матрицы выигрышей можно записать в следующем виде:

при анализе системы по дисперсии D_2

Стратегии $B \setminus A$	$\beta^2_{01} = 1$	$\beta^2_{02} = 2$	Min по j	Max min =
$\alpha_1 = 1$	0,5	0,667	0,5	0,5
$\alpha_2 = 2$	0,333	0,4	0,4	
Max по i	0,5	0,667		
Min max =	0,5			

при анализе системы по дисперсии D_1

Стратегии В \ А	$\beta^2_{01} = 1$	$\beta^2_{02} = 2$	Min по j	Max min =
$\alpha_1 = 1$	0,25	0,011	0,011	
$\alpha_2 = 2$	0,011	0,08	0,08	0,08
Max по i	0,011	0,08		
Min max =	0,08			

при анализе системы по дисперсии D_Σ

Стратегии В \ А	$\beta^2_{01} = 1$	$\beta^2_{02} = 2$	Min по j	Max min =
$\alpha_1 = 1$	0,75	0,678	0,678	
$\alpha_2 = 2$	0,344	0,48	0,48	0,48
Max по i	0,75	0,48		
Min max =	0,48			

Все три антагонистические игры имеют седловые точки:

$$\max_i \min_j D_{\mu}^{ij} = \min_j \max_i D_{\mu}^{ij}, \quad \mu=(1, 2, 3), \quad i=J=(1, 2)$$

в чистых стратегиях, но цены игр у них разные. Интересным результатом анализа дисперсий ошибок анализируемой системы является то, что гарантированная суммарная ошибка системы (равна 0,48) меньше гарантированной ошибки, получаемой только при учете нормальной составляющей ошибки (равной 0,5). Понятно, что полученное уменьшение дисперсии суммарной ошибки в результате отказа от стратегии α_1 , которая ведет к увеличению дисперсии нормальной составляющей ошибки. Это также связано с тем, что дисперсии ошибок от изменения сигнала помехи существенно меньше при α_2 влияют на дисперсию D_1 . По этой причине если игрок **А** будет выбирать свою

стратегию из условия увеличения дисперсии помехи, то он может быть наказан игроком В, который в этом случае может реализовать свою стратегию $\beta_{01}^2 = 1$ (провести адаптивное управление), благодаря этому получить дополнительный выигрыш, который будет равен 0,344, т. е. меньше гарантированного выигрыша на 0,135 (уменьшает дисперсию на 28,5%). Таким образом интеллектуальная подсистема на основе получения оценок интенсивности шума может определить, какой стратегии придерживается игрок А при противодействии игроку В. Логика рассуждений интеллектуальной подсистемы может быть продолжена в том направлении, если игрок В откажется от использования гарантированных оценок, а будет применять критерии: минимаксного сожаления Сэвиджа, пессимизма-оптимизма Гурвица, "недостаточного основания" Бернулли [25, 27].

Рассмотрим теперь случай, когда для достижения цели управления необходимо выполнение нескольких операций по слежению согласно определенному сценарию (рис. 4.9), в котором последовательно реализуются три ситуации со временем управления $T_1=1$, $T_2=2$, $T_3=3$, а также показано начало работы системы и достижение цели в случае успешного выполнения всех планируемых операций.

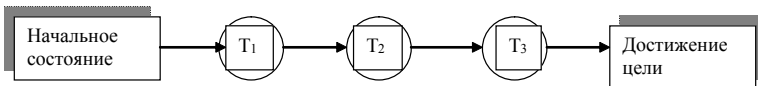


Рис. 4.9. Сценарий на основе сетевой модели, вершины которой соответствуют фактам, а дуги — связям

В качестве критерия эффективности достижения цели примем условие получения оптимальных оценок в каждой ситуации, как это было сделано в первой ситуации при T1.

Для произвольного T_v дисперсии и определяемые весовые функции будут равны:

$$D_{1^{ij}} = T_v \alpha_i (\alpha_i \beta_{0j}^2 + T_v)^{-2},$$

$$D_{2^{ij}} = \alpha_i^2 \beta_{0j}^2 (\alpha_i + T_v \beta_{0j}^2)^{-2},$$

$$\omega_{ij}(\tau) = \beta_{0j}^2 (\alpha_i + T_v \beta_{0j}^2)^{-1},$$

где $v = (1, 2, 3)$, $i = (1, 2)$, $J = (1, 2)$, если использовать данные, принятые в начальных расчетах дисперсий и весовой функции при T1.

Одним из интересных предложений по развитию интеллектуальной подсистемы для оценки степени антагонизма между игроками и соответственно между интеллектуальными системами в условиях конфликта стало использование теории матричных игр.

ВИДЕОРОЛИКИ ПО ГЛАВЕ

<p>Интеллектуальная система управления транспортными потоками http://www.youtube.com/watch?v=_aJEt5UDS5U</p>	
<p>Интеллектуальная система "Умный дом" http://www.youtube.com/watch?v=GfdoLiuI82w</p>	
<p>Интеллектуальная система кормления животных http://www.youtube.com/watch?v=WbKIEHKPpW0</p>	
<p>Сравнение Google Search в Android 4.1 и Siri http://habrahabr.ru/post/146962/</p>	

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите основные направления развития диалоговых интерфейсов интеллектуальных систем.
2. Приведите примеры применения систем распознавания голоса.
3. Приведите примеры интеллектуальных систем с биологической обратной связью.

4. Приведите примеры интеллектуальных систем виртуальной реальности.
5. Интеллектуальный анализ данных.
6. Перечислите основные направления эволюционного моделирования.
7. Объясните принцип работы генетического алгоритма.
8. Основные этапы технологии генетического программирования.

5. ОСОБЕННОСТИ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ЭКСПЕРТНЫХ СИСТЕМ

5.1. НАЗНАЧЕНИЕ, КЛАССИФИКАЦИЯ И ПРИНЦИПЫ ПОСТРОЕНИЯ ЭКСПЕРТНЫХ СИСТЕМ

5.1.1. Назначение экспертных систем

В начале 1980-х годов в исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название "экспертные системы" (ЭС). Цель исследований по ЭС состоит в разработке программ, которые при решении задач, трудных для эксперта-человека, получают результаты, не уступающие по качеству и эффективности решениям, получаемым экспертом. Исследователи в области ЭС для названия своей дисциплины часто используют также термин "инженерия знаний", введенный Е. Фейгенбаумом как "привнесение принципов и инструментария исследований из области искусственного интеллекта в решение трудных прикладных проблем, требующих знаний экспертов".

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

- технология экспертных систем существенно расширяет круг практически значимых задач, решаемых на компьютерах, решение которых приносит значительный экономический эффект;
- технология ЭС является важнейшим средством в решении глобальных проблем традиционного программиро-

вания: длительность и, следовательно, высокая стоимость разработки сложных приложений;

- высокая стоимость сопровождения сложных систем, которая часто в несколько раз превосходит стоимость их разработки; низкий уровень повторной используемости программ и т. п.;
- объединение технологии ЭС с технологией традиционного программирования добавляет новые качества к программным продуктам. Это достигается за счет: обеспечения динамичной модификации приложений пользователем, а не программистом; большей "прозрачности" приложения (например, знания хранятся на ограниченном ЕЯ, что не требует комментариев к знаниям, упрощает обучение и сопровождение); лучшей графики; интерфейса и взаимодействия.

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

- ЭС будут играть ведущую роль во всех фазах проектирования, разработки, производства, распределения, продажи, поддержки и оказания услуг;
- технология ЭС, получившая коммерческое распространение, обеспечит революционный прорыв в интеграции приложений из готовых интеллектуально взаимодействующих модулей;
- ЭС предназначены для так называемых неформализованных задач, т. е. ЭС не отвергают и не заменяют традиционного подхода к разработке программ, ориентированного на решение формализованных задач.

Неформализованные задачи обычно обладают следующими особенностями:

- ошибочностью, неоднозначностью, неполнотой и противоречивостью исходных данных;

- ошибочностью, неоднозначностью, неполнотой и противоречивостью знаний о проблемной области и решаемой задаче;
- большой размерностью пространства решения, т. е. перебор при поиске решения весьма велик;
- динамически изменяющимися данными и знаниями.

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

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

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

цветическое производство, космос, металлургия, горное дело, химия, образование, целлюлозно-бумажная промышленность, телекоммуникации и связь и др.

Коммерческий успех к фирмам-разработчикам систем искусственного интеллекта (СИИ) пришел не сразу. На протяжении 1960–1985 годов успехи ИИ касались в основном исследовательских разработок, которые демонстрировали пригодность СИИ для практического использования. Начиная примерно с 1985 года (в массовом масштабе с 1988–1990 годов) в первую очередь ЭС, а в последние годы системы, воспринимающие естественный язык (ЕЯ-системы), и нейронные сети (НС) стали активно использоваться в коммерческих приложениях.

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

технологическими тенденциями традиционного программирования, что снимает проблемы, возникающие при создании интегрированных приложений.

Причины, приведшие СИИ к коммерческому успеху, следующие.

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

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

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

Архитектура "клиент — сервер". Разработаны ИС ИИ, поддерживающие распределенные вычисления по архитектуре "клиент — сервер", что позволило:

- снизить стоимость оборудования, используемого в приложениях;
- децентрализовать приложения;

- повысить надежность и общую производительность (так как сокращается количество информации, пересылаемой между ЭВМ, и каждый модуль приложения выполняется на адекватном ему оборудовании).

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

5.1.2. Классификация экспертных систем

Общепринятая классификация экспертных систем отсутствует, однако наиболее часто экспертные системы различают по назначению, предметной области, методам представления знаний, динамичности и сложности (рис. 5.1).

По **назначению** классификацию экспертных систем можно провести следующим образом:

- диагностика состояния систем, в том числе мониторинг (непрерывное отслеживание текущего состояния);
- прогнозирование развития систем на основе моделирования прошлого и настоящего;
- планирование и разработка мероприятий в организационном и технологическом управлении;
- проектирование или выработка четких предписаний по построению объектов, удовлетворяющих поставленным требованиям;

- автоматическое управление (регулирование);
- обучение пользователей и др.



Рис. 5.1. Классификация экспертных систем

По **предметной области** наибольшее количество экспертных систем используется в военном деле, геологии, инженерном деле, информатике, космической технике, математике, медицине, метеорологии, промышленности, сельском хозяйстве, управлении процессами, физике, филологии, химии, электронике, юриспруденции.

Классификация экспертных систем по **методам представления знаний** делит их на традиционные и гибридные. Традиционные экспертные системы используют в основном эм-

пирические модели представления знаний и исчисление предикатов первого порядка. Гибридные экспертные системы используют все доступные методы, в том числе оптимизационные алгоритмы и концепции баз данных.

По **степени сложности** экспертные системы делят на поверхностные и глубинные. Поверхностные экспертные системы представляют знания в виде правил "ЕСЛИ, ТО". Условием выводимости решения является безразрывность цепочки правил. Глубинные экспертные системы обладают способностью при обрыве цепочки правил определять (на основе метазнаний), какие действия следует предпринять для продолжения решения задачи. Кроме того, к сложным относятся предметные области, в которых текст записи одного правила на естественном языке занимает более 1/3 страницы.

Классификация экспертных систем по **динамичности** делит экспертные системы на статические и динамические. Предметная область называется статической, если описывающие ее исходные данные не изменяются во времени. Статичность области означает неизменность описывающих ее исходных данных. При этом производные данные (выводимые из исходных) могут и появляться заново, и изменяться (не изменяя, однако, исходных данных).

Если исходные данные, описывающие предметную область, изменяются за время решения задачи, то предметную область называют динамической. В архитектуру динамической экспертной системы по сравнению со статической вводятся два компонента:

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

Последняя подсистема осуществляет связи с внешним миром через систему датчиков и контроллеров. Кроме того, традиционные компоненты статической экспертной системы (база знаний и механизм логического вывода) претерпевают существенные изменения, чтобы отразить временную логику происходящих в реальном мире событий.

5.1.3. Структура экспертных систем

Типичная статическая ЭС состоит из следующих основных компонентов (рис. 5.2):

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

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

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

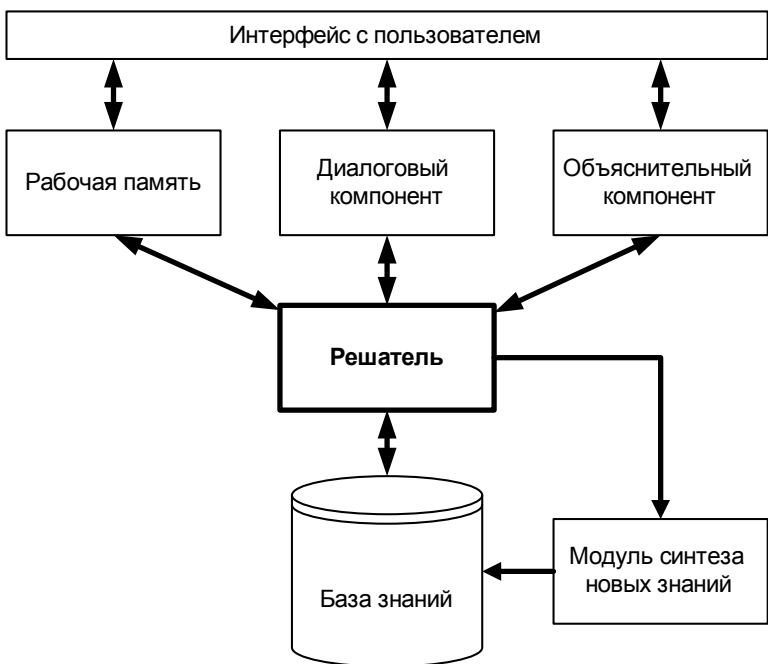


Рис. 5.2. Структура статической ЭС

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

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

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

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

В разработке ЭС участвуют представители следующих специальностей:

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

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

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

Инженер по знаниям помогает эксперту выявить и структурировать знания, необходимые для работы ЭС. Он же осуществляет выбор того ИС, которое наиболее подходит для данной проблемной области, и определяет способ представления знаний в этом ИС; выделяет и программирует (традиционными средствами) стандартные функции (типичные для данной проблемной области), которые будут использоваться в правилах, вводимых экспертом.

Программист разрабатывает ИС (если ИС разрабатывается заново), содержащее в пределе все основные компоненты

ЭС, и осуществляет его сопряжение с той средой, в которой оно будет использовано.

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

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

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

В режиме консультации общение с ЭС осуществляет конечный пользователь, которого интересует результат и/или способ его получения. Необходимо отметить, что в зависимости от назначения ЭС пользователь может не быть специалистом в данной проблемной области (в этом случае он обращается к ЭС за результатом, не умея получить его сам) или быть специалистом (в этом случае пользователь может сам получить резуль-

тат, но он обращается к ЭС с целью либо ускорить процесс получения результата, либо возложить на ЭС рутинную работу). В режиме консультации данные о задаче пользователя после обработки их диалоговым компонентом поступают в рабочую память. Решатель на основе входных данных из рабочей памяти, общих данных о проблемной области и правил из БЗ формирует решение задачи. ЭС при решении задачи не только исполняет предписанную последовательность операций, но и предварительно формирует ее. Если реакция системы не понятна пользователю, то он может потребовать объяснения:

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

Структуру, приведенную на рисунке 5.3, называют *структурой динамической ЭС*. ЭС данного типа используются в тех приложениях, где можно не учитывать изменения окружающего мира, происходящие за время решения задачи. Первые ЭС, получившие практическое использование, были статическими.

На рисунке 5.3 показано, что в архитектуру динамической ЭС по сравнению со статической ЭС вводятся два компонента: подсистема моделирования внешнего мира и подсистема связи с внешним окружением. Последняя подсистема осуществляет связи с внешним миром через систему датчиков и контроллеров. Кроме того, традиционные компоненты статической ЭС (база знаний и машина вывода) претерпевают существенные изменения, чтобы отразить временную логику происходящих в реальном мире событий.



Рис. 5.3. Архитектура динамической ЭС

Подчеркнем, что структура ЭС, представленная на рисунках 5.2 и 5.3, отражает только компоненты (функции), и многое остается "за кадром". На рисунке 5.3 приведена обобщенная

структура современного ИС для создания динамических ЭС, содержащая кроме основных компонентов те возможности, которые позволяют создавать интегрированные приложения в соответствии с современной технологией программирования.

5.2. МЕТОДОЛОГИЯ РАЗРАБОТКИ ЭКСПЕРТНЫХ СИСТЕМ

Рассмотрим методику формализации экспертных знаний на примере создания экспертных диагностических систем (ЭДС).

Целью создания ЭДС является определение состояния объекта диагностирования (ОД) и имеющихся в нем неисправностей.

Состояниями ОД могут быть: исправно, неисправно, работоспособно. Неисправностями, например, радиоэлектронных ОД являются обрыв связи, замыкание проводников, неправильное функционирование элементов и т. д.

Число неисправностей может быть достаточно велико (несколько тысяч). В ОД может быть одновременно несколько неисправностей. В этом случае говорят, что неисправности кратные.

Введем следующие определения. Разные неисправности ОД проявляются во внешней среде информационными параметрами. Совокупность значений информационных параметров определяет "информационный образ" (ИО) неисправности ОД. ИО может быть полным, т. е. содержать всю необходимую информацию для постановки диагноза, или соответственно неполным. В случае неполного ИО постановка диагноза носит вероятностный характер.

Основой для построения эффективных динамических ЭС являются знания эксперта для постановки диагноза, записанные в виде информационных образов, а также система представления знаний, встраиваемая в информационные системы обеспечения функционирования и контроля ОД, интегрируемые с соответствующей технической аппаратурой.

Для описания своих знаний эксперт с помощью инженера по знаниям должен выполнить следующее.

Выделить множество всех неисправностей ОД, которые должна различать ЭДС.

Выделить множество информативных (существенных) параметров, значения которых позволяют различить каждую неисправность ОД и поставить диагноз с некоторой вероятностью.

Для выбранных параметров следует выделить информативные значения или информативные диапазоны значений, которые могут быть как количественными, так и качественными. Например, точные количественные значения могут быть записаны: задержка 25 нс, задержка 30 нс и т. д. Количественный диапазон значений может быть записан следующим образом: задержка 25–40, 40–50, 50 нс и выше. Качественный диапазон значений может быть записан: индикаторная лампа светится ярко, светится слабо, не светится.

Для более удобного дальнейшего использования качественный диапазон значений может быть закодирован, например, следующим образом:

светится ярко $P1 = +++$ (или $P1 = 3$),

светится слабо $P1 = ++$ (или $P1 = 2$),

не светится $P1 = +$ (или $P1 = 1$).

Процедура получения информации по каждому из параметров определяется индивидуально в каждой конкретной системе диагностирования. Эта процедура может заключаться в автоматическом измерении параметров в ЭДС, в ручном измерении параметра с помощью приборов, качественном определении параметра, например светится слабо, и т. д.

Процедура создания полных или неполных ИО каждой неисправности в алфавите значений информационных параметров может быть определена следующим образом. Составляются диагностические правила, определяющие вероятный диагноз на основе различных сочетаний диапазонов значений выбранных параметров ОД. Правила могут быть записаны в различной форме. Ниже приведена форма записи правил в виде таблицы 5.1.

Для записи правил с учетом изменений по времени следует ввести еще один параметр P0 — время (еще один столбец в таблице). В этом случае диагноз может ставиться на основе нескольких строк таблицы, а в графе Примечания могут быть указаны использованные тесты. Диагностическая таблица в этом случае представлена в таблице 5.2.

Таблица 5.1. Диагностические правила

Номер	P1	P2	P3	Диагноз	Вероятность диагноза	Примечания
1		+++		Неисправен блок А1	0,95	
2	12–15	+		Неисправен блок А2	0,80	

Таблица 5.2. Динамические диагностические правила

Номер	P0	P1	P2	P3	Диагноз	Вероятность диагноза	Примечания
1	12:00	+	+	+			Тест Т1
2	12:15	++	++	+	Неисправен блок А3	0,90	

Для записи последовательности проведения тестовых процедур и задания ограничений (если они есть) на их проведение может быть предложен аналогичный механизм.

Механизм записи последовательности проведения тестовых процедур в виде правил реализуется, например, следующим образом:

ЕСЛИ: P2 = 1;

ТО: тест = Т1, Т3, Т7,

где Т1, Т3, Т7 — тестовые процедуры, подаваемые на ОД при активизации (срабатывании) соответствующей продукции.

В современных динамических ЭС применяются различные стратегии поиска решения и постановки диагноза, которые позволяют определить необходимые последовательности тестовых процедур. Однако приоритет в ЭС отдается прежде всего знаниям и опыту, а лишь затем логическому выводу.

Данная методика будет применена при создании экспертной системы управления технологическим процессом.

5.3. ЭТАПЫ РАЗРАБОТКИ ЭКСПЕРТНЫХ СИСТЕМ

Разработка ЭС имеет существенные отличия от разработки обычного программного продукта. Опыт создания ЭС показал, что использование при их разработке методологии, приня-

той в традиционном программировании, либо чрезмерно затягивает процесс создания ЭС, либо вообще приводит к отрицательному результату.

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

1) существуют эксперты в данной области, которые решают задачу значительно лучше, чем начинающие специалисты;

2) эксперты сходятся в оценке предлагаемого решения, иначе нельзя будет оценить качество разработанной ЭС;

3) эксперты способны вербализовать (выразить на естественном языке) и объяснить используемые ими методы, в противном случае трудно рассчитывать на то, что знания экспертов будут "извлечены" и вложены в ЭС;

4) решение задачи требует только рассуждений, а не действий;

5) задача не должна быть слишком трудной (т. е. ее решение должно занимать у эксперта несколько часов или дней, а не недель);

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

7) решение задачи не должно в значительной степени использовать "здоровый смысл" (т. е. широкий спектр общих сведений о мире и о способе его функционирования, которые знает

и умеет использовать любой нормальный человек), так как подобные знания пока не удастся (в достаточном количестве) вложить в системы искусственного интеллекта.

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

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

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

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

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

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

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

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

Прототип должен продемонстрировать пригодность методов инженерии знаний для данного приложения. В случае успеха эксперт с помощью инженера по знаниям расширяет знания прототипа о проблемной области. При неудаче может потребоваться разработка нового прототипа или разработчики могут прийти к выводу о непригодности методов ЭС для данного приложения. По мере увеличения знаний прототип может достигнуть такого состояния, когда он успешно решает все задачи

данного приложения. Преобразование прототипа ЭС в конечный продукт обычно приводит к перепрограммированию ЭС на языках низкого уровня, обеспечивающих как увеличение быстродействия ЭС, так и уменьшение требуемой памяти. Трудоемкость и время создания ЭС в значительной степени зависят от типа используемого инструментария.

В ходе работ по созданию ЭС сложилась определенная технология их разработки, включающая шесть следующих этапов (рис. 5.4): идентификацию, концептуализацию, формализацию, выполнение, тестирование, опытную эксплуатацию. На этапе *идентификации* определяются задачи, которые подлежат решению, выявляются цели разработки, определяются эксперты и типы пользователей.

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

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

На этапе *выполнения* осуществляется наполнение экспертом базы знаний. В связи с тем, что основой ЭС являются знания, данный этап является наиболее важным и трудоемким этапом разработки ЭС. Процесс приобретения знаний разделяют на извлечение знаний из эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном ЭС. Процесс приобретения знаний

осуществляется инженером по знаниям на основе анализа деятельности эксперта по решению реальных задач.

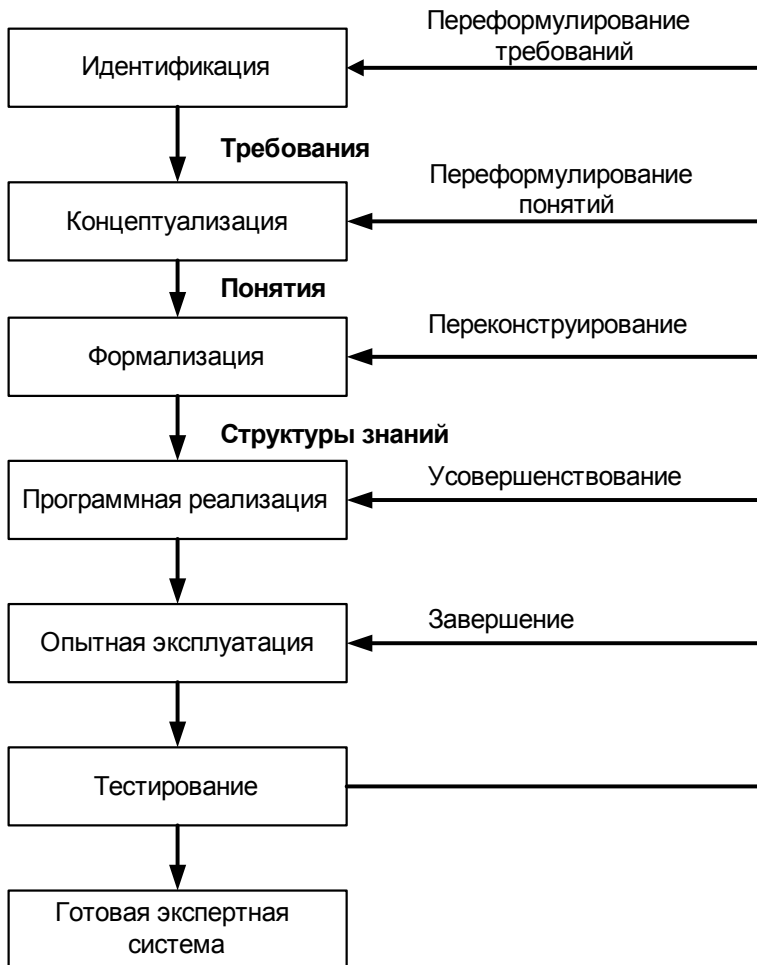


Рис. 5.4. Технология разработки ЭС

5.4. ВЗАИМОДЕЙСТВИЕ ИНЖЕНЕРА ПО ЗНАНИЯМ (КОГНИТОЛОГА) С ЭКСПЕРТОМ

5.4.1. Представление знаний в экспертных системах

Первый и основной вопрос, который надо решить при представлении знаний, — это вопрос определения состава знаний, т. е. определение того, "ЧТО ПРЕДСТАВЛЯТЬ" в экспертной системе. Второй вопрос касается того, "КАК ПРЕДСТАВЛЯТЬ" знания. Необходимо отметить, что эти две проблемы не являются независимыми. Действительно, выбранный способ представления может оказаться непригодным в принципе либо неэффективным для выражения некоторых знаний.

Вопрос "КАК ПРЕДСТАВЛЯТЬ" можно разделить на две в значительной степени независимые задачи: как организовать (структурировать) знания и как представить знания в выбранном формализме.

Стремление выделить организацию знаний в самостоятельную задачу вызвано, в частности, тем, что эта задача возникает для любого языка представления и способа решения этой задачи являются одинаковыми (либо сходными) вне зависимости от используемого формализма.

Итак, в круг вопросов, решаемых при представлении знаний, будем включать следующие:

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

Состав знаний ЭС определяется следующими факторами:

- проблемной средой;
- архитектурой экспертной системы;

- потребностями и целями пользователей;
- языком общения.

В соответствии с общей схемой статической экспертной системы (см. рис. 5.2) для ее функционирования требуются следующие знания:

- знания о процессе решения задачи (т. е. управляющие знания), используемые интерпретатором (решателем);
- знания о языке общения и способах организации диалога, используемые лингвистическим процессором (диалоговым компонентом);
- знания о способах представления и модификации знаний, используемые компонентом приобретения знаний;
- поддерживающие структурные и управляющие знания, используемые объяснительным компонентом.

Для динамической ЭС, кроме того, необходимы следующие знания:

- 1) знания о методах взаимодействия с внешним окружением;
- 2) знания о модели внешнего мира.

Зависимость состава знаний от требований пользователя проявляется в следующем:

- какие задачи (из общего набора задач) и с какими данными хочет решать пользователь;
- каковы предпочтительные способы и методы решения;
- при каких ограничениях на количество результатов и способы их получения должна быть решена задача;
- каковы требования к языку общения и организации диалога;
- какова степень общности (конкретности) знаний о проблемной области, доступная пользователю;
- каковы цели пользователей.

Состав знаний о языке общения зависит как от языка общения, так и от требуемого уровня понимания.

С учетом архитектуры экспертной системы знания целесообразно делить на *интерпретируемые* и *неинтерпретируемые*. К первому типу относятся те знания, которые способен интерпретировать решатель (интерпретатор). Все остальные знания относятся ко второму типу. Решатель не знает их структуры и содержания. Если эти знания используются каким-либо компонентом системы, то он не "осознает" этих знаний. Неинтерпретируемые знания подразделяются на *вспомогательные* знания, хранящие информацию о лексике и грамматике языка общения, информацию о структуре диалога, и *поддерживающие* знания. Вспомогательные знания обрабатываются естественно-языковой компонентой, но ход этой обработки решатель не осознает, так как этот этап обработки входных сообщений является вспомогательным для проведения экспертизы. Поддерживающие знания используются при создании системы и выполнении объяснений. Поддерживающие знания выполняют роль описаний (обоснований) как интерпретируемых знаний, так и действий системы. Поддерживающие знания подразделяются на *технологические* и *семантические*. Технологические поддерживающие знания содержат сведения о времени создания описываемых ими знаний, об авторе знаний и т. п. Семантические поддерживающие знания содержат смысловое описание этих знаний. Они содержат информацию о причинах ввода знаний, о назначении знаний, описывают способ использования знаний и получаемый эффект. Поддерживающие знания имеют описательный характер.

Интерпретируемые знания можно разделить на *предметные знания, управляющие знания и знания о представлении*. Знания о представлении содержат информацию о том, каким образом (в каких структурах) в системе представлены интерпретируемые знания.

Предметные знания содержат данные о предметной области и способах преобразования этих данных при решении поставленных задач. Отметим, что по отношению к предметным знаниям знания о представлении и знания об управлении являются *метазнаниями*. В предметных знаниях можно выделить описатели и собственно предметные знания. Описатели содержат определенную информацию о предметных знаниях, такую как коэффициент определенности правил и данных, меры важности и сложности. Собственно предметные знания разбиваются на *факты и исполняемые утверждения*. Факты определяют возможные значения сущностей и характеристик предметной области. Исполняемые утверждения содержат информацию о том, как можно изменять описание предметной области в ходе решения задач. Говоря другими словами, исполняемые *утверждения* — это знания, задающие процедуры обработки. Однако мы избегаем использовать термин "процедурные знания", так как хотим подчеркнуть, что эти знания могут быть заданы не только в процедурной, но и в декларативной форме.

Управляющие знания можно разделить на *фокусирующие и решающие*. Фокусирующие знания описывают, какие знания следует использовать в той или иной ситуации. Обычно фокусирующие знания содержат сведения о наиболее перспективных объектах или правилах, которые целесообразно использовать при проверке соответствующих гипотез. В первом случае

внимание фокусируется на элементах рабочей памяти, во втором — на правилах базы знаний. Решающие знания содержат информацию, используемую для выбора способа интерпретации знаний, подходящего к текущей ситуации. Эти знания применяются для выбора стратегий или эвристик, наиболее эффективных для решения данной задачи.

Качественные и количественные показатели экспертной системы могут быть значительно улучшены за счет использования *метазнаний*, т. е. знаний о знаниях. Метазнания не представляют некоторую единую сущность, они могут применяться для достижения различных целей. Перечислим возможные значения метазнаний:

- 1) метазнания в виде стратегических метаправил используются для выбора релевантных правил;
- 2) метазнания используются для обоснования целесообразности применения правил из области экспертизы;
- 3) метаправила используются для обнаружения синтаксических и семантических ошибок в предметных правилах;
- 4) метаправила позволяют системе адаптироваться к окружению путем перестройки предметных правил и функций;
- 5) метаправила позволяют явно указать возможности и ограничения системы, т. е. определить, что система знает, а чего не знает.

Вопросы организации знаний необходимо рассматривать в любом представлении, и их решение в значительной степени не зависит от выбранного способа (модели) представления. Выделим следующие аспекты проблемы организации знаний:

- организация знаний по уровням представления и по уровням детальности;

- организация знаний в рабочей памяти;
- организация знаний в базе знаний.

5.4.2. Уровни представления и уровни детальности

Для того чтобы экспертная система могла управлять процессом поиска решения, была способна приобретать новые знания и объяснять свои действия, она должна уметь не только использовать свои знания, но и обладать способностью понимать и исследовать их. То есть экспертная система должна иметь знания о том, как представлены ее знания о проблемной среде. Если знания о проблемной среде назвать знаниями нулевого уровня представления, то первый уровень представления содержит метазнания, т. е. знания о том, как представлены во внутреннем мире системы знания нулевого уровня. Первый уровень содержит знания о том, какие средства используются для представления знаний нулевого уровня. Знания первого уровня играют существенную роль при управлении процессом решения, при приобретении и объяснении действий системы. В связи с тем, что знания первого уровня не содержат ссылок на знания нулевого уровня, знания первого уровня независимы от проблемной среды.

Число уровней представления может быть больше двух. Второй уровень представления содержит сведения о знаниях первого уровня, т. е. знания о представлении базовых понятий первого уровня. Разделение знаний по уровням представления обеспечивает расширение области применимости системы.

Выделение уровней детальности позволяет рассматривать знания с различной степенью подробности. Количество уровней детальности во многом определяется спецификой ре-

шаемых задач, объемом знаний и способом их представления. Как правило, выделяется не менее трех уровней детальности, отражающих соответственно общую, логическую и физическую организацию знаний. Введение нескольких уровней детальности обеспечивает дополнительную степень гибкости системы, так как позволяет производить изменения на одном уровне, не затрагивая другие. Изменения на одном уровне детальности могут приводить к дополнительным изменениям на этом же уровне, что оказывается необходимым для обеспечения согласованности структур данных и программ. Однако наличие различных уровней препятствует распространению изменений с одного уровня на другие.

5.4.3. Организация знаний в рабочей системе

Рабочая память (РП) экспертных систем предназначена для хранения данных. Данные в рабочей памяти могут быть однородны или разделяются на уровни по типам данных. В последнем случае на каждом уровне рабочей памяти хранятся данные соответствующего типа. Выделение уровней усложняет структуру экспертной системы, но делает систему более эффективной. Например, можно выделить уровень планов, уровень агенды (упорядоченного списка правил, готовых к выполнению) и уровень данных предметной области (уровень решений).

В современных экспертных системах данные в рабочей памяти рассматриваются как изолированные или как связанные. В первом случае рабочая память состоит из множества простых элементов, а во втором — из одного или нескольких (при нескольких уровнях в РП) сложных элементов (например,

объектов). При этом сложный элемент соответствует множеству простых, объединенных в единую сущность. Теоретически оба подхода обеспечивают полноту, но использование изолированных элементов в сложных предметных областях приводит к потере эффективности.

Данные в РП в простейшем случае являются *константами* и/или *переменными*. При этом переменные могут трактоваться как характеристики некоторого объекта, а константы — как значения соответствующих характеристик. Если в РП требуется анализировать одновременно несколько различных объектов, описывающих текущую проблемную ситуацию, то необходимо указывать, к каким объектам относятся рассматриваемые характеристики. Одним из способов решения этой задачи является явное указание того, к какому объекту относится характеристика.

Если РП состоит из сложных элементов, то связь между отдельными объектами указывается явно, например заданием семантических отношений. При этом каждый объект может иметь свою внутреннюю структуру. Необходимо отметить, что для ускорения поиска и сопоставления данные в РП могут быть связаны не только логически, но и ассоциативно.

5.4.4. Организация знаний в базе данных

Показателем интеллектуальности системы с точки зрения представления знаний считается способность системы использовать в нужный момент необходимые (*релевантные*) знания. Системы, не имеющие средств для определения релевантных знаний, неизбежно сталкиваются с проблемой "комбинаторного взрыва". Можно утверждать, что эта проблема является одной

из основных причин, ограничивающих сферу применения экспертных систем. В проблеме доступа к знаниям можно выделить три аспекта: *связность знаний и данных, механизм доступа к знаниям и способ сопоставления.*

Связность (агрегация) знаний является основным способом, обеспечивающим ускорение поиска релевантных знаний. Большинство специалистов пришли к убеждению, что знания следует организовывать вокруг наиболее важных объектов (сущностей) предметной области. Все знания, характеризующие некоторую сущность, связываются и представляются в виде отдельного объекта. При подобной организации знаний, если системе потребовалась информация о некоторой сущности, она ищет объект, описывающий эту сущность, а затем уже внутри объекта отыскивает информацию о данной сущности. В объектах целесообразно выделять два типа связей между элементами: *внешние* и *внутренние*. Внутренние связи объединяют элементы в единый объект и предназначены для выражения структуры объекта. Внешние связи отражают взаимозависимости, существующие между объектами в области экспертизы. Многие исследователи классифицируют внешние связи на *логические* и *ассоциативные*. Логические связи выражают семантические отношения между элементами знаний. Ассоциативные связи предназначены для обеспечения взаимосвязей, способствующих ускорению процесса поиска релевантных знаний.

Основной проблемой при работе с большой базой знаний является проблема поиска знаний, релевантных решаемой задаче. В связи с тем, что в обрабатываемых данных может не содержаться явных указаний на значения, требуемые для их

обработки, необходим более общий механизм доступа, чем метод прямого доступа (метод явных ссылок). Задача этого механизма состоит в том, чтобы по некоторому описанию сущности, имеющемуся в рабочей памяти, найти в базе знаний объекты, удовлетворяющие этому описанию. Очевидно, что упорядочение и структурирование знаний могут значительно ускорить процесс поиска.

Нахождение желаемых объектов в общем случае уместно рассматривать как двухэтапный процесс. На первом этапе, соответствующем процессу выбора по ассоциативным связкам, совершается предварительный выбор в базе знаний потенциальных кандидатов на роль желаемых объектов. На втором этапе путем выполнения операции сопоставления потенциальных кандидатов с описаниями кандидатов осуществляется окончательный выбор искомым объектов. При организации подобного механизма доступа возникают определенные трудности: как выбрать критерий пригодности кандидата; как организовать работу в конфликтных ситуациях и т. п.

Операция сопоставления может использоваться не только как средство выбора нужного объекта из множества кандидатов; она может быть использована для классификации, подтверждения, декомпозиции и коррекции. Для идентификации неизвестного объекта он может быть сопоставлен с некоторыми известными образцами. Это позволит классифицировать неизвестный объект как такой известный образец, при сопоставлении с которым были получены лучшие результаты. При поиске сопоставление используется для подтверждения некоторых кандидатов из множества возможных. Если осуществлять сопоставление некоторого известного объекта с неизвестным описанием, то в

случае успешного сопоставления будет осуществлена частичная декомпозиция описания.

Операции сопоставления весьма разнообразны. Обычно выделяют следующие их формы: *синтаксическое*, *параметрическое*, *семантическое* и *принуждаемое сопоставления*. В случае *синтаксического сопоставления* соотносят формы (образцы), а не содержание объектов. Успешным является сопоставление, в результате которого образцы оказываются идентичными. Обычно считается, что переменная одного образца может быть идентична любой константе (или выражению) другого образца. Иногда на переменные, входящие в образец, накладывают требования, определяющие тип констант, с которыми они могут сопоставляться. Результат синтаксического сопоставления является бинарным: образцы сопоставляются или не сопоставляются. В *параметрическом сопоставлении* вводится параметр, определяющий степень сопоставления. В случае *семантического сопоставления* соотносятся не образцы объектов, а их функции. В случае *принуждаемого сопоставления* один сопоставляемый образец рассматривается с точки зрения другого. В отличие от других типов сопоставления, здесь всегда может быть получен положительный результат. Вопрос состоит в силе принуждения. Принуждение могут выполнять специальные процедуры, связываемые с объектами. Если эти процедуры не в состоянии осуществить сопоставление, то система сообщает, что успех может быть достигнут только в том случае, если определенные части рассматриваемых сущностей можно считать сопоставляющимися.

5.5. МЕТОДЫ ПОИСКА РЕШЕНИЙ В ЭКСПЕРТНЫХ СИСТЕМАХ

Методы решения задач, основанные на сведении их к поиску, зависят от особенностей предметной области, в которой решается задача, и от требований, предъявляемых пользователем к решению. Особенности предметной области с точки зрения методов решения можно характеризовать следующими параметрами:

- размер, определяющий объем пространства, в котором предстоит искать решение;
- изменяемость области, характеризует степень изменяемости области во времени и пространстве (здесь будем выделять статические и динамические области);
- полнота модели, описывающей область, характеризует адекватность модели, используемой для описания данной области. Обычно если модель не полна, то для описания области используют несколько моделей, дополняющих друг друга за счет отражения различных свойств предметной области;
- определенность данных о решаемой задаче, характеризует степень точности (ошибочности) и полноты (неполноты) данных. Точность (ошибочность) является показателем того, что предметная область с точки зрения решаемых задач описана точными или неточными данными; под полнотой (неполнотой) данных понимается достаточность (недостаточность) входных данных для однозначного решения задачи.

Требования пользователя к результату задачи, решаемой с помощью поиска, можно характеризовать количеством реше-

ний и свойствами результата и/или способом его получения. Параметр "количество решений" может принимать следующие основные значения: одно решение, несколько решений, все решения. Параметр "свойства" задает ограничения, которым должен удовлетворять полученный результат или способ его получения. Так, например, для системы, выдающей рекомендации по лечению больных, пользователь может указать требование не использовать некоторое лекарство (в связи с его отсутствием или в связи с тем, что оно противопоказано данному пациенту). Параметр "свойства" может определять и такие особенности, как время решения ("не более чем", "диапазон времени" и т. п.), объем памяти, используемой для получения результата, указание об обязательности (невозможности) использования каких-либо знаний (данных) и т. п.

Итак, сложность задачи, определяемая вышеприведенным набором параметров, варьируется от простых задач малой размерности с неизменяемыми определенными данными и отсутствием ограничений на результат и способ его получения до сложных задач большой размерности с изменяемыми, ошибочными и неполными данными и произвольными ограничениями на результат и способ его получения. Из общих соображений ясно, что каким-либо одним методом нельзя решить все задачи. Обычно одни методы превосходят другие только по некоторым из перечисленных параметров.

Рассмотренные ниже методы могут работать в статических и динамических проблемных средах. Для того чтобы они работали в условиях динамики, необходимо учитывать время жизни значений переменных, источник данных для переменных, а также обеспечивать возможность хранения истории значений

переменных, моделирования внешнего окружения и оперирования временными категориями в правилах.

Существующие методы решения задач, используемые в экспертных системах, можно классифицировать следующим образом:

- методы поиска в одном пространстве — методы, предназначенные для использования в следующих условиях: области небольшой размерности, полнота модели, точные и полные данные;
- методы поиска в иерархических пространствах — методы, предназначенные для работы в областях большой размерности;
- методы поиска при неточных и неполных данных;
- методы поиска, использующие несколько моделей, предназначенные для работы с областями, для адекватного описания которых одной модели недостаточно.

Предполагается, что перечисленные методы при необходимости должны объединяться для того, чтобы позволить решать задачи, сложность которых возрастает одновременно по нескольким параметрам.

5.6. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ ЭКСПЕРТНЫХ СИСТЕМ

Инструментальные средства для проектирования и разработки экспертных систем можно классифицировать по следующим основным параметрам:

- уровень используемого языка;
- парадигмы программирования и механизмы реализации;

- способ представления знаний;
- механизмы вывода и моделирования;
- средства приобретения знаний.

Уровень используемого языка

Мощность и универсальность языка программирования определяет трудоемкость разработки ЭС.

1. Традиционные (в том числе объектно-ориентированные) языки программирования типа С, С++ (как правило, они используются не для создания ЭС, а для создания инструментальных средств).

2. Специальные языки программирования (например, язык LISP, ориентированный на обработку списков; язык логического программирования PROLOG; язык рекурсивных функций РЕФАЛ и т. д.). Их недостатком является слабая приспособленность к объединению с программами, написанными на языках традиционного программирования.

3. Инструментальные средства, содержащие многие, но не все компоненты ЭС (например, система OPS 5, которая поддерживает производственный подход к представлению знаний; языки KRL и FRL, используемые для разработки ЭС с фреймовым представлением знаний). Такое программное обеспечение предназначено для разработчиков, владеющих технологиями программирования и умеющих интегрировать разнородные компоненты в программный комплекс.

4. Оболочки ЭС общего назначения, содержащие все программные компоненты, но не имеющие знаний о конкретных предметных средах. Средства этого типа и последующего не требуют от разработчика приложения знания программирования. Примерами являются ЭКО, Leonardo, Nexpert Object, Кар-

ра, EXSYS, GURU, ART, KEE и др. В последнее время все реже употребляется термин "оболочка", его заменяют более широким термином "среда разработки". Если хотят подчеркнуть, что средство используется не только на стадии разработки приложения, но и на стадиях использования и сопровождения, то употребляют термин "полная среда" (complete environment). Для поддержания всего цикла создания и сопровождения программ используются интегрированные инструментальные системы, например KEATS, VITAL. Основными компонентами системы KEATS являются: ACQUIST — средства фрагментирования текстовых источников знаний, позволяющие разбивать текст или протокол беседы с экспертом на множество взаимосвязанных, аннотированных фрагментов и создавать понятия (концепты); FLIK — язык представления знаний средствами фреймовой модели; GIS — графический интерфейс, используемый для создания гипертекстов и концептуальных моделей, а также для проектирования фреймовых систем; ERI — интерпретатор правил, реализующий процедуры прямого и обратного вывода; TRI — инструмент визуализации логического вывода, демонстрирующий последовательность выполнения правил; Tabes — интерфейс манипулирования таблицами, используемыми для хранения знаний в БЗ; CS — язык описания и распространения ограничений; TMS — система поддержания истинности.

При использовании инструментария данного типа могут возникнуть следующие трудности:

а) управляющие стратегии, заложенные в механизм вывода, могут не соответствовать методам решения, которые использует эксперт, взаимодействующий с данной системой, что

может привести к неэффективным, а возможно, и неправильным решениям;

б) способ представления знаний, используемый в инструментари, мало подходит для описания знаний конкретной предметной области.

Большая часть этих трудностей разрешена в проблемно/предметно-ориентированных средствах разработки ИИС.

5. Проблемно/предметно-ориентированные оболочки и среды (не требуют знания программирования):

- проблемно-ориентированные средства — предназначены для решения задач определенного класса (задачи поиска, управления, планирования, прогнозирования и др.) и содержат соответствующие этому классу альтернативные функциональные модули;
- предметно-ориентированные средства — включают знания о типах предметных областей, что сокращает время разработки БЗ.

При использовании оболочек и сред разработчик приложения полностью освобождается от программирования, его основные трудозатраты связаны с формированием базы знаний.

Парадигмы программирования и механизмы реализации

Способы реализации механизма исполняемых утверждений часто называют парадигмами программирования. К основным парадигмам относят следующие:

- процедурное программирование;
- программирование, ориентированное на данные;
- программирование, ориентированное на правила;
- объектно-ориентированное программирование.

Парадигма процедурного программирования является самой распространенной среди существующих языков программирования (например, С и Паскаль). В процедурной парадигме активная роль отводится процедурам, а не данным; причем любая процедура активизируется вызовом. Подобные способы задания поведения удобны для описаний последовательности действий одного процесса или нескольких взаимосвязанных процессов.

При использовании программирования, ориентированного на данные, активная роль принадлежит данным, а не процедурам. Здесь со структурами активных данных связывают некоторые действия (процедуры), которые активизируются тогда, когда осуществляется обращение к этим данным.

В парадигме, ориентированной на правила, поведение определяется множеством правил вида "условие — действие". Условие задает образ данных, при возникновении которого действие правила может быть выполнено. Правила в данной парадигме играют такую же роль, как и операторы в процедурной парадигме. Однако если в процедурной парадигме поведение задается последовательностью операторов, не зависящей от значений обрабатываемых данных, то в парадигме, ориентированной на правила, поведение не задается заранее предписанной последовательностью правил, а формируется на основе значений данных, которые в текущий момент обрабатываются программой. Подход, ориентированный на правила, удобен для описания поведения, гибко и разнообразно реагирующего на большое многообразие состояний данных.

Парадигма объектного программирования, в отличие от процедурной парадигмы, не разделяет программу на процедуры

и данные. Здесь программа организуется вокруг сущностей, называемых объектами, которые включают локальные процедуры (методы) и локальные данные (переменные). Поведение (функционирование) в этой парадигме организуется путем пересылки сообщений между объектами. Объект, получив сообщение, осуществляет его локальную интерпретацию, основываясь на локальных процедурах и данных. Такой подход позволяет описывать сложные системы наиболее естественным образом. Он особенно удобен для интегрированных ЭС.

Способ представления знаний

Наличие многих способов представления знаний вызвано стремлением представить различные типы проблемных сред с наибольшей эффективностью. Обычно способ представления знаний в ЭС характеризуется моделью представления знаний. Типичными моделями представления знаний являются правила (продукции), фреймы (или объекты), семантические сети, логические формулы. Инструментальные средства, имеющие в своем составе более одной модели представления знаний, называют гибридными. Большинство современных средств, как правило, использует объектно-ориентированную парадигму, объединенную с парадигмой, ориентированной на правила.

Механизмы вывода и моделирования

В статических ЭС единственным активным агентом, изменяющим информацию, является механизм вывода экспертной системы. В динамических ЭС изменение данных происходит не только вследствие функционирования механизма исполняемых утверждений, но также в связи с изменениями окружения задачи, которые моделируются специальной подсистемой или по-

ступают извне. Механизмы вывода в различных средах могут отличаться способами реализации следующих процедур.

1. Структура процесса получения решения:

- построение дерева вывода на основе обучающей выборки (индуктивные методы приобретения знаний) и выбор маршрута на дереве вывода в режиме решения задачи;
- компиляция сети вывода из специфических правил в режиме приобретения знаний и поиск решения на сети вывода в режиме решения задачи;
- генерация сети вывода и поиск решения в режиме решения задачи, при этом генерация сети вывода осуществляется в ходе выполнения операции сопоставления, определяющей пары "правило — совокупность данных", на которых условия этого правила удовлетворяются;
- в режиме решения задач ЭС осуществляет выработку правдоподобных предположений (при отсутствии достаточной информации для решения); выполнение рассуждений по обоснованию (опровержению) предположений; генерацию альтернативных сетей вывода; поиск решения в сетях вывода.

2. Поиск (выбор) решения:

- направление поиска — от данных к цели, от целей к данным, двунаправленный поиск;
- порядок перебора вершин в сети вывода — "поиск в ширину", при котором сначала обрабатываются все вершины, непосредственно связанные с текущей обрабатываемой вершиной G; "поиск в глубину", когда сначала раскрывается одна наиболее значимая верши-

на — G1, связанная с текущей G, затем вершина G1 делается текущей и для нее раскрывается одна наиболее значимая вершина G2 и т. д.

3. Процесс генерации предположений и сети вывода:

- режим — генерация в режиме приобретения знаний, генерация в режиме решения задачи;
- полнота генерируемой сети вывода — операция сопоставления применяется ко всем правилам и ко всем типам указанных в правилах сущностей в каждом цикле работы механизма вывода (обеспечивается полнота генерируемой сети); используются различные средства для сокращения количества правил и/или сущностей, участвующих в операции сопоставления, например применяется алгоритм сопоставления или используются знания более общего характера (метазнания).

Механизм вывода для динамических проблемных сред дополнительно содержит: планировщик, управляющий деятельностью ЭС в соответствии с приоритетами; средства, гарантирующие получение лучшего решения в условиях ограниченности ресурсов; систему поддержания истинности значений переменных, изменяющихся во времени.

В динамических инструментальных средствах могут быть реализованы следующие варианты подсистемы моделирования:

- система моделирования отсутствует;
- существует система моделирования общего назначения, являющаяся частью инструментальной среды;

- существует специализированная система моделирования, являющаяся внешней по отношению к программному обеспечению, на котором реализуется ЭС.

Средства приобретения знаний

В инструментальных системах они характеризуются следующими признаками.

1. Уровень языка приобретения знаний:

- формальный язык;
- ограниченный естественный язык;
- язык пиктограмм и изображений;
- ЕЯ и язык изображений.

2. Тип приобретаемых знаний:

- данные в виде таблиц, содержащих значения входных и выходных атрибутов, по которым индуктивными методами строится дерево вывода;
- специализированные правила;
- общие и специализированные правила.

3. Тип приобретаемых данных:

- атрибуты и значения;
- объекты;
- классы структурированных объектов и их экземпляры, получающие значения атрибутов путем наследования.

5.7. ТРУДНОСТИ РАЗРАБОТКИ ЭКСПЕРТНЫХ СИСТЕМ

Разработка ЭС связана с определенными трудностями, которые необходимо хорошо знать, так же как и способы их преодоления. Рассмотрим подробнее эти проблемы.

1. Проблема извлечения знаний экспертов. Ни один специалист никогда просто так не раскроет секреты своего профессионального мастерства, свои сокровенные знания в профессиональной области. Он должен быть заинтересован материально или морально, причем хорошо заинтересован. Никто не хочет рубить сук, на котором сидит. Часто такой специалист опасается, что, раскрыв все свои секреты, он будет не нужен компании. Вместо него будет работать экспертная система. Избежать эту проблему поможет выбор высококвалифицированного эксперта, заинтересованного в сотрудничестве.

2. Проблема формализации знаний экспертов. Эксперты-специалисты в определенной области, как правило, не в состоянии формализовать свои знания. Часто они принимают правильные решения на интуитивном уровне и не могут аргументированно объяснить, почему принято то или иное решение. Иногда эксперты не могут прийти к взаимопониманию (фраза "встретились два геолога, у них было три мнения" — не шутка, а реальная жизнь). В таких ситуациях поможет выбор эксперта, умеющего ясно формулировать свои мысли и легко объяснять другим свои идеи.

3. Проблема нехватки времени у эксперта. Выбранный для разработки эксперт не может найти достаточно времени для выполнения проекта. Он слишком занят. Он всем нужен. У него есть проблемы. Чтобы избежать этой ситуации, необходимо получить от эксперта, прежде чем начнется проект, согласие тратить на проект время в определенном фиксированном объеме.

4. Правила, формализованные экспертом, не дают необходимой точности. Проблемы можно избежать, если решать вместе с экспертом реальные задачи. Не надо придумывать

"игрушечных" ситуаций или задач. В условиях задач нужно использовать реальные данные, такие как лабораторные данные, отчеты, дневники и другую информацию, взятую из практических задач. Постарайтесь говорить с экспертом на одном языке, используя единую терминологию. Эксперт, как правило, легче понимает правила, записанные на языке, близком к естественному, а не на языке типа LISP или PROLOG.

5. Недостаток ресурсов. В качестве ресурсов выступают персонал (инженеры знаний, разработчики инструментальных средств, эксперты) и средства построения ЭС (средства разработки и средства поддержки). Недостаток благожелательных и грамотных администраторов порождает скептицизм и нетерпение у руководителей. Повышенное внимание в прессе и преувеличения вызвали нереалистические ожидания, которые приводят к разочарованию в отношении экспертных систем. ЭС могут давать не самые лучшие решения на границе их применимости, при работе с противоречивыми знаниями и в рассуждениях на основе здравого смысла. Могут потребоваться значительные усилия, чтобы добиться небольшого увеличения качества работы ЭС. Экспертные системы требуют много времени на разработку. Так, создание системы PUFF для интерпретации функциональных тестов легких потребовало 5 человеко-лет, на разработку системы PROSPECTOR для разведки рудных месторождений ушло 30 человеко-лет, система XCON для расчета конфигурации компьютерных систем на основе VAX 11/780 потребовала 8 человеко-лет. ЭС последних лет разрабатываются более быстрыми темпами за счет развития технологий ЭС, но проблемы остались. Увеличение персонала не сокращает время разработки, потому что процесс создания ЭС — это процесс

со множеством обратных связей. Все это необходимо учитывать при планировании создания ЭС.

6. Неадекватность инструментальных средств решаемой задаче. Часто определенные типы знаний (например, временные или пространственные) не могут быть легко представлены на одном ЯПЗ, так же как и разные схемы представления (например, фреймы и продукции) не могут быть достаточно эффективно реализованы на одном ЯПЗ. Некоторые задачи могут быть непригодными для решения по технологии ЭС (например, отдельные задачи анализа сцен). Необходим тщательный анализ решаемых задач, чтобы определить пригодность предлагаемых инструментальных средств и сделать правильный выбор.

5.8. ПЕРСПЕКТИВЫ РАЗВИТИЯ ЭКСПЕРТНЫХ СИСТЕМ

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

Значимость инструментальных средств реального времени определяется не столько их бурным коммерческим успехом (хотя и это достойно тщательного анализа), но в первую очередь тем, что только с помощью подобных средств создаются стратегически значимые приложения в таких областях, как управление непрерывными производственными процессами в химии, фармакологии, производстве цемента, продуктов питания и т. п., аэрокосмические исследования, транспортировка и переработка нефти и газа, управление атомными и тепловыми

электростанциями, финансовые операции, связь и многие другие.

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

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

1) представлять изменяющиеся во времени данные, поступающие от внешних источников, обеспечивать хранение и анализ изменяющихся данных;

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

3) обеспечивать механизм рассуждения при ограниченных ресурсах (время, память). Реализация этого механизма предъявляет требования к высокой скорости работы системы, способности одновременно решать несколько задач (т. е. операционные системы UNIX, VMS, Windows);

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

5) моделировать "окружающий мир", рассматриваемый в данном приложении, обеспечивать создание различных его состояний;

6) протоколировать свои действия и действия персонала, обеспечивать восстановление после сбоя;

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

8) обеспечивать настройку системы на решаемые задачи (проблемная/предметная ориентированность);

9) обеспечивать создание и поддержку пользовательских интерфейсов для различных категорий пользователей;

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

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

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

Основное предназначение программных продуктов фирмы Gensym (США) — помочь предприятиям сохранять и использо-

вать знания и опыт их наиболее талантливых и квалифицированных сотрудников в интеллектуальных системах реального времени, повышающих качество продукции, надежность и безопасность производства и снижающих производственные издержки. О том, как фирме Gensym удается справиться с этой задачей, говорит хотя бы то, что сегодня ей принадлежат 50% мирового рынка экспертных систем, используемых в системах управления.

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

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

- мониторинг в реальном масштабе времени;
- системы управления верхнего уровня;
- системы обнаружения неисправностей;
- диагностика;
- составление расписаний;
- планирование;
- оптимизация;
- системы — советчики оператора;
- системы проектирования.

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

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

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

ровался по большей части бессистемно, под влиянием резких колебаний в экономике.

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

- интеллектуальное управление производством — G2, G2 Diagnostic Assistant (GDA), NeurOn-Line (NOL), Statistical Process Control (SPC), BatchDesign_Kit;
- оперативное планирование — G2, G2 Scheduling Toolkit (GST), Dynamic Scheduling Packadge (DSP);
- разработка и моделирование производственных процессов — G2, ReThink, BatchDesign_Kit;
- управление операциями и корпоративными сетями — G2, Fault Expert.

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

используют G2. На базе G2 написано более 500 действующих приложений.

Успех G2 объясняется прежде всего тем, что G2 — динамическая система в полном смысле этого слова. G2 — это объектно-ориентированная интегрированная среда для разработки и сопровождения приложений реального времени, использующих базы знаний. G2 функционирует на большинстве существующих платформ (табл. 5.3). База знаний G2 сохраняется в обычном ASCII-файле, который однозначно интерпретируется на любой из поддерживаемых платформ.

Перенос приложения не требует его перекомпиляции и заключается в простом переписывании файлов. Функциональные возможности и внешний вид приложения не претерпевают при этом никаких изменений.

Таблица 5.3. Платформы, на которых функционирует G2

Фирма-производитель	Вычислительная система	Операционная среда
Digital Equipment	VAX 3xxx,4xxx,6xxx,7xxx, 8xxx,9xxx	VMS
	DECstation 3xxx, 6xxx	ULTRIX
	DEC Alpha APX	Open VMS, OSF/1, Windows
SUN Microsystems	SUN-4	Sun OS
	SPARC 1,2, 10, LX,	Sun OS/Solaris 1, Solaris
	Classic	2.x
Hewlett Packard	HP9000/4xx, 7xx, 8xx	HP-UX

Продолжение табл. 5.3

Фирма-производитель	Вычислительная система	Операционная среда
IBM	RISC 6000	AIX
Data General	AViiON	DG/UX
Silicon Graphics	IRIS, INDIGO	IRIX
ПЭВМ	Intel Pentium	Windows XX
Motorola	Motorola 88000	UNIX
NEC	EWS 4800	EWS-UX/V

G2 объединяет в себе как универсальные технологии построения современных информационных систем (стандарты открытых систем, архитектура клиент/сервер, объектно-ориентированное программирование, использование ОС, обеспечивающих параллельное выполнение в реальном времени многих независимых процессов), так и специализированные методы (рассуждения, основанные на правилах, рассуждения, основанные на динамических моделях, или имитационное моделирование, процедурные рассуждения, активная объектная графика, структурированный естественный язык для представления базы знаний), а также интегрирует технологии систем, основанных на знаниях с технологией традиционного программирования (с пакетами программ, с СУБД, с контроллерами и концентраторами данных и т. д.) (рис. 5.5).

Поскольку многие организации увеличивают свои инвестиции в информационные технологии и системы автоматизации, то людям становится все труднее интерпретировать и реагировать на большое количество данных, которые динамически изменяются во времени выполнения, не говоря уже о предупреждении критических ситуаций. Интеллектуальные справоч-

ные системы предохраняют людей от информационной перегрузки, перерабатывая оперативные данные в полезную информацию так, чтобы нужные действия могли быть приняты в нужное время.

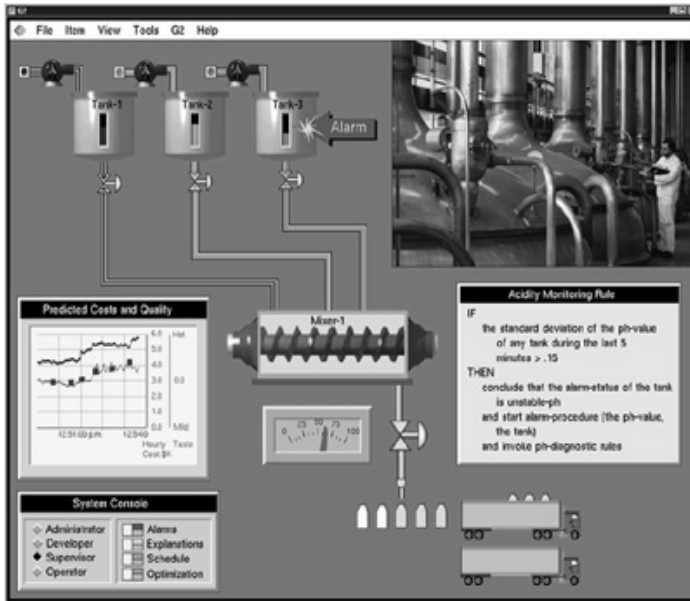


Рис. 5.5. Экспертная система, построенная в оболочке G2

G2 — мощная среда для разработки и развертывания интеллектуальных динамических систем управления. Прикладные программы, написанные в среде G2, могут существенно повысить эффективность выполнения операций благодаря следующим факторам:

- непрерывному контролю над потенциальными проблемами прежде, чем они проявят неблагоприятное воздействие;

- принятию комплексных оперативных решений на основе информации, полученной посредством рассуждений и анализа данных, содержащихся в интеллектуальной модели процесса;
- диагностированию основных случаев возникновения проблем, критичных ко времени выполнения и выработке последовательности правильных действий;
- поддержанию оптимальных рабочих условий;
- координированию действий и информации в выполнении сложных оперативных процессов.

G2 увеличивает вашу производительность так, что вы получаете готовые прикладные программы быстрее и они обладают более мощными функциональными возможностями, чем при использовании других программных средств. Результатом является более быстрое получение интеллектуальных прикладных программ. Это происходит из-за:

- более быстрого комплексного использования новой системы;
- сокращения времени разработки;
- уменьшения проектного риска;
- понижения расходов на сопровождение программы;
- более простого интегрирования с другим программным обеспечением и системами;
- быстрого создания прототипа системы, чтобы продолжить проектирование совместно с конечным пользователем.

Объекты, используемые в прикладных программах, написанных в среде G2, описывают свойства и действия динамиче-

ских операций. Эти объекты легко определяются, а затем многократно используются. Синтаксические ошибки в описаниях объектов, правил и процедур исправляются мгновенно. Результат — быстрая разработка прикладных систем реального времени.

Объекты в G2 — это мощный и интуитивный способ представления материальных и абстрактных сущностей в прикладных программах. Мощность объектно-ориентированного подхода к разработке программ позволяет быстро и легко:

- встроить модули и объекты из других прикладных программ;
- графически определить объекты, их свойства и действия;
- создать новые образцы объектов, имитируя существующие объекты.

Объекты или класс объектов, определенные однажды, могут использоваться многократно. Любой объект или группа объектов могут иметь несколько экземпляров. Каждый экземпляр наследует все свойства и поведение первоначального объекта(-ов). Объекты, правила и процедуры можно группировать в библиотеки, которые будут общими для всех прикладных программ. Для моделирования широкого разнообразия действий типа производственных процессов, сетевых топологий, информационных маршрутизаций или логических потоков объекты можно объединять графически на экране дисплея.

В G2 можно эффективно создавать и применять общие знания, создавая универсальные правила, процедуры, формулы и зависимости, которые являются применимыми для полных классов объектов. В результате сокращается время на разработку и увеличивается эффективность приложений.

В G2 для представления знаний используется структурный естественный язык, что позволяет облегчить чтение, редактирование и поддержку баз знаний. Это облегчает использование и редактирование приложений пользователем-непрограммистом. Для создания и редактирования баз знаний используется редактор баз знаний.

Для представления знаний эксперта о проблемной области используются правила. Правила могут быть как общими, т. е. относящимися ко всему классу, так и специфическими, относящимися к конкретным экземплярам класса. Правила возбуждаются автоматически в следующих случаях:

- появились новые данные (вывод от фактов к цели);
- требуется найти данные (вывод от цели к фактам) для автоматического вызова других правил, процедур или формул;
- требуется определить значения переменных;
- каждые n секунд для оценивания правила в указанном интервале времени.

Подобно правилам процедуры в G2 выполняются в реальном времени. Процедуры, правила и модели выполняются одновременно согласно их приоритетам. Процедуры могут использоваться для эффективного представления поведения объектов. Для процедур можно определить состояния ожидания и выполнять обработку информации параллельно. В результате это позволяет формировать мощные прикладные системы реального времени проще и быстрее, чем с помощью традиционных инструментальных средств программирования.

Работа в реальном времени, операционные решения и реакции зачастую должны быть выполнены мгновенно. Среда

G2 предназначена для эффективной работы в реальном времени. Прикладные программы в G2 могут одновременно выполнять рассуждения относительно многократно выполняемых действий в реальном режиме времени, перерабатывая тысячи правил, выполняя процедуры и модели согласно их приоритетам. Для хранения хронологий данных и событий и для рассуждения относительно поведения через какое-то время используются переменные типа время. G2 может сохранять мгновенные снимки состояния прикладной программы во время ее выполнения. Поэтому в случае сбоя питания G2 может выполнить "перезапуск из памяти" самого последнего сохраненного состояния.

G2-графика может моделировать знание, представляя объекты, связи и зависимости между объектами. G2 может рассуждать в терминах связи, следуя сети связанных объектов для определения причин и результатов. Графическая связность объектов G2 позволяет расширить прикладную программу, используя графическое объединение аналогов. Графика включает встроенные диаграммы (графики), таблицы и рисунки и т. д.

G2 также работает с утилитами графического интерфейса Windows. Эти утилиты используют все преимущества объектно-ориентированных возможностей G2.

G2 позволяют динамически моделировать системы и процессы, используя объекты, правила, процедуры и формулы. Во время разработки модели используются вместо объектов реального мира, что позволяет непрерывно проверять прикладные программы в течение их разработки. Модели могут использоваться на этапе эксплуатации как часть прикладной программы G2 для сравнения фактических и модельных знаний.

Модели можно также использовать для проведения анализа и ответа на вопросы "что, если", определения, например, лучших рабочих условий или лучших проектов. Модели можно также использовать для предсказания важных параметров действий в реальном времени, например качества изделия или затрат.

Поскольку G2 имеет способность объединять опытное знание с аналитическими методами, большие индустриальные компании используют прикладные программы, основанные на G2 для минимизации потребления энергии, максимизации производительности, оптимизации ресурсов и управления процессами в окружающей среде.

Способность G2 обрабатывать события, критичные ко времени выполнения, и развитый интерфейс помогают операторам избежать аварийных ситуаций, которые могут приводить к ухудшению качества изделия (программы), задержкам графика, повреждению оборудования и другим существенным затратам.

G2 разработан для эффективного управления сложными, динамическими проблемами. Мощные технологии G2 позволяют:

- эффективно выполнить прикладные программы, критичные ко времени выполнения;
- использовать архитектуру клиент/сервер;
- использовать эффективный интерфейс с базами данных в реальном режиме времени;
- улучшить принятие решений, эффективно применяя знания экспертов совместно с рассуждениями, основанными на правилах моделях;

- облегчить редактирование прикладных программ благодаря использованию структурированного естественного языка и возможностей удаленного доступа;
- быстро поставить сетевые прикладные программы;
- многократно использовать знания, хранящиеся в библиотеках.

Шлюз G2 позволяет прикладным программам связываться с помощью интерфейса с широким разнообразием систем передачи и обработки данных в реальном режиме времени, а также с базами данных. Шлюз может осуществлять буферизацию данных, обработку протокола, подтверждение связи, восстановление данных и многие другие функции. Межсетевой проект позволяет одновременно получать сообщения от многих источников данных. Шлюз действует как сервер данных для G2. Шлюз может выполняться на той же платформе, что и G2.

Все это позволяет с помощью данной оболочки создавать практически любые большие приложения значительно быстрее, чем с использованием традиционных методов программирования, и снизить трудозатраты на сопровождение готовых приложений и их перенос на другие платформы.

ВИДЕОРОЛИКИ ПО ГЛАВЕ

<p>Экспертная система Витулкаса в программе Radar 10 http://www.youtube.com/watch?v=_cP5gm_Pnmk</p>	
<p>Джордж Витулкас о создании Экспертной системы Витулкаса http://www.youtube.com/watch?v=VaHuH758dbU</p>	
<p>БИТ:УКС (CRM) Экспертная система для проведения опросов http://www.youtube.com/watch?v=RjQTcb6VkCs</p>	
<p>Среда разработки IBM Eclipse http://www.youtube.com/watch?v=oan72NAyT8M</p>	

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите и охарактеризуйте основные компоненты статических экспертных систем.
2. Какого профиля специалисты привлекаются для разработки экспертных систем? Каковы их функции?
3. Чем отличаются динамические экспертные системы от статических?

4. Охарактеризуйте экспертную систему по следующим параметрам: типу приложения, стадии существования, масштабу, типу проблемной среды, типу решаемой задачи.

5. Расскажите об основных характеристиках инструментальных средств, предназначенных для разработки интеллектуальных информационных систем.

6. Опишите основные технологические этапы разработки экспертных систем: идентификацию, концептуализацию, формализацию, выполнение, тестирование, опытную эксплуатацию.

7. Какие типы задач решаются с применением ЭС? Приведите примеры.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Суркова, Н. Е.* Методы проектирования информационных систем / Н. Е. Суркова, А. В. Остроух. — М.: РосНОУ, 2004. — 144 с.
2. *Остроух, А. В.* Основы построения систем искусственного интеллекта для промышленных и строительных предприятий: монография. — М.: ООО "Техполиграфцентр", 2008. — 280 с.
3. *Остроух, А. В.* Автоматизация транспортировки продукции / А. В. Остроух, Н. Г. Куфтинова. — Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2011. — 146 с.
4. *Остроух, А. В.* Рефакторинг баз данных. Автоматизация технологических процессов рефакторинга баз данных промышленных предприятий / А. В. Остроух, Д. А. Пшеничный, О. Б. Рогова. — Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2013. — 133 с.
5. *Остроух, А. В.* Автоматизация управления производством. Повышение эффективности автоматизированных аналитических систем предприятий автомобильной промышленности / А. В. Остроух, Э. А. Чернов, Д. Т. Нгуен. — Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2013. — 285 с.
6. *Остроух, А. В.* Системы планирования перевозок. Программно-технологические решения по разработке системы планирования заданий для заказных пассажирских перевозок / А. В. Остроух, А. Б. Львова, А. Р. Исмаилов. — Saarbrücken, Germany: LAP LAMBERT Academic Publishing, 2013. — 121 с.

7. *Остроух, А. В.* Интеллектуальные системы в науке и производстве / А. В. Остроух, А. Б. Николаев. — Saarbrücken, Germany : Palmarium Academic Publishing, 2012. — 312 с.
8. *Остроух, А. В.* Мультимедиа технологии / А. В. Остроух, А. М. Васьковский, А. Б. Николаев. — Saarbrücken, Germany : Palmarium Academic Publishing, 2012. — 228 с.
9. *Остроух, А. В.* Ввод и обработка цифровой информации : учебник для нач. проф. образования. — М. : Издат. центр "Академия", 2012. — 288 с.
10. *Остроух, А. В.* Оперативный контроль транспортно-экспедиционной деятельности. Процессный подход к агрегированию системы показателей деятельности транспортно-экспедиционного предприятия / А. В. Остроух, А. М. Ивахненко, Н. А. Крупенский. — Saarbrücken, Germany : Palmarium Academic Publishing, 2013. — 221 с.
11. *Исмоилов, М. И.* Подготовка и переподготовка персонала предприятий промышленного и транспортного комплексов с применением мобильных технологий : монография / М. И. Исмоилов, А. Б. Николаев, А. В. Остроух. — Saint-Louis, MO, USA : Publishing House Science and Innovation Center, 2013. — 166 с.
12. *Николаев, А. Б.* Информационные технологии в менеджменте и транспортной логистике : учеб. пособие / А. Б. Николаев, А. В. Остроух. — Saint-Louis, MO, USA : Publishing House Science and Innovation Center, 2013. — 254 с.
13. *Остроух, А. В.* Системы искусственного интеллекта в промышленности, робототехнике и транспортном комплексе : монография / А. В. Остроух. — Красноярск : Научно-инновационный центр, 2013. — 326 с.

14. *Остроух, А. В.* Основы информационных технологий : учебник для сред. проф. образования. — М. : Издат. центр "Академия", 2014. — 208 с.
15. *Суркова, Н. Е.* Методология структурного проектирования информационных систем : монография / Н. Е. Суркова, А. В. Остроух. — Красноярск : Научно-инновационный центр, 2014. — 190 с.
16. *Покровский, А. К.* Системный анализ и компьютерные технологии при управлении транспортным обслуживанием предприятий : монография / А. К. Покровский, А. В. Остроух, А. М. Ивахненко. — М. : Известия, 2014. — 261 с.
17. *Демидов, Д. Г.* Программные и аппаратные средства систем мультимедиа : учеб. пособие / Д. Г. Демидов, А. М. Васьковский, А. Б. Николаев, А. В. Остроух, П. И. Лукашук, В. А. Виноградов. — М. : МГУП им. Ивана Федорова, 2014. — Ч. 1 : Аппаратные средства. — 78 с.
18. *Демидов, Д. Г.* Программные и аппаратные средства систем мультимедиа : учеб. пособие / Д. Г. Демидов, А. М. Васьковский, А. Б. Николаев, А. В. Остроух, П. И. Лукашук, В. А. Виноградов. — М. : МГУП им. Ивана Федорова, 2014. — Ч. 2 : Программные средства. — 70 с.
19. *Остроух, А. В.* Проектирование системы распределенных баз данных / А. В. Остроух, А. В. Помазанов. — Saarbrücken, Germany : Palmarium Academic Publishing, 2015. — 117 с.
20. *Остроух, А. В.* Автоматизация управления автотранспортными предприятиями. Новый подход на основе интеллектуальных мультиагентных систем / А. В. Остроух, А. В. Воробьева, Н. Е. Суркова. — Saarbrücken, Germany : LAP LAMBERT Academic Publishing, 2015. — 117 с.

21. *Суркова, Н. Е.* Профессиональные информационные системы и базы данных: методические указания к лабораторным работам / Н. Е. Суркова, А. В. Остроух, Т. И. Еремина. — Красноярск : Научно-инновационный центр, 2015. — 49 с.
22. *Остроух, А. В.* Об одном походе к построению автоматизированной навигационной системы мониторинга и диспетчерского контроля и учета работы транспорта нефтедобывающих предприятий / А. В. Остроух, Д. Б. Ефименко, В. Н. Жучков // Инженер-нефтяник. — 2011. — № 2. — С. 43–47.
23. *Остроух, А. В.* Разработка информационно-аналитической системы мониторинга технологических процессов предприятия автомобильной промышленности / А. В. Остроух, Ю. Тянь // В мире научных открытий. — Красноярск : Научно-инновационный центр, 2013. — № 8.2 (44). — С. 191–205.
24. *Остроух, А. В.* Интеграция компонентов системы мониторинга / А. В. Остроух, Ю. Тянь // Молодой ученый. — 2013. — № 10. — С. 182–185.
25. *Остроух, А. В.* Исследование начального периода моделирования на точность среднеинтегральной оценки имитационных моделей / А. В. Остроух, А. А. Солнцев, Н. В. Солдатов, К. А. Новицкий, П. С. Якунин // Вестник МАДИ. — 2010. — Вып. 2 (21). — С. 61–65.
26. *Остроух, А. В.* Математическая модель связей в системе диагностики электрооборудования автомобилей / А. В. Остроух, А. А. Солнцев, О. Ф. Калухов, Г. Г. Ягудаев // Вестник МАДИ. — 2010. — Вып. 2 (21). — С. 66–70.

27. Антонов, П. Д. User Is A Great Obstacle For Security Systems / П. Д. Антонов, А. В. Остроух // Молодой ученый. — 2011. — Т. 3. — № 4. — С. 62–63.
28. Структура системы электронного документооборота для управления научно-образовательной деятельностью высшего учебного заведения / М. Н. Краснянский, С. В. Карпушкин, А. Д. Обухов, Н. В. Молоткова [и др.] // Промышленные АСУ и контроллеры. — 2014. — № 8. — С. 23–31.
29. Krasnyanskiy, M. N. Information Technology for the Development of Automated Control System for University Projects / M. N. Krasnyanskiy, A. V. Ostroukh, S. V. Karpushkin, A. D. Obukhov // Математические методы в технике и технологиях — ММТТ-27 : сб. тр. XXVII Междунар. науч. конф. : в 12 т. Т. 4. Секции 10, 11 / под общ. ред. А. А. Большакова. — Тамбов : Тамбов. гос. техн. ун-т, 2014. — С. 42–45.
30. Краснянский, М. Н. Разработка комплексной системы управления научно-инновационной деятельностью: взаимодействие базы данных изделий и архива документации / М. Н. Краснянский, А. В. Остроух, С. В. Карпушкин, А. Д. Обухов // Приборы и системы. Управление, контроль, диагностика. — 2014. — № 12. — С. 19–23.
31. Сальный, А. Г. Исследование производительности файловых систем ядра Linux / А. Г. Сальный, А. В. Остроух // Автоматизация и управление в технических системах. — 2014. — № 4 (12). — С. 158–167.
32. Обухов, П. А. Исследование эффективности работы сетевых серверов в среде имитационного моделирования AnyLogic / П. А. Обухов, А. Б. Николаев, А. В. Остроух // Международный журнал экспериментального образования. — 2015. — № 3–3. — С. 338–342.

33. *Пастухов, Д. А.* Сравнительный анализ гипервизоров / Д. А. Пастухов, П. Ф. Юрчик, А. В. Остроух // Международный журнал экспериментального образования. — 2015. — № 3–3. — С. 346–350.
34. *Сальный, А. Г.* Исследование эффективности структур хранения данных ядра Linux / А. Г. Сальный, А. В. Остроух // Международный журнал экспериментального образования. — 2015. — № 3. — С. 366–367.
35. *Чувиков, Д. А.* Исследование 3D-форматов хранения данных в интеллектуальных системах виртуальной реальности / Д. А. Чувиков, В. П. Феоктистов, А. В. Остроух // Международный журнал экспериментального образования. — 2015. — № 3–3. — С. 416–420.
36. *Терентьев, Д. И.* Исследование дисковых массивов RAID по параметрам надежности и быстродействия / Д. И. Терентьев, А. Б. Николаев, А. В. Остроух // Международный журнал экспериментального образования. — 2015. — № 3–3. — С. 423–427.
37. *Круглов, А. М.* Актуализация сведений о данных информационной системы средствами активного словаря-справочника данных / А. М. Круглов, А. В. Будихин, Д. А. Буров, А. В. Остроух // Научный вестник МГТУ ГА. Сер. "Аэромеханика и прочность". — 2007. — № 119 (9). — С. 166–171.
38. *Николаев, А. В.* Принципы организации динамических интерфейсов доступа к данным с использованием словарей-справочников данных / А. В. Николаев, А. В. Будихин, Д. А. Буров, А. В. Остроух // Научный вестник МГТУ ГА. Сер. "Аэромеханика и прочность". — 2007. — № 119 (9). — С. 172–178.
39. *Остроух, А. В.* Использование словаря-справочника данных для реализации пользовательских средств обработки

- информации / А. В. Остроух, А. П. Баринов, А. В. Николаев, С. А. Будихин // Приборы и системы. Управление, контроль, диагностика. — 2008. — № 3. — С. 13–17.
40. *Пшеничный, Д. А.* Анализ параметров и сравнение СУБД для реализации информационного обеспечения промышленного предприятия / Д. А. Пшеничный, А. В. Будихин, А. В. Остроух // Промышленные АСУ и контроллеры. — 2010. — № 12. — С. 7–11.
41. *Помазанов, А. В.* Методика оптимизации баз данных / А. В. Помазанов, А. И. Белоусова, А. О. Васильева, А. В. Остроух // В мире научных открытий. Сер. "Проблемы науки и образования". — 2012. — № 12. — С. 49–54.
42. *Ostroukh, A. V.* Development of Automated Control System for University Research Projects / A. V. Ostroukh, M. N. Krasnyanskiy, S. V. Karpushkin, A. D. Obukhov // Middle East Journal of Scientific Research. — 2014. — Vol. 20 (12). — P. 1780–1784.
43. *Ostroukh, A.* Realtime Development and Testing of Distributed Data Processing System for Industrial Company / A. Ostroukh, A. Pomazanov // Middle East Journal of Scientific Research. — 2014. — Vol. 20 (12). — P. 2184–2193.
44. *Ostroukh, A. V.* Problems of organization and search the knowledge base in the CRM-systems / A. V. Ostroukh, A. I. Belousova, D. A. Pavlov, P. F. Yurchik // IOSR Journal of Engineering (IOSRJEN). — 2014. — Vol. 4, Iss. 02. V3. — P. 18–23.
45. *Krasnyanskiy, M. N.* Automated control system for university research projects / M. N. Krasnyanskiy, S. V. Karpushkin, A. D. Obukhov, A. V. Ostroukh // International Journal of Advanced Studies (iJAS). — 2014. — Vol. 4, Iss. 1. — P. 22–26.

46. *Krasnyanskiy, M. N.* Electronic Document Management Systems Structure for University Research and Education / M. N. Krasnyanskiy, A. V. Ostroukh, S. V. Karpushkin, A. D. Obukhov, N. V. Molotkova, I. V. Galygina // *Journal of Engineering and Applied Sciences.* — 2014. — Vol. 9, Iss. 5. — P. 182–189.
47. *Помазанов, А. В.* Создание и тестирование распределённой системы работы с удалёнными узлами / А. В. Помазанов, А. В. Остроух // *Автоматизация и современные технологии.* — 2014. — № 7. — С. 17–23.
48. *Помазанов, А. В.* Новый подход к разработке прототипа распределенной системы баз данных промышленного предприятия / А. В. Помазанов, А. В. Остроух // *Промышленные АСУ и контроллеры.* — 2014. — № 9. — С. 11–20.
49. *Суркова, Н. Е.* Методика обучения технологии баз данных для студентов непрофильных направлений подготовки в технических вузах / Н. Е. Суркова, А. В. Остроух // *Инженерная педагогика.* — М.: МАДИ, 2015. — Вып. 17. — Т. 3. — С. 146–156.
50. *Суркова, Н. Е.* Профессиональные информационные системы и базы данных: методические указания к лабораторным работам [Электронный ресурс] / Н. Е. Суркова, А. В. Остроух, Т. И. Еремина // *Хроники объединенного фонда электронных ресурсов наука и образование.* — 2015. — № 3 (70). — С. 10.
51. *Krasnyanskiy, M. N.* Model of Documents Management for Academic and Research Universities on Basis Set Theory / M. N. Krasnyanskiy, S. V. Karpushkin, A. D. Obukhov, A. V. Ostroukh // *American-Eurasian Journal of Agricultural & Environmental Sciences.* — 2015. — Vol. 15. — № 5. — P. 824–831.

52. *Краснянский, М. Н.* Математическая модель электронного документооборота на основе теории множеств / М. Н. Краснянский, С. В. Карпушкин, А. Д. Обухов, А. В. Остроух // Промышленные АСУ и контроллеры. — 2015. — № 5. — С. 52–58.
53. *Краснянский, М. Н.* Разработка информационной системы электронного документооборота управления фундаментальных и прикладных исследований / М. Н. Краснянский, А. Д. Обухов, С. В. Карпушкин, А. В. Остроух // Вестник Тамбовского государственного технического университета. — 2015. — Т. 21. — № 2. — С. 216–230.
54. *Ostroukh, A. V.* Simulated operation of the computer network / A. V. Ostroukh, K. N. Mezencev, M. N. Krasnyanskiy, J. Punam, N. E. Surkova // SGEM2015 Conference Proceedings, June 18–24. — 2015. — В. 2. — Vol. 1. — P. 299–306.
55. *Ostroukh, A. V.* Development of automated information systems for monitoring of intellectual activity results / A. V. Ostroukh, M. N. Krasnyanskiy, A. D. Obukhov, S. V. Karpov, D. L. Dedov // SGEM2015 Conference Proceedings, June 18–24. — 2015. — В. 2. — Vol. 1. — P. 101–108.
56. *Баринов, К. А.* Опыт разработки и использования ролевых игр для подготовки и переподготовки специалистов предприятий промышленности и транспортного комплекса / К. А. Баринов, Д. А. Буров, А. В. Бугаев, А. В. Остроух // Научный вестник МГТУ ГА. — 2009. — № 141. — С. 189–197.
57. *Николаев, А. Б.* Разработка информационно-образовательной системы для повышения квалификации персонала предприятий химической промышленности и транспортного комплекса [Электронный ресурс] / А. Б. Николаев, А. В. Остроух, М. Н. Краснянский // Электронное обу-

- чение и дистанционные образовательные технологии. — 2013. — № 1. — Режим доступа: eodot.esrae.ru/1-10.
58. *Лукашук, П. И.* Адаптивная методика прогнозирования пассажиропотоков в АСУ пассажирского автотранспортного предприятия / П. И. Лукашук, С. Бенгедаш, А. Г. Николаев, А. В. Остроух // Приборы и системы. Управление, контроль, диагностика. — 2006. — № 11. — С. 7–11.
59. *Кузнецов, И. А.* Особенности реализации автоматизированной информационно-аналитической системы центра планирования перевозок строительных грузов / И. А. Кузнецов, А. В. Остроух // Вестник МАДИ (ГТУ). — М. : МАДИ (ГТУ), 2008. — Вып. 1 (12). — С. 92–96.
60. *Куфтинова, Н. Г.* Процессно-ориентированный подход к автоматизации планирования и управления транспортировкой продукции предприятий промышленности / Н. Г. Куфтинова, А. В. Остроух // Вестник МАДИ — 2010. — Вып. 4 (23). — С. 62–66.
61. *Остроух, А. В.* Имитационное моделирование управления транспортными потоками в мегаполисе / А. В. Остроух, Н. Г. Куфтинова // Автотранспортное предприятие. — 2010. — № 12. — С. 41–42.
62. *Остроух, А. В.* Система автоматизации и контроля за продажами и производством автомобилей / А. В. Остроух, А. В. Алтунина // Автотранспортное предприятие. — 2011. — № 3. — С. 41–43.
63. *Польгун, М. Б.* Анализ моделей оперативного диспетчерского управления городским пассажирским транспортом / М. Б. Польгун, А. В. Воробьева, А. В. Остроух // Молодой ученый. — 2011. — № 4. — Т. 3. — С. 9–13.

64. *Ефименко, Д. Б.* Автоматизированная навигационная система диспетчерского контроля и учета работы транспорта нефтедобывающих предприятий / Д. Б. Ефименко, А. И. Губанов, А. Б. Николаев, А. В. Остроух // Молодой ученый. — 2011. — № 4. — Т. 3. — С. 18–21.
65. *Соломатин, А. В.* Об одном подходе к решению задачи регулирования дорожного движения: автоматизация системы управления светофорами / А. В. Остроух, А. В. Соломатин, О. О. Варламов // Автотранспортное предприятие. — 2011. — № 6. — С. 43–45.
66. *Ефименко, Д. Б.* Развитие навигационной системы диспетчерского управления грузовым транспортом (на примере нефтедобывающих предприятий) / Д. Б. Ефименко, А. В. Остроух, С. А. Филатов // Автотранспортное предприятие. — 2011. — № 11. — С. 32–34.
67. *Ефименко, Д. Б.* Концепция автоматизированной навигационной системы диспетчерского контроля и учета работы транспорта нефтедобывающих и нефтеперерабатывающих предприятий / Д. Б. Ефименко, А. В. Остроух, А. Б. Николаев, А. И. Губанов // Автоматизация, телемеханизация и связь в нефтяной промышленности. — 2011. — № 11. — С. 12–14.
68. *Остроух, А. В.* Научный подход к разработке автоматизированной навигационной системы диспетчерского контроля и учета работы транспорта нефтедобывающих и нефтеперерабатывающих предприятий / А. В. Остроух, А. Б. Николаев, Д. Б. Ефименко, С. В. Жанказиев // Автоматизированные системы управления и приборы автоматики. — Харьков : ХНУРЭ, 2011. — Вып. 157. — С. 48–58.
69. *Ostroukh, A. V.* Automation of Planning and Management of the Transportation of Production for Food Processing Industry

- Enterprises / A. V. Ostroukh, N. G. Kuftinova // Automatic Control and Computer Sciences. — 2012. — Vol. 46. — № 1. — P. 41–48.
70. *Остроух, А. В.* Автоматизация планирования и управления транспортировкой продукции предприятий пищевой промышленности / А. В. Остроух, Н. Г. Куфтинова // Автоматика и вычислительная техника. — Рига : ИЭ и ВТ Латвийского университета, 2012. — № 1. — С. 57–67.
71. *Остроух, А. В.* Розробка автоматизованої навігаційної системи диспетчерського контролю та обліку роботи транспорту нафтовидобувних підприємств / А. В. Остроух, А. Б. Николаев, Д. Б. Ефименко, С. В. Жанказиев // Нафтова і газова промисловість. — Киев : Укргазпром, 2012. — № 3. — С. 48–52.
72. *Данчук, К. А.* Автоматизированные информационные системы на автотранспортном предприятии / К. А. Данчук, А. Б. Львова, С. А. Порфирьева, А. В. Остроух, П. С. Якунин // В мире научных открытий. — 2012. — № 26 (27). — С. 34–38.
73. *Ефименко, Д. Б.* Использование программного обеспечения радионавигационных диспетчерских систем для транспортного обслуживания специальных объектов нефтедобывающих компаний / Д. Б. Ефименко, А. В. Остроух, А. Б. Николаев, А. Р. Исмаилов // Автотранспортное предприятие. — 2012. — № 2. — С. 42–44.
74. *Волков, Н. Н.* Автоматизация процессов начальной установки бинкулярных астровизиров в бортовых автономных информационно-вычислительных системах ориентации и навигации / Н. Н. Волков, А. В. Остроух, Е. А. Прохорова // Авиакосмическое приборостроение. — 2012. — № 10. — С. 8–15.

75. *Куфтинова, Н. Г.* Разработка информационно-логической модели транспортной сети мегаполиса / Н. Г. Куфтинова, А. В. Остроух // Бюллетень транспортной информации. — 2013. — № 1 (211). — С. 23–26.
76. *Польгун, М. Б.* Автоматизация процессов диспетчерского управления городским пассажирским транспортом / М. Б. Польгун, А. В. Остроух, А. Б. Николаев, Д. Б. Ефименко // Промышленные АСУ и контроллеры. — 2013. — № 5. — С. 10–16.
77. *Исмаилов, А. Р.* Разработка архитектуры подсистемы планирования организации заказных перевозок клиентских групп при проведении XXII зимних Олимпийских игр / А. Р. Исмаилов, А. В. Остроух, А. Б. Львова // Бюллетень транспортной информации. — 2013. — № 12 (222). — С. 3–10.
78. *Исмаилов, А. Р.* Автоматизированная система планирования заданий для заказных перевозок пассажиров при проведении Олимпийских игр / А. Р. Исмаилов, А. В. Остроух, А. Б. Львова, А. Б. Николаев, П. Ю. Збавитель // Транспорт Российской Федерации. — 2013. — № 6 (49). — С. 46–51.
79. *Исмаилов, А. Р.* Программно-технологические решения по разработке подсистем планирования заданий для заказных перевозок пассажиров при проведении Олимпийских игр / А. Р. Исмаилов, А. Б. Львова, А. В. Остроух // Приборы и системы. Управление, контроль, диагностика. — 2013. — № 11. — С. 74–85.
80. *Ismailov, A. R.* Organization and Management of Transport Provided for the Guests and Participants of the Olympic Games / A. R. Ismailov, A. B. Lvova, A. B. Nikolaev, A. V. Ostroukh // Middle East Journal of Scientific Research. — 2013. — Vol. 17 (8). — P. 1098–1104.

81. *Башмаков, И. А.* Процессная модель технологии транспортировки бетонных смесей автомобильным транспортом / И. А. Башмаков, А. В. Остроух // Автоматизация и управление в технических системах. — 2013. — № 4.1. — С. 75–81.
82. *Джха, П.* Автоматизированная система управления складом железобетонных изделий / П. Джха, А. В. Остроух // Автоматизация и управление в технических системах. — 2013. — № 4.1. — С. 51–56.
83. *Остроух, А. В.* Исследование информационных систем управления взаимоотношениями с поставщиками / А. В. Остроух, Б. Р. Синха // Автоматизация и управление в технических системах. — 2013. — № 4.1. — С. 56–62.
84. *Башмаков, И. А.* Обзор технологий транспортировки бетонных смесей автомобильным транспортом / И. А. Башмаков, М. Б. Польгун, П. Джха, А. В. Остроух // Автоматизация и управление в технических системах. — 2013. — № 4.2. — С. 178–189.
85. *Башмаков, И. А.* Оптимизация параметров процессов автотранспортного обслуживания потребителей бетонных смесей / И. А. Башмаков, М. Б. Польгун, А. В. Остроух // Автоматизация и управление в технических системах. — 2013. — № 4.2. — С. 189–198.
86. *Башмаков, И. А.* Минимизация производственных рисков при автотранспортном обслуживании потребителей бетонных смесей / И. А. Башмаков, А. В. Остроух // Автоматизация и управление в технических системах. — 2013. — № 4.2. — С. 83–90.
87. *Ostroukh, A. V.* Automation of processes supervisory control urban passenger transport / A. V. Ostroukh, M. B. Polgun // In-

- ternational Journal of Advanced Studies (iJAS). — 2013. — Vol. 3. — № 3. — P. 3–9.
88. *Ostroukh, A. V.* New approaches to development of automated supervisory systems of industrial enterprises transport / A. V. Ostroukh, M. B. Polgun // International Journal of Advanced Studies (iJAS). — 2013. — Vol. 3. — № 4. — P. 3–9.
89. *Остроух, А. В.* Проектирование программно-технологических решений подсистемы планирования заданий для заказных перевозок пассажиров при проведении XXII зимних Олимпийских игр / А. В. Остроух, А. Р. Исмаилов, А. Б. Львова // Автоматизация и управление в технических системах. — 2014. — № 1.1 (8). — С. 98–111.
90. *Остроух, А. В.* Применение графов дорожной сети в автоматизированной системе управления транспортировкой дорожно-строительных материалов / А. В. Остроух, М. Б. Польшун, В. В. Тихоцкий // Приборы и системы. Управление, контроль, диагностика. — 2014. — № 1. — С. 12–17.
91. *Ostroukh, A. V.* Process Model of the Technology of Concrete Mixtures Transportation by Road / A. V. Ostroukh, I. A. Bashmakov, N. E. Surkova // World Applied Sciences Journal (WASJ). — 2014. — Vol. 31. — № 4. — P. 500–507.
92. *Остроух, А. В.* Разработка логической модели данных подсистемы планирования заказных перевозок при проведении XXII зимних Олимпийских игр / А. В. Остроух, А. Р. Исмаилов, А. Б. Львова // Автоматизация и управление в технических системах. — 2014. — № 1.2 (9). — С. 111–122.
93. *Krupensky, N. A.* Process-oriented aggregation scorecards operational control forwarding activities / N. A. Krupensky, A. M. Ivakhnenko, A. V. Ostroukh // Автоматизация и управ-

- ление в технических системах. — 2014. — № 1.2 (9). — С. 123–142.
94. *Ostroukh, A. V.* Process-Functional Model of Transportation Mix Concrete / A. V. Ostroukh, I. A. Bashmakov, M. B. Polgun // *Journal of Transportation Technologies (JTT)*. — 2014. — Vol. 4, Iss. 2. — P. 157–163.
95. *Башмаков, И. А.* Процессная модель технологии транспортировки бетонных смесей автомобильным транспортом / И. А. Башмаков, А. В. Остроух // *Промышленные АСУ и контроллеры*. — 2014. — № 3. — С. 10–18.
96. *Ostroukh, A. V.* Development of Process-Oriented System For Operational Control of Freight Forwarding Activity / A. V. Ostroukh, A. M. Ivakhnenko, N. A. Krupensky // *Journal of Applied Sciences (JAS)*. — 2014. — Vol. 14. — № 20. — P. 2601–2607.
97. *Ostroukh, A. V.* Architecture of Automated Navigation System of Passenger Transportation at Winter Olympic Games / A. V. Ostroukh, A. R. Ismailov, A. B. Lvova, A. B. Nikolaev // *Trends in Applied Sciences Research (TASR)*. — 2014. — Vol. 9. — № 8. — P. 425–437.
98. *Куфтинова, Н. Г.* Разработка автоматизированной системы обследования пассажиропотоков / Н. Г. Куфтинова, А. В. Остроух // *Автоматизация и управление в технических системах*. — 2014. — № 3 (11). — С. 23–33.
99. *Ostroukh, A. V.* Comparative Study Of Routing Protocols In Vehicular Ad-Hoc Networks (Vanets) / A. V. Ostroukh, H. Elhadi // *International Journal of Advanced Studies (iJAS)*. — 2014. — Vol. 4. — № 2. — P. 9–14.
100. *Kuftinova, N. G.* Development of an automated system of survey passenger traffics / N. G. Kuftinova, A. V. Ostroukh // In-

- ternational Journal of Advanced Studies (iJAS). — 2014. — Vol. 4. — № 4. — P. 3–9.
101. *Kuftinova, N. G.* Automated Control System For Survey Passenger Traffics / N. G. Kuftinova, A. V. Ostroukh, A. V. Vorobieva // International Journal of Applied Engineering Research. — 2015. — Vol. 10. — № 7. — P. 16419–16427.
 102. *Гусеница, Д. О.* Применение облачного хранения данных в автоматизированной системе диспетчерского управления транспортом / Д. О. Гусеница, П. Ф. Юрчик, А. В. Остроух // Промышленные АСУ и контроллеры. — 2015. — № 5. — С. 59–66.
 103. *Gusenitsa, D. O.* Cloud Computing Application on Transport Dispatching Informational Support Systems / D. O. Gusenitsa, P. F. Yurchik, A. V. Ostroukh // International Journal of Advanced Studies (iJAS). — 2015. — Vol. 5. — № 1. — P. 22–27.
 104. *Ostroukh, A. V.* Automated Supervisory Control System Of Urban Passenger Transport / A. V. Ostroukh, N. E. Surkova, M. B. Polgun, A. V. Vorobieva // ARPN Journal of Engineering and Applied Sciences. — 2015. — Vol. 10. — № 10. — P. 4334–4340.
 105. *Остроух, А. В.* Математическая модель системы дистанционной диагностики неисправностей автомобилей / А. В. Остроух, Н. Е. Суркова, А. В. Воробьева, Х. С. Салих // В мире научных открытий. — 2015. — № 6. — С. 63–70.
 106. *Васюгова, С. А.* Исследование перспектив и проблем интеграции человека с компьютером: искусственный интеллект, робототехника, технологическая сингулярность и виртуальная реальность / С. А. Васюгова, А. В. Остроух, М. Н. Краснянский, А. Самаратунга // Перспективы нау-

- ки. — Тамбов : ТМБПринт, 2011. — № 4 (19). — С. 109–112.
107. Белоусова, А. И. Подход к формированию многоуровневой модели мультиагентной системы с использованием миваров / А. И. Белоусова, О. О. Варламов, М. Н. Краснянский, А. В. Остроух // Перспективы науки. — Тамбов : ТМБПринт, 2011. — № 5 (20). — С. 57–61.
108. Варламов, О. О. Анализ возможностей миварного подхода для систем искусственного интеллекта и современной робототехники / О. О. Варламов, А. В. Остроух, М. Н. Краснянский, Т. Л. Давыдова // Вестник ТГТУ. — 2011. — Т. 17. — № 3. — С. 687–694.
109. Ostroukh, A. V. Distributed System of Real Time Head Gesture Recognition in Development of Contactless Interfaces / A. V. Ostroukh, V. Nikonov, I. Ivanova, T. Morozova, V. Strakhov // Middle East Journal of Scientific Research. — 2014. — Vol. 20 (12). — P. 2177–2183.
110. Ostroukh, A. V. Development of Contactless Integrated Interface of Complex Production Lines / A. V. Ostroukh, V. Nikonov, I. Ivanova, T. Morozova, K. Sumkin, D. Akimov // Journal of Artificial Intelligence (JAI). — 2014. — Vol. 7. — № 1. — P. 1–12.
111. Morozova, T. Contactless integrated interface of production lines / T. Morozova, K. Sumkin, D. Akimov, A. V. Ostroukh // International Journal of Advanced Studies (iJAS). — 2014. — Vol. 4, Iss. 1. — P. 32–38.
112. Омар, М. Применение систем распознавания образов в различных предметных областях / М. Омар, Ф. Омар, М. И. Исмоилов, А. В. Остроух // Автоматизация и управление в технических системах. — 2014. — № 4 (12). — С. 32–47.

113. *Омар, М.* Анализ современного состояния развития интеллектуальных роботов / М. Омар, Ф. Омар, М. И. Исмоилов, А. В. Остроух // Автоматизация и управление в технических системах. — 2014. — № 4 (12). — С. 48–54.
114. *Ле, К. Х.* Генетические алгоритмы в задачах рациональной организации информационно-вычислительных процессов сетей / К. Х. Ле, Н. Е. Суркова, А. В. Остроух // Автоматизация и управление в технических системах. — 2014. — № 4 (12). — С. 82–99.
115. *Ivchenko, V.* The Remotely Reconfigurable Intelligence of the Space-Based Mobile Robot / V. Ivchenko, P. Krug, T. Morozova, A. V. Ostroukh, S. Pavelyev // Journal of Engineering and Applied Sciences. — 2014. — Vol. 9, Iss. 10. — P. 389–395.
116. *Ostroukh, A. V.* Integration of PDM and ERP systems within a unified information space of an enterprise / A. V. Ostroukh, D. O. Gusenitsa, V. B. Golubkova, P. F. Yurchik // IOSR Journal of Computer Engineering (IOSR-JCE). — 2014. — Vol. 16, Iss. 02. V6. — P. 31–33.

*Андрей Владимирович ОСТРОУХ,
Андрей Борисович НИКОЛАЕВ*

ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

МОНОГРАФИЯ

Издание второе, стереотипное

Зав. редакцией литературы по информационным
технологиям и системам связи *О. Е. Гайнутдинова*

ЛР № 065466 от 21.10.97

Гигиенический сертификат 78.01.10.953.П.1028
от 14.04.2016 г., выдан ЦГСЭН в СПб

Издательство «ЛАНЬ»

lan@lanbook.ru; www.lanbook.com

196105, Санкт-Петербург, пр. Ю. Гагарина, д. 1, лит. А.

Тел./факс: (812) 336-25-09, 412-92-72.

Бесплатный звонок по России: 8-800-700-40-71

Подписано в печать 09.07.21.

Бумага офсетная. Гарнитура Школьная. Формат 60×90^{1/16}.

Печать офсетная. Усл. п. л. 19,25. Тираж 500 экз.

Заказ № 892-21.

Отпечатано в полном соответствии

с качеством предоставленного оригинал-макета
в АО «Т8 Издательские Технологии».

109316, г. Москва, Волгоградский пр., д. 42, к. 5.