

**О. К. ЗАКУСИЛО, А. Ф. ВЕРЛАНЬ,
И. А. ЧМЫРЬ**

МОДЕЛИРОВАНИЕ ДИАЛОГОВЫХ ПРОЦЕССОВ

**Одесса
2014**

УДК 004.94:004.78

ББК

3-

3-

Закусило О. К., Верлань А. Ф., Чмырь И. А.

Моделирование диалоговых процессов / О. К. Закусило, А. Ф. Верлань, И. А. Чмырь. — Одесса : **Изд-во**, 2014. — ??? с.

ISBN

Книга посвящена моделированию диалоговых процессов. С целью исследования естественного диалогового процесса, или разговора, свойственного человеку, вводится онтологический базис, на основе которого строится модель диалоговой транзакции в контексте представления декларативных знаний, ассоциированных с транзакцией. Искусственный эротический диалоговый процесс моделируется рядом формальных моделей, различающихся сложностью и сферой применимости. Главной отличительной особенностью предложенных моделей является их ориентация на диалогический подход к моделированию и проектированию программных систем. Диалогический подход противопоставляется алгоритмическому и предполагает, что как сценарий диалога, так и его компоненты моделируются средствами баз данных. На основе модели predetermined procedural knowledge активного диалогового агента, воплощенной в диалоговую базу знаний, рассматриваются некоторые вопросы проектирования диалоговых систем, ориентированных на решение плохо формализуемых задач. Внимание сфокусировано на следующих классах таких систем: персонализированные системы обучения; электронные тексты со встроенным интеллектуальным компонентом; интеллектуальные системы автоматизированного проектирования; системы диалогового общения с базой данных. Развивается идея проблемно-независимого диалогового решателя задач и предлагается его логическая организация. Книга ориентирована на специалистов и аспирантов в области искусственного интеллекта, когнитивных наук и компьютерного обучения.

УДК 004.94:004.78

ББК

ISBN

© Закусило О. К., Верлань А. Ф.,
Чмырь И. А., 2014

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	5
ВВЕДЕНИЕ	6

Часть 1. ЕСТЕСТВЕННЫЙ И ФОРМАЛЬНЫЙ ДИАЛогоВЫЕ ПРОЦЕССЫ

1. ДИАЛОГОВЫЙ ПРОЦЕСС В СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА	16
1.1. Ранние вопросно-ответные системы	16
1.1.1. Вопросно-ответные системы поддержки естественно-языкового общения	16
1.1.2. Вопросно-ответные системы баз данных и анализа текстов	24
1.2. Диалог и дедукция в интеллектуальных системах обучения	28
1.2.1. Интеллектуальные компьютерные тьюторы	29
1.2.2. Обучающие среды с интеллектуальной поддержкой	36
1.2.3. Интеллектуальные компьютерные репетиторы	39
2. ЕСТЕСТВЕННЫЙ ДИАЛОГОВЫЙ ПРОЦЕСС	46
2.1. Когнитивные основы естественного диалогового процесса	46
2.2. Онтологический базис естественного диалогового процесса.	52
2.2.1. Диалоговый агент и его роли.	55
2.2.2. Межличностный и внутренний диалоги.	56
2.2.3. Транзакция диалогового протокола и ее компоненты.	57
2.2.4. Типы информационных посылок активного агента	60
2.2.5. Типы информационных посылок реактивного агента	65
2.3. Диалоговая транзакция в контексте представления знаний	67
2.3.1. Знания, ассоциированные с диалоговой транзакцией.	67
2.3.2. Тернарное представление декларативных знаний	68
2.3.3. Логическая структура диалоговой транзакции	71
2.3.4. Модели диалоговой транзакции в контексте тернарного представления декларативных знаний	78
3. ЭРОТЕТИЧЕСКИЙ ДИАЛОГОВЫЙ ПРОЦЕСС	83
3.1. Автоматная модель	84
3.2. Петри-модель.	89
3.2.1. Применимость Петри-модели	96
3.3. Сетевая модель.	101
3.3.1. Структура модели	101
3.3.2. Аналитическое представление модели.	107
3.4. Проблема моделирования «неискренности» реактивного агента.	114

4. ДАТАЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ СТРУКТУРЫ ВОПРОСА	118
4.1. Атрибутивные и неатрибутивные описатели объекта сцены	119
4.2. Классы атрибутов видеообъекта сцены	121
4.3. Проблема фрагментации атрибутивного описания объектов сцены.	131
4.4. Модель памяти вопросов	135

Часть 2. ДИАЛОГОВЫЙ РЕШАТЕЛЬ ЗАДАЧ

5. ЭРОТЕТИЧЕСКИЙ ДИАЛОГ КАК МЕТОД МОДЕЛИРОВАНИЯ ПРОЦЕДУРНЫХ ЗНАНИЙ ЭКСПЕРТА	142
5.1. Моделирующие возможности диалогового процесса.	143
5.2. Диалоговый и дедуктивный процессы в экспертных системах	148
5.3. Дедуктивные связи в модели диалогового метода доступа к памяти вопросов	158
5.3.1. Когнитивные основы дедуктивных связей в модели диалогового метода доступа	161
5.4. Формальное описание дедуктивных связей в модели диалогового метода доступа	166
5.4.1. Логический вывод в сети диалогового метода доступа	166
5.4.2. Вопросно-ответные отношения в контексте логического вывода	171
6. РЕШЕНИЕ ПЛОХО ФОРМАЛИЗУЕМЫХ ЗАДАЧ ДИАЛОГОВЫМИ МЕТОДАМИ	177
6.1. Персонализированные системы обучения.	178
6.1.1. Обучающие стимулы	178
6.1.2. Диагностические стимулы	182
6.1.3. Диалоговый сценарий, совмещающий диагностику и обучение	191
6.2. Электронный текст со встроенным интеллектуальным компонентом	195
6.2.1. Текст, расширенный ресурсами	195
6.2.2. Задачи встроенного интеллектуального компонента	197
6.2.3. Организация встроенного интеллектуального компонента	201
6.3. Интеллектуальные системы автоматизированного проектирования	209
6.3.1. Диалоговая имитация работы эксперта при моделировании явлений тепломассопереноса	210
6.3.2. Организация интеллектуальной системы автоматизированного проектирования общего назначения	214
6.4. Диалоговый метод формирования запросов к базе данных	219
7. ОРГАНИЗАЦИЯ ПРОБЛЕМНО-НЕЗАВИСИМОГО ДИАЛОГОВОГО РЕШАТЕЛЯ ЗАДАЧ	224
Список использованных источников	227
ПРИЛОЖЕНИЕ	243

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

<i>Que</i>	— вопрос
<i>Ans</i>	— ответ
<i>Subj</i>	— субъект вопроса
<i>Req</i>	— требование вопроса
K_{subj}	— порция декларативных знаний, ассоциированных с субъектом вопроса
K_{ans}	— порция декларативных знаний, ассоциированных с ответом
<i>DiAM</i>	— диалоговый метод доступа к памяти вопросов
<i>QueMem</i>	— память вопросов
R^i	— множество ответов, распознаваемых на i -ой диалоговой транзакции
NR^i	— множество ответов, не распознаваемых на i -ой диалоговой транзакции
<i>DiMC</i>	— унифицированный цикл диалогового метода
<i>QueDemon</i>	— специализированный процесс-демон, обслуживающий вопрос
<i>AnsDemon</i>	— специализированный процесс-демон, обслуживающий ответ
<i>Node1</i>	— узел прямой навигации
<i>Node2</i>	— узел условной навигации
<i>Node3</i>	— целевой узел
<i>VObj</i>	— видеообъект сцены
<i>AObj</i>	— аудиообъект сцены
<i>AFact</i>	— факт antecedента продукционного правила
<i>CFact</i>	— факт консеквента продукционного правила
<i>PreAgent</i>	— презентационный агент
<i>PrsAgent</i>	— воспринимающий агент
<i>AnlAgent</i>	— анализирующий агент
<i>TrnAgent</i>	— агент диалоговой транзакции
<i>DiKB</i>	— диалоговая база знаний
<i>DiGen</i>	— редактор диалогового сценария

ВВЕДЕНИЕ

Книга посвящена исследованию диалоговых процессов и их формальному моделированию. Заниматься исследованием диалоговых процессов сложно, поскольку сегодня отсутствует целостное научное направление, посвященное феномену диалогового поведения и его систематическому изучению. Вопросы теории и практики диалоговых процессов обсуждаются в целом ряде различных научных дисциплин. Сюда относятся как сугубо теоретические, философские дисциплины, так и сугубо прикладные, интересующиеся только конкретными диалоговыми системами. Публикации, посвященные исследованию, моделированию и практическому использованию диалоговых процессов можно встретить в эпистемологии, когнитивной психологии, теоретической лингвистике, эротетической логике, искусственном интеллекте, теории человеко-машинного взаимодействия и ряде других.

В книге рассматриваются вопросы, имеющие отношение к естественному диалогу или разговору. Предлагаются элементы теории естественного диалога и модели диалоговой транзакции в контексте представления декларативных знаний участников диалога. Однако большая часть книги посвящена эротетическому диалогу. Отличительной характеристикой большинства прикладных и встроенных систем искусственного интеллекта является наличие диалогового взаимодействия между пользователем и системой, которое характерно наличием логической связи не только между компонентами диалоговой транзакции, но и между самими транзакциями. Среди различных типов диалоговых взаимодействий особое место занимает эротетический диалоговый процесс. Практически все примеры диалоговых систем, опубликованные в литературе, являются эротетическими с вербальным представлением транзакций. В эротетическом диалоговом процессе отдельная транзакция представляет собой вопросно-ответную пару, а сам диалог — целенаправленный процесс обмена вопросами и ответами активного (спрашивающего) и реактивного (отвечающего) агентов диалога. Эротетический диалог проще естественного и в большинстве случаев его возможностей достаточно для решения широкого спектра практически значимых проблем.

Отличительной особенностью книги является ее ориентация на практическую применимость диалоговых моделей. Главной целью является разработка

таких моделей, которые направлены на создание искусственного диалогового агента, способного поддерживать целенаправленный эротетический диалоговый процесс. Модели, описанные в книге, основаны не на алгоритмическом, а на даталогическом подходе к представлению компонентов диалога и диалогового агента. Даталогический подход, как он понимается нами, основан на убеждении в том, что все, что необходимо диалоговому агенту для ведения целенаправленного диалогового процесса, можно представить в виде системы данных, внутренняя организация которых отражает как логическую структуру диалоговой транзакции, так и законы конкретного диалогового поведения. При таком подходе алгоритмическая часть диалоговой системы может быть унифицирована и заниматься, главным образом, интерпретацией даталогического представления потока диалоговых транзакций.

Материал книги разбит на шесть разделов, сгруппированных в две части. Первая часть носит наименование «Естественный и формальный диалоговые процессы» и состоит из разделов 1–4. Эта часть посвящена обзору систем искусственного интеллекта, в которых диалоговый процесс играет ключевую роль, а также описанию ряда формальных моделей естественного и эротетического диалогового процессов.

Вторая часть называется «Диалоговый решатель задач» и состоит из разделов 5–7. Во второй части рассматриваются дедуктивные свойства эротетического диалогового процесса, а также возможность использования целенаправленного эротетического диалога для решения некоторых плохо формализуемых задач. К плохо формализуемым задачам мы относим задачи, обладающие одной или несколькими из следующих характеристик: задачи не могут быть заданы в числовой форме; цели не могут быть выражены в терминах точно определённой целевой функции; не существует алгоритмического решения задачи. Завершается вторая часть коротким разделом, в котором излагаются идеи построения системы, получившей наименование «Проблемно-независимый диалоговый решатель задач».

Первый раздел «Диалоговый процесс в системах искусственного интеллекта» содержит обзор ряда систем искусственного интеллекта, функционирование которых основано на диалоговом процессе. Здесь описаны наиболее известные системы, разработанные в период «расцвета» вопросно-ответных и диалоговых исследований в искусственном интеллекте. Обзор начинается с анализа программы *Eliza*, разработанной Вейценбаумом и относящейся к классу вопросно-ответных систем поддержки естественно-языкового общения. Программа *Eliza* является одной из наиболее известных ранних программ этого класса. К этому же классу относятся диалоговые системы на базе концептуального разборщика Шенка и Теслера, а также диалоговые системы Абельсона на основе прагматического анализа смысла естественно-языковых предложений. Далее проанализированы диалоговые системы и программы,

относящиеся к классу вопросно-ответных систем баз данных и анализа текстов. Одной из известных программ этого класса является программа *DEDUCOM*, разработанная под руководством Слейгла. В рамках класса вопросно-ответных систем баз данных и анализа текстов рассмотрена также программа *QAZ* Грина и Рафаэля и программа *Protosynthex*, предложенная Саймонсом и его коллегами. Завершается первый раздел анализом интеллектуальных систем обучения, которые выделены в класс систем коммуникации знаний. Все известные системы коммуникации знаний разделены на три подкласса: интеллектуальные компьютерные тьюторы; обучающие среды с интеллектуальной поддержкой и интеллектуальные компьютерные репетиторы. История интеллектуальных компьютерных тьюторов начинается с программы *SCHOLAR*, разработанной под руководством Карбонелла. Известна также серия тьюторов *WHY* Стивена и Коллинса, имитирующих метод убеждения, которым пользовался Сократ. Программы под общим наименованием *SOPHIE*, описанные в работах Брауна и его коллег, являются характерным представителем сред с интеллектуальной поддержкой. Эти программы представляют пользователю среду для обучения путём экспериментирования. Наиболее известными программами из класса компьютерных репетиторов являются «когнитивные тьюторы» Джона Андерсона. В первом разделе описаны принципы организации таких программ и проведен анализ когнитивного тьютора, предназначенного для обучения программированию на языке LISP.

Второй раздел «Естественный диалоговый процесс» посвящен исследованию естественного диалогового процесса и разработке элементов теории естественного диалогового процесса. Раздел начинается с анализа когнитивных основ диалогового процесса. На основе циклической модели восприятия, предложенной Нейсером, предлагается модель когнитивного цикла диалога. Модель строится, исходя из предположения о том, что диалоговый процесс аналогичен процессу перцептивного взаимодействия человека с окружающей средой. Отличие заключается в том, что в процессе диалога главные компоненты сенсорной системы человека — зрительный и слуховой анализаторы — подключены не к окружающей среде, а к потокам зрительных и слуховых стимулов, генерируемых противоположным агентом диалога. Ключевым компонентом когнитивного цикла диалога является сценарий диалога, под которым понимается совокупность всех возможных протоколов диалога для некоторой предметной области. Далее следует анализ протоколов естественных диалогов и формирование онтологического базиса естественного диалога. Вводится ряд фундаментальных понятий, необходимых для описания и моделирования диалоговых процессов и систем. Эти понятия в дальнейшем использованы во всех последующих разделах книги. К ключевым понятиям онтологического базиса относятся понятия активного и реактивного диалогового агента, диалоговой транзакции и информационных посылок активного

и реактивного диалоговых агентов. Дается классификация информационных посылок активного и реактивного агентов. Диалоговая транзакция рассматривается с точки зрения обмена порциями декларативных знаний между активным и реактивным агентами. Предлагаются формальные модели диалоговой транзакции в контексте представления декларативных знаний активного и реактивного диалоговых агентов. В основе моделей лежит теория Белмана и Стила о логической структуре диалоговой транзакции эротетического диалога, а также понятийный аппарат языка тернарного описания систем, предложенного Авениром Уемовым.

Третий раздел «Эротетический диалоговый процесс» посвящен разработке моделей диалогового процесса. Модели, описанные в третьем разделе, являются формальными и моделируют эротетический диалоговый процесс. В эротетическом диалоге информационные посылки активного агента имеют стопроцентную степень интеррогативности и могут быть представлены вопросами в лингвистическом смысле слова, а информационные посылки реактивного агента — ответами на эти вопросы. При разработке моделей предполагается, что диалог является гармоничным. Это означает, что роли диалоговых агентов зафиксированы и агенты удовлетворены принятым распределением ролей. Предполагается также, что агенты обладают знаниями, достаточными для достижения цели диалога, а реактивный агент всегда дает правдивые ответы. В разделе последовательно описаны три модели: автоматная модель, Петри-модель и сетевая модель. Автоматная модель представляет собой конечно-автоматную интерпретацию когнитивного цикла диалогового процесса, рассмотренного во втором разделе. Наличие в структуре когнитивного цикла сценария диалога позволяет активному агенту не синтезировать очередной вопрос, а разыскивать его в памяти вопросов, используя в качестве метода доступа к памяти вопросов метод достижения цели диалога. Поэтому в структуру активного диалогового агента введены такие компоненты, как память вопросов, хранящую декларативные знания активного агента в виде информационных посылок-вопросов, и метод доступа к памяти вопросов, обеспечивающий доступ к поименованным элементам памяти вопросов. Петри-модель является развитием автоматной модели. В отличие от автоматной модели, в структуру Петри-модели введен компонент, моделирующий динамику эротетического диалога и названный унифицированным циклом диалогового метода. Цикл диалогового метода управляет переходом от предыдущей транзакции к последующей, что для Петри-модели эквивалентно открытию одного из переходов транзакции и перемещению фишки. Унифицированность цикла диалогового метода означает, во-первых, что для выполнения любой транзакции необходимо выполнить одну и ту же последовательность действий, а во-вторых, что эта последовательность действий не зависит от предметной области диалогового процесса. Петри-модель имеет более широкую сферу

применимости, чем автоматная модель. Её преимущество в том, что она учитывает динамику диалога, а также нестандартную обработку вопросов и ответов при помощи уникальных процессов-демонов. Сетевая модель развивает главные идеи двух предыдущих моделей, но базируется не на математической структуре (конечный автомат или сеть Петри), а на идее схемы базы данных. В сетевой модели существенно следующее: (1) учитываются как принцип глубины диалога, так и принцип истории ответов реактивного агента; (2) развивается даталогический подход и а priori предполагается, что диалоговый метод доступа к памяти вопросов и спецификации вопросов активного агента, хранящиеся в памяти вопросов, представимы данными, хранящимися в некоторой базе данных; (3) развивается сетевое понимание структуры диалогового метода. В предложенном онтологическом базисе и моделях диалогового процесса отсутствуют элементы, явно учитывающие такое антропоморфное свойство реального диалога, как неискренность реактивного агента. В реальном диалоге ответы реактивного агента не всегда правдивы, особенно в том случае, когда вопросы активного агента касаются нежелательных, запретных или опасных для реактивного агента тем. Как правило, одной из целей реактивного агента в ходе диалогового процесса является уклонение от таких тем и сокрытие информации. В заключительной части третьего раздела показано, каким образом можно учитывать неискренность реактивного агента и строить диалоговые методы, позволяющие достигнуть поставленной цели даже в том случае, когда части ответов нельзя доверять.

Четвертый раздел «Даталогическое моделирование структуры вопроса» посвящен моделированию транзакций эротетического диалога на основе даталогического подхода к проектированию систем. Идея даталогического моделирования транзакций основана на убеждении в том, что декларативные знания, ассоциированные с эротетической транзакцией, есть данные. Расширяется онтологический базис эротетического диалога и вводится понятие «сцена», под которой понимается даталогическое представление вопроса. Предлагается моделировать сцену при помощи цепи объектов, интерпретируемых через видео- и аудиоканалы компьютера. В этом случае любой логический компонент эротетической транзакции моделируется некоторым участком цепи объектов сцены. Объект сцены описывается совокупностью атрибутов, необходимых и достаточных для его интерпретации через видео- или аудиоканал. Однако представление объекта сцены исключительно в виде набора атрибутов возможно лишь для простейших объектов. Для случая сложных объектов атрибутивное описание дополняется данными, которые невозможно или нерационально разлагать на атрибуты. Такое описание названо неатрибутивным. Таким образом, любой объект сцены представляется двумя типами описаний: (1) атрибутивными описаниями и (2) неатрибутивным описанием. В четвертом разделе приведен пример реализации идей даталогического

моделирования сцены и описан один из способов атрибутивного описания видеообъекта. Система атрибутивного описания видеообъекта сцены рассматривается как некоторая система классификации, согласованная с внутренней организацией интерпретирующей системы. Относительно организации интерпретирующей системы приняты следующие соглашения: (1) видеообъект представляет собой прямоугольник, имеющий произвольные размеры и отображаемый в различные участки экрана монитора; (2) после того, как процесс отображения видеообъекта на экран монитора завершён, он может подвергаться топологическим преобразованиям в виде поступательного перемещения, вращения и деформации; (3) неатрибутивное описание выводится внутрь видеообъекта; (4) видеообъект является «окном», через которое просматривается часть неатрибутивного описания; (5) видеообъект предназначен для визуальной презентации логических компонентов вопроса и для связи процессов-демонов со сценой; (6) внешним процессам-демонам доступны память истории ответов и проблемная база данных. В примере атрибутивного описания видеообъекта сцены принято, что любой видеообъект описывается при помощи семи классов атрибутов: (1) атрибуты каркаса; (2) атрибуты видеоэффектов; (3) атрибуты редактируемого текста; (4) атрибуты поступательного перемещения; (5) атрибуты деформации каркаса; (6) атрибуты источника информации; (7) атрибуты «времени жизни». В заключительной части четвертого раздела приведена логическая модель памяти вопросов-сцен, построенная на основе даталогического представления эротетической транзакции. Модель описана в виде набора отношений абстрактной реляционной базы данных.

Пятый раздел «Эротетический диалог как метод моделирования процедурных знаний» посвящен, вообще говоря, исследованию связи между эротетическим диалоговым процессом и процессом логического вывода дедуктивного типа. В начальной части раздела рассмотрены моделирующие возможности диалогового процесса эротетического типа. Продемонстрирована сильная аналогия между поведением эксперта при решении плохо формализуемой задачи и поведением активного агента целенаправленного эротетического диалога. Показано, что с точки зрения типологии методов вычислений, алгоритмический метод вычислений является частным случаем диалогового метода вычислений. В последующей части раздела проведен сравнительный анализ того, как моделируются процедурные знания эксперта (знания о способе решения задачи) в эротетической диалоговой системе и в классической экспертной системе, основанной на правилах. На примере анализа протокола работы экспертной системы *MYCIN* сделан вывод о том, что этот протокол может служить примером протокола эротетического диалога с фиксированным распределением ролей, в котором экспертная система *MYCIN* играет роль активного агента. Диалог в *MYCIN*, как и в других экспертных системах, поддерживается

специальным блоком, форма хранения данных в котором отлична от формы хранения фактов и правил в базе знаний. Поэтому для *MYCIN* диалог является способом реализации интерфейсных функций и не является методом решения задачи. В случае эротетического диалога, сценарий, состоящий из диалогового метода доступа и памяти вопросов, напротив, является одновременно и формой хранения знаний, и формой общения с противоположным агентом. В продукционной экспертной системе после того, как некоторый дедуктивный вывод был однажды проведен, соответствующая цепь вывода в сети вывода может быть «свёрнута» в одно эквивалентное правило продукции, включающее факты определенного вида. В том случае, когда возникнет необходимость провести точно такой же вывод, экспертная система может воспользоваться этим правилом вместо повторного построения сети вывода. Отмеченное правило позволяет завершить вывод за n шагов, если имеется метод, позволяющий на каждом шаге получить точно один требуемый факт. Показано, что эротетический диалоговый процесс может быть использован в качестве отмеченного метода получения требуемых фактов. В последующей части раздела рассмотрены когнитивные основы совмещения эротетического диалогового процесса и процесса дедуктивного вывода. Делается вывод о том, что в качестве когнитивной базы такого совмещения может рассматриваться теория автоматических и осознанных ментальных процессов, а также унифицированная модель ментальной деятельности *SOAR*, предложенная Ньюэллом. Заключительная часть раздела посвящена формальному описанию дедуктивных связей в модели диалогового метода доступа к памяти вопросов.

Шестой раздел «Решение плохо формализуемых задач диалоговыми методами» посвящен исследованию применимости формального эротетического диалога для решения ряда плохо формализуемых задач. В разделе рассмотрены некоторые системы, решающие плохо формализуемые задачи: системы персонализированного обучения, электронный текст, расширенный ресурсами, с интеллектуальной поддержкой, интеллектуальные системы автоматизированного проектирования и системы диалогового общения с проблемными базами данных. Персонализированная система электронного обучения понимается как система, осуществляющая автоматическую и непрерывную адаптацию учебного материала и стратегии обучения к текущим когнитивным характеристикам учащегося. Такая система является диалоговым агентом, способным поддерживать целенаправленный эротетический диалоговый процесс, направленный на решение задачи передачи знаний учащемуся. В начальной части раздела приведена классификация обучающих и диагностических стимулов, необходимых для проектирования сценариев персонализированных тьюторов. Текст, расширенный ресурсами, представляет собой специальным образом адаптированную форму гипертекста, разработанную с целью обеспечения учащихся помощью в процессе формирования семантики

изучаемого текстового или мультимедийного материала. Важной функцией интеллектуального текста, расширенного ресурсами, является уменьшение информационной перегрузки учащегося путём пошагового и селективного представления релевантной информации. Легко представить обширный текст, расширенный ресурсами, включающий тысячи страниц и десятки тысяч ключей, связанных с ресурсами, в качестве примера системы, в которой даже опытный учащийся будет в затруднении осуществлять навигацию самостоятельно. Способность эротетической диалоговой системы отображать персонализированную информацию предотвращает информационную перегрузку учащегося. Из сказанного следует, что эротетические диалоговые системы являются наиболее подходящим типом систем для реализации текста, расширенного ресурсами со встроенным интеллектуальным компонентом. В последующей части раздела рассматриваются варианты организации двух видов интеллектуальной системы автоматизированного проектирования: интеллектуальной системы автоматизированного теплофизического проектирования и интеллектуальной системы автоматизированного проектирования на базе графического пакета программ *AUTOCAD*. Предложена трехблочная архитектура системы автоматизированного теплофизического проектирования, включающая: (1) проблемно-ориентированный блок, предназначенный для снабжения расчётного ядра системы данными, необходимыми и достаточными для выполнения расчётов; (2) методо-ориентированный блок, предназначенный для выполнения расчётов; (3) проблемно-ориентированный блок, предназначенный для интерпретации результатов, полученных с помощью блока 2. Второй и третий блоки трехблочной архитектуры решают плохо формализуемые задачи, методы решения которых представляют собой эвристические правила, объединённые логикой эксперта. Рассматривается пример сценария эротетического диалога, способного решать одну из задач. Предложена архитектура двухмашинной системы автоматизированного проектирования общего назначения. Машина диалога, хранящая процедурные знания эксперта в виде сценария эротетического диалога, в ходе диалогового процесса с оператором вырабатывает последовательность команд, реализующих проектные решения, необходимые для изготовления проектной документации. Эта последовательность команд принимается машиной *AUTOCAD*, которая формирует проектную документацию. В заключительной части раздела рассматривается применимость эротетического диалога для проектирования фронтальных программ прикладных систем баз данных. Структурированные языки манипулирования данными в базах данных, такие, например, как SQL позволяют формировать произвольные и непредопределённые запросы к базам данных. Однако эти языки редко используются непосредственно в прикладных системах баз данных, поскольку для их использования необходимы знания как синтаксиса языка, так и логической структуры базы данных.

Пользователю предоставляется фронтальная программа, автоматически формирующая SQL-запрос. В разделе предлагается способ организации фронтальной программы в виде сценария эротетического диалога, реализующего метод инверсного интеррогативного взаимодействия с базой данных.

Седьмой раздел «Организация проблемно-независимого диалогового решателя задач» посвящена синтезу логической структуры инструментальной программной системы, ориентированный на создание диалоговых приложений в различных предметных областях. Диалоговым решателем задач называется система, позволяющая создавать, хранить и редактировать сценарии, необходимые для поддержки эротетического диалогового процесса, ориентированного на решение плохо формализуемых задач. Как декларативные, так и процедурные знания, необходимые для реализации отмеченного диалогового процесса, хранятся в диалоговой базе знаний, которая определяется как совокупность диалогового метода доступа к памяти вопросов и самой памяти вопросов. В разделе приведена и описана структура Диалогового решателя задач в виде UML-диаграммы классов.

ЧАСТЬ 1

**ЕСТЕСТВЕННЫЙ И ФОРМАЛЬНЫЙ
ДИАЛОГОВЫЕ ПРОЦЕССЫ**

1 ДИАЛОГОВЫЙ ПРОЦЕСС В СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

1.1. Ранние вопросно-ответные системы

Исследования, относящиеся к проблематике диалогового процесса и диалоговых систем, выполненные в конце шестидесятых и начале семидесятых годов прошлого века, ориентировались на использование компьютеров с ограниченными возможностями формы представления информации в ходе диалоговых транзакций. Обмен в обоих направлениях осуществлялся с помощью консоли, представляющей входные и выходные сообщения в виде буквенно-цифровых последовательностей. Отмеченная специфика периферийного оборудования компьютеров тех лет оказала влияние на тематику исследований.

Для удобства анализа результатов этих исследований разделим описанные системы на два класса:

- вопросно-ответные системы поддержки естественно-языкового общения;
- вопросно-ответные системы баз данных и анализа текстов.

1.1.1. Вопросно-ответные системы поддержки естественно-языкового общения

Программа, получившая наименование *Eliza*, является одной из наиболее известных ранних программ имитации естественно-языкового общения. Программа была предложена Вейценбаумом в 1966 году [1]. *Eliza* позволяет сколь угодно долго поддерживать естественно-языковый диалог в рамках заранее оговоренной тематики. На рис. 1.1 приведен фрагмент диалога, в котором *Eliza* играет роль врача-психотерапевта, использующего рожерианскую методику побуждения пациента продолжать разговор.

Все мужчины одинаковые.

В КАКОМ СМЫСЛЕ?

Они всегда заставляют нас что-то делать.

ВЫ ДУМАЕТЕ О КАКОМ-ТО КОНКРЕТНОМ ПРИМЕРЕ?

Да, мой приятель заставил меня прийти сюда.

ВАШ ПРИЯТЕЛЬ ЗАСТАВИЛ ВАС ПРИЙТИ СЮДА?

Он говорит, что я почти всегда нахожусь в подавленном состоянии.

МНЕ ЖАЛЬ СЛЫШАТЬ, ЧТО ВЫ В ПОДАВЛЕННОМ СОСТОЯНИИ.

Это правда. Я несчастлива.

ВЫ ДУМАЕТЕ, ЧТО ПРИХОД СЮДА ПОМОЖЕТ ВАМ НЕ БЫТЬ НЕСЧАСТЛИВОЙ?

Мне нужна чья-то помощь, и это — несомненно.

ЧТО БУДЕТ ЗНАЧИТЬ ДЛЯ ВАС, ЕСЛИ ВЫ ПОЛУЧИТЕ ПОМОЩЬ?

Возможно, я смогу научиться жить со своей мамой.

РАССКАЖИТЕ МНЕ О ВАШЕЙ СЕМЬЕ.

Моя мама заботится обо мне.

КТО ЕЩЁ ИЗ СЕМЬИ ЗАБОТИТСЯ О ВАС?

Мой отец.

ВАШ ОТЕЦ?

Вы, в некотором смысле, похожи на моего отца.

КАКОЕ СХОДСТВО ВЫ ВИДИТЕ?

Вы не очень агрессивны, но, я думаю, вы не хотите это показывать.

ЧТО ЗАСТАВЛЯЕТ ВАС ДУМАТЬ, ЧТО Я НЕ ОЧЕНЬ АГРЕССИВЕН?

Вы не спорите со мной.

ПОЧЕМУ ВЫ ДУМАЕТЕ, ЧТО Я НЕ СПОРЮ С ВАМИ?

Вы меня боитесь.

ВАМ ПРИЯТНО ОТ ТОГО, ЧТО ВЫ ДУМАЕТЕ, ЧТО Я БОЮСЬ ВАС?

Мой отец всех боялся.

О ЧЁМ ЕЩЁ ВЫ ВСПОМИНАЕТЕ, КОГДА ДУМАЕТЕ О СВОЁМ ОТЦЕ?

Рис. 1.1. Фрагмент диалога с программой *Eliza*.

Предложения, генерируемые программой, записаны прописными буквами

Имитировать психотерапевта-роджерсианца сравнительно легко, так как его метод ведения разговора основан на вовлечении пациента в беседу путем повторения ему его же собственных высказываний. Программа, генерирующая выходные предложения, представляет собой множество преобразователей, называемых также трансформаторами. Каждый трансформатор включает шаблон выходного предложения и оператор. Шаблон — это частично сформированное выходное предложение, а оператор дополняет шаблон словом или словосочетанием, экстрагированным из входного предложения диалогового партнера. Выбор трансформатора осуществляется на основании распознавания ключевых слов во входном предложении. Автор называет сценарием набор правил, каждое из которых включает как трансформатор (правая часть правила), так и шаблон ключевых слов, используемый при анализе входного предложения (левая часть

правила). На рис. 1.2 приведен пример, иллюстрирующий структуру правила сценария диалога *Eliza*.

Входное предложение	You are very helpful
Правило сценария	(0 you are 0) = (What makes you think I am <4>)
Выходное предложение	What makes you think I am very helpful?

Рис. 1.2. Структура правила сценария, специфицирующего диалог программы *Eliza*

В начальной фазе алгоритма трансформации из входного предложения выделяются ключевые слова “*you are*” и разыскивается соответствующее правило путем анализа шаблонов, находящихся в левой части правил. В примере, приведенном на рис. 1.2, выбрано правило с шаблоном (0 you are 0). Затем *Eliza* структурирует входное предложение следующим образом:

1: (empty); 2: you; 3: are; 4: very helpful.

Структурирование позволяет определить словосочетание, которое должно быть подставлено в шаблон правой части правила. В рассматриваемом примере это словосочетание “very helpful”. Выходное предложение формируется из правой части правила после соответствующей подстановки.

Отметим следующие отличительные особенности программы *Eliza*, важные для проблематики диалогового поведения.

- *Eliza* является сугубо диалоговой программой. Информация, содержащаяся в протоколе диалога, распределена между партнерами. *Eliza* не может функционировать, не используя информацию партнера, поскольку процесс формирования очередной транзакции включает анализ входного предложения.
- Выходное предложение определяется только входным предложением и образуется путем его трансформации.
- Трансформатор входного предложения в выходное представляет собой эмпирическое правило, сводимое к продукционному. Так, например, правило, приведенное на рис. 1.2, можно переписать в следующей нотации:

ЕСЛИ <входное предложение содержит “*you are*” >
 ТО <выходное предложение это шаблон: «*What makes you think I am <4>*» с подстановкой 4-го элемента структурного разбора входного утверждения>

- Множество входных предложений бесконечно или, по крайней мере, очень велико, а множество ключевых слов или словосочетаний, при помощи которых распознаются входные предложения, конечно и равно количеству правил. На каждом шаге *Eliza* должна просматривать всё множество правил.
- Поведение программы *Eliza* не является целенаправленным, а диалог, который она поддерживает, не является “проблемно-решающим”. Разговор с программой может быть прекращён принудительно в любом месте, а не автоматически в результате достижения цели.
- *Eliza* не запоминает входные предложения и, следовательно, не использует ранее введенные входные предложения для формирования выходных предложений или логического вывода.
- Распределение ролей между партнерами диалога не является существенным для функционирования программы. Как *Eliza*, так и её собеседник могут функционировать в роли или задающего вопросы, или отвечающего партнера.
- *Eliza* не является “хранителем” специфического диалогового метода решения задачи.

Несмотря на то, что автор программы *Eliza* весьма скептически относился к её успеху как к демонстрации общего решения задачи моделирования естественно-языкового диалога [2], программа серьёзно изучалась с точки зрения её применимости для решения ряда практических задач. Так, например, Тэйлор [3] исследовал применимость программы *Eliza* в области компьютерно-ориентированного обучения, а Эллис и Тидеман [4] — в области компьютерного моделирования поведения репетитора. В обоих случаях авторы сделали многообещающие выводы.

Колби с коллегами в серии публикаций [5–8] описал диалоговую естественно-языковую систему, которая первоначально развивала метод программы *Eliza*, поскольку использовала множество шаблон-ориентированных трансформаторов. Отмеченная система в ряде публикациях называлась “система убеждений” (belief system). Целью диалога было накопление программой предложений партнёра, характеризующих его убеждения. Программа запоминала входные предложения и, таким образом, исключала возможность алогичного диалога, например:

Когда мой отец умер, моя мать долго болела.

РАССКАЖИТЕ МНЕ БОЛЬШЕ О НЁМ.

Мой отец был писателем. Он был добр, но немного циничен...

ОН И СЕЙЧАС ТАКОЙ?

Он мёртв.

На более поздней фазе исследований для моделирования системы убеждений использовалась графовая модель, которая описывала как отношения между множеством концептов, характеризующих убеждения, так и цепи логического вывода. Графовая модель содержала встроенные правила вывода типа: « A имплицирует B », где антецедент интерпретируется как множество условий, а консеквент — как психологическое ожидание. Таким образом, отмеченное правило вывода может быть переформулировано в виде: «при выполнении множества условий A ожидается B ».

Общим свойством как программы *Eliza*, так и belief system Колби является ориентация на использование при генерации выходных предложений синтаксических, а не семантических структур.

Шенк и Теслер [9] описали семантическую систему для возможного использования совместно с belief system Колби, названную ими концептуальный разборщик (conceptual parser). Система базировалась на зависимости грамматики входного предложения от концептуальной базы данных, отображающей некоторую «семантическую реальность». Разборщик генерирует множество фактов, семантически связанных с входным предложением. Например, из входного предложения

John saw Texas flying to California

разборщик генерирует факты

$$\begin{array}{ccc} & P & \uparrow P \text{ to} \\ \text{John} \Leftrightarrow \text{see} \leftarrow \text{Texas} & & \text{John} \Leftrightarrow \text{fly} \leftarrow \text{California,} \end{array}$$

где символ \Leftrightarrow обозначает отношение субъект-предикат, символ \leftarrow — прямое дополнение, символ \leftarrow — предложную зависимость, символ \uparrow — модификатор, а метка P — прошедшее время. Концептуальный разборщик положил начало работы Шенка и его коллег над теорией и методами обработки естественного языка. Предложенная ими в [10] теория, названная теорией концептуальной зависимости, рассматривает проблему обработки естественного языка в трёх аспектах.

- Отображение входных предложений в их смысловое представление.
- Хранение предложений в памяти и осуществление умозаключений относительно полученного смысла.
- Перевод смыслового представления на естественный язык.

Практическим выходом отмеченных исследований является программа *MARGIE* (*Memory, Analysis, Response Generation In English*), имеющая два режима работы: перефразирование и умозаключение. На рис. 1.3 приведены примеры работы программы *MARGIE*, дающие представление о её возможностях [11].

Режим перефразирования.

Вход: Джон посоветовал Мэри выпить вина.

Выход: Джон сказал Мэри, что питьё вина будет ей полезно.

Вход: Джон убил Мэри, задушив её.

Выход1: Джон задушил Мэри.

Выход2: Джон душил Мэри, и она умерла, потому что она не могла дышать.

Выход3: Мэри умерла, потому что она не могла вдохнуть воздух, и она не могла вдохнуть воздух, потому что Джон схватил её за шею.

Режим умозаключений

Вход: Джон сказал Мэри, что Биллу нужна книга.

Выход1: Книга о чём?

Выход2: Мэри знает, что Биллу нужна книга.

Выход3: Билл хочет прийти за книгой.

Выход4: Билл хочет, чтобы у кого-то больше не было книги.

Выход5: Билл хочет читать книгу.

Вход: Джон побил Мэри палкой.

Выход1: Палка касалась Мэри.

Выход2: Мэри было больно.

Выход3: Джон хотел, чтобы Мэри было больно.

Выход4: Почему Джон был злой?

Рис. 1.3. Примеры работы программы *MARGIE* в режимах перефразирования и умозаключений

В программе может храниться сколь угодно большое количество возможных для данного диалогового поведения реакций, а на каждом шаге диалога для распознавания предложения-реакции просматривается не всё множество возможных реакций (как это делает, например, *Eliza*), а только ассоциированное с текущим шагом множество ожидаемых реакций.

Таким образом, судя по имеющимся публикациям, в *MARGIE* предложено решение задачи формирования множества ожидаемых реакций партнера диалога. Решение этой задачи является необходимым условием построения системы естественно-языкового диалога. Программа *MARGIE* была ориентирована на обработку предложений, а не текста. В дальнейшем Шенк и его коллеги разработали серию программ, ориентированных на обработку текстов [11].

Программа *SAM* (*Script Applier Mechanism*) работает с текстом, дополненным сценарием, который представляет собой множество правил, описывающих связи между отдельными предложениями. Программа *SAM* была применена

для автоматического перевода газетных объявлений о дорожно-транспортных происшествиях. Выбор этой категории текстов был обусловлен тем, что все газетные сообщения о дорожно-транспортных происшествиях в США стереотипны и могут быть описаны одним сценарием. Для работы с текстами при помощи метода, использованного в *SAM*, сценарий должен быть разработан а priori для каждого текста, что является существенным недостатком метода.

Программа *PAM (Plan Applier Mechanism)* может работать с текстом даже в том случае, если он не снабжен сценарием. Приводятся различные примеры работы *PAM*, в которых программа демонстрирует способность понимать текст путем описания одного и того же события с точки зрения различных участников этого события. Получив на вход текст, описывающий, например, ограбление магазина, программа продуцирует на выходе два текста, описывающих ограбление как с точки зрения владельца магазина, так и с точки зрения грабителя.

Программа *POLITICS*, судя по приведенным примерам её работы, в состоянии поддерживать осмысленный диалог консультирующего типа, выступая в роли отвечающего партнера. На рис. 1.4 приведен фрагмент примечательного диалога на тему: «Конгресс США ожидает утверждения договора о Панамском канале», в котором *POLITICS* играет роль американского сенатора-консерватора.

- Q1: Должны ли США одобрить договор?
A1: НЕТ, ЭТОТ ДОГОВОР ПЛОХОЙ ДЛЯ СОЕДИНЁННЫХ ШТАТОВ.
Q2: Почему этот договор плохой для США?
A2: СОЕДИНЁННЫЕ ШТАТЫ ПРОИГРАЮТ КАНАЛ В ПОЛЬЗУ ПАНАМЫ И СТАНУТ СЛАБЕЕ.
Q3: Что может произойти, если США потеряют Канал?
A3: РОССИЯ БУДЕТ ПЫТАТЬСЯ УПРАВЛЯТЬ КАНАЛОМ.
Q4: Почему Россия будет пытаться контролировать Канал?
A4: РОССИЯ ХОЧЕТ РАСШИРИТЬ ВОЕННОЕ ПРИСУТВИЕ. КАНАЛ ИМЕЕТ БОЛЬШОЕ ВОЕННОЕ ЗНАЧЕНИЕ.
Q5: Как это может произойти?
A5: США БУДУТ НЕ В СОСТОЯНИИ ОСТАНОВИТЬ КОГО-БЫ ТО НИ БЫЛО КОНТРОЛИРОВАТЬ КАНАЛ. ТОРРИЛОС КОММУНИСТ. ТОРРИЛОС РАЗРЕШИТ РОССИИ КОНТРОЛИРОВАТЬ ПАНАМУ.
Q6: Почему США должны беспокоиться, что Россия будет контролировать Канал?
A6: ПОТОМУ ЧТО РОССИЯ ХОЧЕТ УПРАВЛЯТЬ МИРОМ.
Q7: Что должны делать США, чтобы не дать России контролировать Канал?
A7: КОНГРЕСС ДОЛЖЕН ПРОГОЛОСОВАТЬ ПРОТИВ ДОГОВОРА О ПАНАМСКОМ КАНАЛЕ.

Рис. 1.4. Фрагмент диалога консультирующего типа программы *POLITICS*.
Ответы программы записаны прописными буквами

К сожалению, среди доступных публикаций отсутствует описание методов, лежащих в основе программы *POLITICS*. Отмечается лишь, что эти методы учитывают такие ментальные феномены, как система убеждений, логический вывод, планирование, целенаправленность, сценарное поведение.

Абельсон [12] использовал прагматический анализ для понимания смысла предложений. Например, на утвердительное входное предложение “I went to three drugstores” возможны следующие различные реакции, в зависимости от ориентации системы.

Синтаксически ориентированная система: How did you go to three drugstores?

Семантически ориентированная система: What useful things did you buy in three drugstores?

Прагматически ориентированная система: How come the first two drugstores didn't have what you wanted?

Прагматическим анализом Абельсон называет процесс генерации множества «правдоподобных импликаций». В качестве одного из возможных подходов к прагматическому анализу Абельсон предложил концепцию «импликационной молекулы», представляющую собой множество классов предложений, связанных между собой психологической импликацией. Иными словами, предложения представляют собой пропозициональные элементы, а импликационные связи объединяют их в импликационную молекулу. Примером импликационной молекулы, обозначаемой как

$$\text{PURPOSE}(Y,A,X),$$

может служить множество

$$\{A \text{ does } X, X \text{ causes } Y, A \text{ wants } Y\}$$

Абельсон постулирует тенденцию к завершенности как свойство человека — пользователя языка. Поэтому он считает, что наличие нескольких элементов импликационной молекулы достаточно, чтобы получить остальные путем логического вывода. Таким образом, если дано, что $A \text{ does } X$ и $X \text{ causes } Y$, то вероятен вывод, что $A \text{ wants } Y$. Правдоподобно также, что, имея только $A \text{ does } X$, можно вывести, что $X \text{ causes some } Y \text{ that } A \text{ wants}$.

Приведенная выше импликационная молекула с именем *PURPOSE* позволяет программе делать вывод, что A сделало X с намерением Y . Импликационные молекулы расширяют возможности программы естественно-языкового общения и позволяют при формировании выходного предложения предугадывать логические посылки и намерения партнёра.

Абельсон отмечает, что использование импликационных молекул совместно с системами типа *belief system* существенно расширяет возможности

“разговаривающей” программы по генерированию логичных реакций на входные предложения.

1.1.2. Вопросно-ответные системы баз данных и анализа текстов

Вопросно-ответные системы, ориентированные на работу с базами данных, отличаются две особенности. Во-первых, фиксированное распределение ролей диалоговых партнеров, при котором пользователь всегда играет роль спрашивающего партнера, а программа — роль отвечающего партнера. Во-вторых, ответы на вопросы синтезируются в результате дедуктивных умозаключений.

Одной из известных программ этого класса является *DEDUCOM* (*DEDUctive COMmunicator*), разработанная под руководством Слейгла [13]. *DEDUCOM* представляет собой Lisp-программу, которая хранит некоторое множество фактов в виде Lisp-выражений вида

На кисти имеется 5 пальцев,
На руке имеется одна кисть,
У человека имеется 2 руки,

и правила вывода в виде условных предложений с переменными вида

IF there are **m X**'s on a **V**
AND IF there are **n V**'s on a **Y**,
THEN there are **mn X**'s on a **Y**.

Путём подстановки фактов в правила вывода программа *DEDUCOM* в состоянии ответить на вопрос: How many fingers on a man?

В процессе экспериментов программа *DEDUCOM*, после сообщения ей 68 фактов, ответила на 10 вопросов, касающихся этих фактов. В дальнейшем Слейгл предложил программу *DEDUCOM II*, отвечающую на вопросы путём дедуктивного вывода на основе метода резолюций. [14] На рис. 1.5. приведен фрагмент гипотетического диалога с *DEDUCOM II*, дающий представление как о языке общения с программой, так и о ее дедуктивных способностях. Спрашивающий партнер — пользователь предлагает программе либо утверждения-факты, начинающиеся с буквы s, либо вопросы, начинающиеся с буквы q.

Как следует из фрагмента, приведенного на рис. 1.5, общение с программой асимметрично с точки зрения использования естественного языка. Утверждения и вопросы спрашивающего партнера-пользователя кодируются при помощи специальной скобочной нотации, а ответы, генерируемые программой, представляют собой полноценные фразы естественного языка.

s(in John boy)
 OK
 q(in John boy)
 YES
 q(exist **x** (in John **x**))
 YES
 q(find **x** (in John **x**))
 YES WHEN **X** IS BOY
 q(in Mike boy)
 I CANNOT PROVE THE STATEMENT FROM THE FACTS I KNOW
 s(not (in John boy))
 YOUR STATEMENT OBVIOUSLY CONFLICTS WITH PREVIOUS FACTS
 s(not (in Mary boy))
 OK
 q(is (in Mary boy))
 NO
 s(fact **x** (if (in **x** boy) (in **x** human)))
 OK
 q(find **x** (in **x** human))
 YES WHEN **X** IS JOHN

Рис. 1.5. Фрагмент диалога с программой *DEDUCOM II*. Пользователь предлагает программе утверждения-факты, начинающиеся с буквы *s*, и вопросы, начинающиеся с буквы *q*. Ответы, генерируемые программой, записаны прописными буквами

Аналогичная скобочная нотация используется в программе, предложенной Эллиотом [15]. Каноническая форма этой нотации может быть представлена в виде

$$(\text{Оператор}(\text{Данное1})(\text{Отношение})(\text{Данное2}))$$

Необходимость использования приведенной скобочной нотации для кодирования утверждений и вопросов активного диалогового партнёра объясняется желанием разработчиков системы ускорить обработку и сократить время реакции.

Вопросно-ответная система *QAZ* Грина и Рафаэля [16, 17] основана на использовании процедуры доказательства теорем на базе метода резолюций. Принципы вопросно-ответной системы Грина и Рафаэля относительно просты. Система знаний о мире вопросно-ответной системы сформулирована в виде набора аксиом, а задаваемые ей вопросы представляются как подлежащие доказательству теоремы. Процесс доказательства теоремы, таким образом, совпадает с процессом ответа на вопрос. Например, факт «Джордж дома» представляется как аксиома $AT(\text{George}, \text{home})$. Вопрос «Дома ли Джордж?» выражается как теорема $AT(\text{George}, \text{home})$. Если доказано, что теорема истинна, ответ гласит: «Да».

Система, способная доказывать теоремы, может быть использована также для нахождения или построения объектов, удовлетворяющих некоторым специальным условиям. Например, вопрос «Где Джордж?» требует отыскания некоторого места x , удовлетворяющего условию $AT(\text{George}, x)$. Система доказательства теорем является частью более общей системы, которая управляет ею, распоряжается базой данных и взаимодействует с пользователем.

На рис. 1.6 приведен пример диалога с *Q43*. Входные предложения записаны в нотации логики предикатов первого порядка. Для вопросов 4, 8, 10 и 12 ответом является терм, выработанный процедурой доказательства. Такого рода ответы называются «конструктивными».

Кроме ответов, приведенных на рис. 1.6, имеются еще два типа ответов: “NO PROOF FOUND” и “INSUFFICIENT INFORMATION”. Система выводит ответ “NO PROOF FOUND” в том случае, когда, израсходовав все внутренние ресурсы, она не смогла ни доказать, ни опровергнуть входное предложение. Ответ “INSUFFICIENT INFORMATION” выводится в том случае, когда ни ответ «да», ни ответ «нет» не могут быть получены из имеющихся данных.

1. Statement: $\text{man}(\text{Smith})$
OK
2. Question: $\text{man}(\text{Smith})$
YES
3. Statement: $(\forall x) [\text{man}(x) \Rightarrow \text{animal}(x)]$
OK
4. Question: $(\exists y) \text{animal}(y)$
ANSWER: YES, $y = \text{Smith}$
5. Statement: $(\forall x) [\text{robot}(x) \Rightarrow \text{machine}(x)]$
OK
6. Statement: $\text{robot}(\text{Rob})$
OK
7. Statement: $(\forall x) [\text{machine}(x) \Rightarrow \neg \text{animal}(x)]$
OK
8. Question: $(\forall x) \text{animal}(x)$
ANSWER: NO, $x = \text{Rob}$
9. Statement: $AT(\text{Smith}, \text{work}) \vee AT(\text{Jones}, \text{work})$
OK
10. Question: $(\exists x) (AT(x, \text{work}))$
ANSWER: YES, $x = \text{Smith}$ OR $x = \text{Jones}$.
11. Statement: $(\forall z) P(z, f(z))$
OK
12. Question: $(\forall x) (\exists y) P(x, y)$
ANSWER: YES, $y = f(x)$

Рис. 1.6. Пример диалога с программой *Q43*. Ответы, генерируемые *Q43*, записаны прописными буквами

Как это видно из приведенных примеров, вопросно-ответные системы, ориентированные на работу с базами данных, с точки зрения использования естественного языка, отличаются тем, что, во-первых, функционируют в некотором узком подмножестве естественного языка, а во-вторых, это подмножество хранится в предопределенной базе данных в виде коротких утверждений-фактов.

Вопросно-ответные системы, ориентированные на анализ текстов, отличаются возможностью манипулирования гораздо более широким подмножеством естественного языка, а также встроенной моделью пользователя языка. Примером может служить система *Teachable Language Comprehender*, предложенная Квилианом [18], базирующаяся на им же предложенной структурной модели семантической памяти [19].

Система *Protosynthex III*, предложенная Саймонсом и его коллегами [20], успешно анализирует широкий спектр английских утвердительных и вопросительных предложений, дедуктивно отвечает на многие типы вопросов и реагирует либо ответом, либо перефразированным входным утверждением. Отмечается, что *Protosynthex III* не отвечает на «как» и «почему» вопросы. Система была проверена на множестве вопросов, выбранных из Комптоновской энциклопедии (*Compton's Encyclopedia*). Например:

Кто проиграл битву при Ватерлоо?

НАПОЛЕОН КОМАНДОВАЛ ФРАНЦУЗСКОЙ АРМИЕЙ, КОТОРАЯ ПРОИГРАЛА БИТВУ ПРИ ВАТЕРЛОО В 1815 ГОДУ

В основе системы *Protosynthex III* лежит предложенная модель концептуальной структуры семантической памяти человека, представляемая при помощи вложенных триплетов (Концепт-Отношение-Концепт). С помощью отмеченной модели предложение: «The angry pitcher struck the umpire who called the game» (Злой подающий ударил судью, который начал игру) будет представлено при помощи следующего множества триплетов.

((((Pitcher MOD angry) TMOD the)(strike T past),
(umpire SMOD (umpire(call T past)(game TMOD the))))

Каждый терм в триплете представляет собой однозначный и недвусмысленный выбор смыслового значения слова. Каждый средний терм триплета — отношение (не обязательно хорошо определенное). Структура рассматривается как формальный язык, который выражает семантику предложения в виде множества вложенных триплетов-отношений. Трансформация из естественного языка в формальный осуществляется при помощи восходящего синтаксического анализа с использованием алгоритма Кока (*Cocke*) [21].

После того, как сложный вопрос трансформирован в отмеченный формальный язык, каждый триплет интерпретируется как элементарный вопрос, а исходный, сложный вопрос, как множество вложенных элементарных вопросов.

Ответы на элементарные вопросы первоначально разыскиваются в накопленной базе данных. Если прямой поиск ответа является безуспешным, то система пытается вывести ответ при помощи дедуктивного процесса. Дедукция основывается на множестве правил, представляющих собой либо программные функции, либо имеют вид шаблона-оператора. Например:

$$((A \text{ sister } B) \text{ and } (B \text{ mother } C) \Rightarrow (A \text{ aunt } C))$$

1.2. Диалог и дедукция в интеллектуальных системах обучения

Диалоговый и дедуктивный процессы являются неотъемлемыми атрибутами интеллектуальных систем обучения. Идеальную интеллектуальную систему обучения можно описать следующим образом. Система в процессе взаимодействия с пользователем непрерывно «подстраивается» к его текущему уровню знаний и степени понимания; выбирает учебный материал, для которого имеет место несоответствие в уровне знаний и/или степени понимания; автоматически или по запросу пользователя дополняет учебный материал тематически связанной информацией (например, справочного характера); уделяет больше внимания «трудным местам» и пропускает уже известный материал; тестирует пользователя и определяет его когнитивные характеристики; проводит процесс обучения методом, соответствующим текущим когнитивным способностям и когнитивному стилю пользователя.

Очевидно, что отмеченное поведение системы возможно только при наличии постоянного и целенаправленного информационного обмена с пользователем. Поэтому Венгер [22], подчеркивая коммуникационный характер поведения таких систем, назвал их системы коммуникации знаний (knowledge communication systems). Коммуникация знаний определяется им как способность инициировать и поддерживать целенаправленный обмен знаниями между двумя носителями знаний при помощи ограниченного набора коммуникационных ресурсов.

Поскольку термин «коммуникация знаний» не является широко распространенным, представляется целесообразным дать некоторые пояснения. Отметим, во-первых, что специфицирование ограниченного множества коммуникационных ресурсов, таких как язык, визуальные образы или набор упражнений исключают полный и непосредственный контроль внутренних состояний системы. Во-вторых, приведенное понимание коммуникации знаний достаточно широкое, поскольку предполагает широкий спектр способов приобретения чьих-то

знаний кем-то другим. В-третьих, приведенное определение коммуникации знаний никак не ограничивает ни способы представления знаний, ни способы их коммуникации. Это необходимо для того, чтобы не ограничивать разработчиков систем коммуникации знаний в их выборе способов представления и коммуникации знаний.

Коммуникация знаний может рассматриваться как отдельный раздел дисциплины «искусственный интеллект» наряду с такими разделами, как решение проблем, распознавание образов и др. Коммуникация знаний — это широко распространённое интеллектуальное поведение. И если человек рассматривается как хороший пример интеллектуальной системы, то способность к коммуникации знаний является одним из необходимых условий его интеллектуальности.

Все известные системы коммуникации знаний разделим на три класса:

- интеллектуальные компьютерные тьюторы;
- обучающие среды с интеллектуальной поддержкой;
- интеллектуальные компьютерные репетиторы.

С точки зрения декларативно-процедурной дихотомии знаний, передаваемых в процессе обучения, приведенные классы систем коммуникации характеризуются следующей ориентацией. Интеллектуальные компьютерные тьюторы ориентированы на передачу, главным образом, декларативных знаний, а обучающие среды и компьютерные репетиторы — на передачу процедурных знаний или навыков.

1.2.1. Интеллектуальные компьютерные тьюторы

Одной из первых систем коммуникации знаний является программа *SCHOLAR*, предложенная Карбонелом [23]. Существенной, для проблематики диалогового поведения, характеристикой *SCHOLAR* является её способность поддерживать диалог, в котором роли партнёров не зафиксированы. Это означает, что в процессе обучения вопросы может задавать не только система, но и пользователь. *SCHOLAR* использует ограниченный английский язык. На рис. 1.7 приведен фрагмент диалога пользователя и *SCHOLAR*, который иллюстрирует свободное распределение ролей партнёров.

Следующей существенной характеристикой *SCHOLAR* является способ моделирования предметной области диалога (география Южной Америки) и экспертных знаний тьютора. В этой системе впервые, в качестве абстрактной среды для построения модели, используется концепция семантической сети. Карбонелл возлагал большие надежды на семантические сети и ввёл понятие модель идеальных знаний пользователя, или модель идеального студента (*perfect student*), как синоним понятия модели предметной области в виде полной семантической сети. Пример фрагмента модели предметной области *SCHOLAR* приведен на рис. 1.8.

КАКОВА ПРИБЛИЗИТЕЛЬНАЯ ПЛОЩАДЬ БРАЗИЛИИ?

2,500,000 квадратных миль.

НЕВЕРНО. ПОЖАЛУЙСТА, ОТМЕТЬТЕ, ПРАВИЛЬНО ЛИ СЛЕДУЮЩЕЕ УТВЕРЖДЕНИЕ: «ПЛОЩАДЬ ПАРАГВАЯ ОКОЛО 47,432 КВАДРАТНЫХ МИЛЬ»

Какова площадь Бразилии?

ПЛОЩАДЬ БРАЗИЛИИ ОКОЛО 3,600,000 КВАДРАТНЫХ МИЛЬ.
ТЕПЕРЬ ОТВЕЬТЕ НА ВОПРОС, НА КОТОРЫЙ ВЫ НЕ ОТВЕТИЛИ РАНЕЕ.

Ложно.

ПЕРЕФРАЗИРУЙТЕ ОТВЕТ. Я НЕ МОГУ ПОНЯТЬ ЕГО.

Неправильно.

ОЧЕНЬ ХОРОШО. ПЛОЩАДЬ ПАРАГВАЯ ОКОЛО 157,000 КВАДРАТНЫХ МИЛЬ.
В БРАЗИЛИИ ГОВОРЯТ ПО-ФРАНЦУЗСКИ. ТАК ИЛИ НЕТ?

Не так.

ВЫ ПРАВЫ. В БРАЗИЛИИ ГОВОРЯТ ПО-ПОРТУГАЛЬСКИ.
ПРАВДА ЛИ, ЧТО СТОЛИЦЕЙ БРАЗИЛИИ ЯВЛЯЕТСЯ БРАЗИЛИЯ?

Да.

ОТЛИЧНО. ПРИМЕРНО, КАКОВО НАСЕЛЕНИЕ БРАЗИЛИИ?

60,000,000.

ПРИМЕРНО ВЕРНО, НО ТОЧНЕЕ — 70,000,000. ЧТО ТАКОЕ БРАЗИЛИЯ?

Расскажите мне о Перу.

МИНУТКУ. Я ПРОВЕРЯЮ ФАЙЛЫ. ТЕПЕРЬ ПРОДОЛЖИМ.
ПЕРУ — ЭТО СТРАНА. ОНА НАХОДИТСЯ В ЮЖНОЙ АМЕРИКЕ.
НАСЕЛЕНИЕ ПРИМЕРНО — 7,000,000 ЧЕЛОВЕК. СТОЛИЦА — ЛИМА.
ТЕПЕРЬ ОТВЕЬТЕ НА ВОПРОС, НА КОТОРЫЙ ВЫ НЕ ОТВЕТИЛИ РАНЕЕ.

в/о.

ХОТИТЕ ЛИ ВЫ ИЗМЕНИТЬ СПОСОБ ИНТЕРАКТИВНОСТИ?
ОТВЕЧАЙТЕ, ПОЖАЛУЙСТА, «ДА» ИЛИ «НЕТ».

Да.

ХОРОШО МР. ДЖОНС. ПРОДОЛЖИМ.

Пожалуйста, расскажите мне больше о Перу.

ПЛОЩАДЬ ПЕРУ ОКОЛО 480,000 КВАДРАТНЫХ МИЛЬ.
В ПЕРУ ГОВОРЯТ ПО-ИСПАНСКИ.

Рис. 1.7. Фрагмент диалога пользователя и *SCHOLAR*.
Предложения, генерируемые *SCHOLAR*, записаны прописными буквами

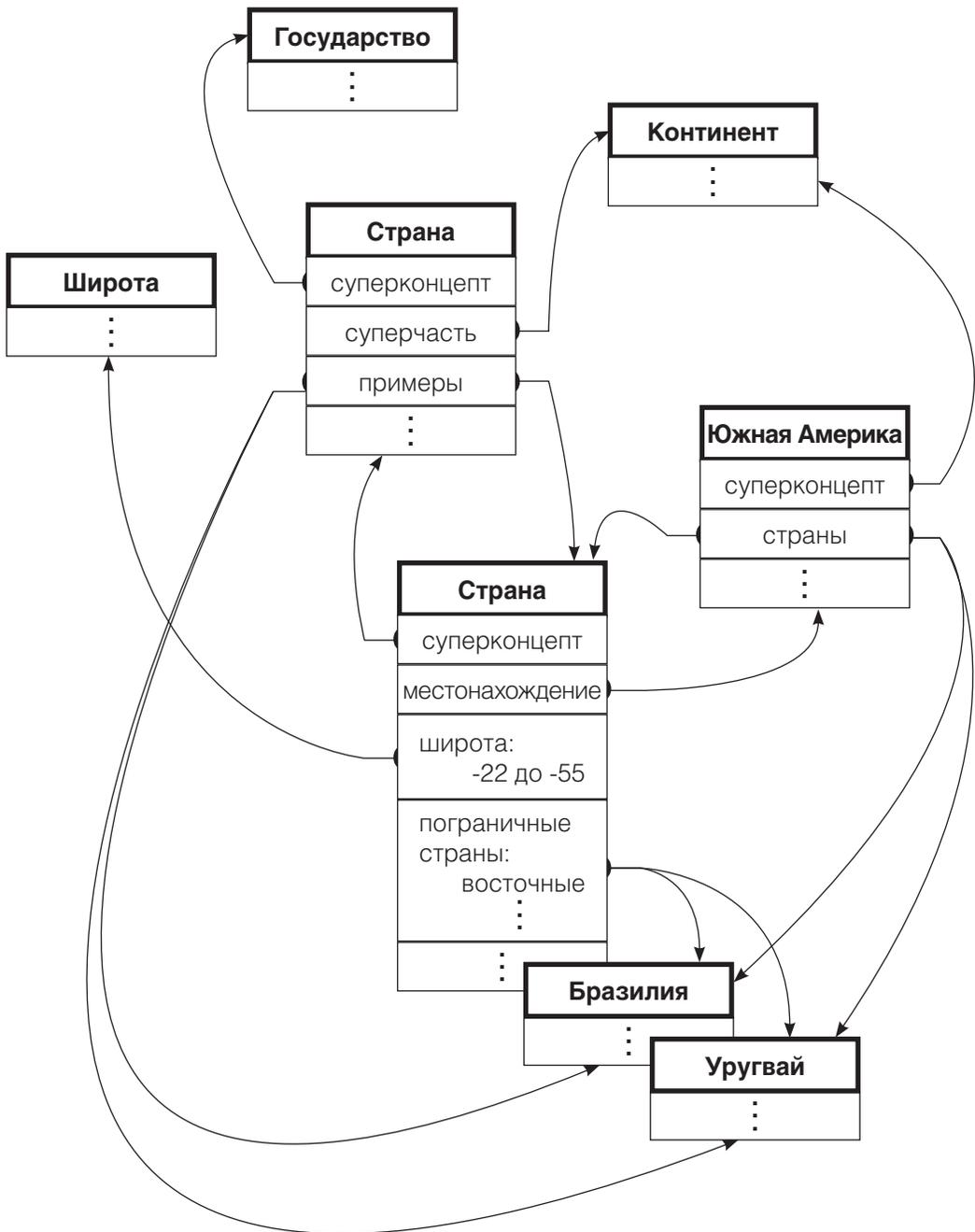


Рис. 1.8. Фрагмент сетевой модели предметной области диалога, используемой в *SCHOLAR*

Идеи первоначальной версии *SCHOLAR* получили развитие в работах Коллинса и его коллег [24, 25] в направлении дальнейшего развития механизма

вывода и стратегии обучающего диалога. Развитие механизма вывода осуществлялось в направлении моделирования механизмов вывода, выполняемого человеком в условиях неопределенности, или, так называемого, правдоподобного вывода (*plausible inference*).

Например, человек может сделать правдоподобное заключение о том, что на континенте не производится сахар, из знаний о том, что ни в одной стране континента не производится сахар и что он располагает полным списком стран. Ясно, что получить такой вывод путем простой навигации по узлам сети невозможно, так как невозможно включить в семантическую сеть все сведения о том, что не производится в Южной Америке.

Среди прочих предложений по дальнейшему развитию механизма вывода *SCHOLAR* отметим технику вывода, названную авторами функциональным анализом, под которым понимался анализ фактов, от которых зависела некоторая гипотеза. Например, *SCHOLAR* получила возможность делать вывод о климате в регионе путем анализа его местоположения.

Развитие стратегии обучения осуществлялось в направлении замены процедуры случайного выбора локальных тем, процедурой, действующей на основе набора эвристических правил, полученных из наблюдений за работой опытных тьюторов. Тема, выбранная либо на основе отмеченных эвристических правил, либо по запросу пользователя, динамически включалась в *план занятия*, и ей выделялось время на обсуждение, пропорциональное её важности. Таким образом, динамически редактируемый план занятий становится главной управляющей структурой тьюторского диалога *SCHOLAR*.

Диалоговая система обучения *WHY*, предложенная Стивенем и Коллинсом (Steven and Collins) [26], развивает основные идеи *SCHOLAR* в направлении моделирования предметной области и стратегии обучения. В *WHY* впервые была сделана попытка компьютерной реализации “некомпьютерной” стратегии обучения, известной как метод Сократа.

Метод Сократа является сугубо диалоговым и предполагает такой способ ведения диалога, при котором пользователь, «ведомый» вопросами тьютора, ставится в положение, когда он должен при помощи дедуктивных умозаключений сам выводить новые, не известные ему ранее, каузальные отношения между известными фактами предметной области. Метод Сократа, таким образом, подчиняется некоторым общим правилам, определяющим диалоговое поведение тьютора. Коллинс и его коллеги предложили более полусотни правил, сформулированных в виде условных предложений и определяющих поведение тьютора, использующего метод Сократа применительно к предметной области *WHY* (метеорология) [27, 28]. Каждое правило связывает последний ответ пользователя и последующий обучающий стимул и вопрос тьютора. На рис. 1.9 приведено одно из правил, определяющих диалоговое поведение *WHY*.

<p>ПРАВИЛО 6 <u>Общее правило для несущественного фактора.</u></p> <p>ЕСЛИ пользователь даёт объяснение одному или нескольким несущественным факторам,</p> <p>ТО сформулируйте общее правило, утверждающее, что данный фактор является существенным</p> <p>И спросите пользователя, является ли это правило истинным.</p> <p>Основание для использования Научить пользователя обращать внимание на остальные каузальные факторы.</p> <p>Пример использования Если пользователь называет воду причиной выращивания риса в Китае, спросите его: «Считаете ли вы, что в любом месте с достаточным количеством воды может произрастать рис?»»</p>
--

Рис. 1.9. Пример правила, используемого в интеллектуальном тьюторе *WHY* для управления диалогом

Правило иллюстрирует способ, при помощи которого тьютор стимулирует пользователя обнаруживать ошибки, пытаясь вовлечь его в “ловушку”, заключающуюся в том, что среди последовательности сделанных им ранее заключений имеются противоречивые.

На рис. 1.10 приведен пример сократовского диалога программы *WHY*, иллюстрирующий применимость правила, приведенного на рис. 1.9 (четвёртый вопрос).

Как видно из приведенного фрагмента диалога, *WHY* поддерживает диалог, в котором роли партнеров зафиксированы. Тьютор выступает в роли спрашивающего партнера, а пользователь — в роли отвечающего.

Знания о предметной области в программе *WHY* хранятся в виде иерархически упорядоченной системы сценариев. Общее определение сценария, предложенное Коллинсом и его коллегами, определяет его как набор каузальных и темпоральных отношений между событиями, под которыми естественно понимать вопросы тьютора и ответы пользователя.

Таким образом, процессы генерации вопросов и оценки ответов в *WHY* задаются множеством сценариев, упорядоченных иерархически. Сценарии наиболее высокого уровня хранят знания в наиболее общей форме, а их детализация осуществляется сценариями более низкого уровня.

Предполагается, что обучение начинается со сценария, находящегося на наиболее высоком уровне. Сценарии *WHY* хранят знания о предметной области и стратегию обучения, но не используются для хранения стратегии диагностики пользователя. Поэтому *WHY* обладает ограниченными диагностическими

возможностями, сводимыми к независимому анализу текущих ответов пользователя. Последующие исследования, направленные на дальнейшее приближение *WHY* к идеальной тьюторской системе, проводились по двум направлениям: моделирование целенаправленного глобального обучающего поведения и расширение концепции сценарной модели предметной области.

WHY: КАК ВЫ ДУМАЕТЕ, В АМАЗОНСКИХ ДЖУНГЛЯХ СИЛЬНЫЕ ИЛИ СЛАБЫЕ ДОЖДИ?

(Rule 1: Ask for a prediction about a particular case.)

Студент: Сильные дожди

WHY: ПОЧЕМУ В АМАЗОНИИ СИЛЬНЫЕ ДОЖДИ?

(Rule 2: Ask for prior cause.)

Студент: Потому что она рядом с Атлантикой

WHY: ДА, АМАЗОНСКИЕ ДЖУНГЛИ РЯДОМ С ЭКВАТОРИАЛЬНОЙ АТЛАНТИКОЙ. КАК ЭТО ВЛИЯЕТ НА ДОЖДИ В АМАЗОНСКИХ ДЖУНГЛЯХ?

(Rule 3: Ask for intermediate cause.)

Студент: Вода в течении тёплая

.....

WHY: ДУМАЕТЕ ЛИ ВЫ, ЧТО В ЛЮБОМ МЕСТЕ С ГОРАМИ ИДУТ СИЛЬНЫЕ ДОЖДИ?

(Rule 6: Form a general rule for an insufficient cause.)

Студент: Да

WHY: В ЮЖНОЙ КАЛИФОРНИИ ЕСТЬ ГОРЫ. ПОЧЕМУ В ЮЖНОЙ КАЛИФОРНИИ НЕ БЫВАЕТ СИЛЬНЫХ ДОЖДЕЙ?

(Rule 7: Pick a counter-example for an insufficient cause, and Rule 2 Ask for prior causes.)

Рис. 1.10. Пример диалога программы *WHY*, который иллюстрирует применимость правила, приведенного на рис. 1.9 (четвёртый вопрос)

Моделирование целенаправленного глобального обучающего поведения *WHY* базировалось на изучении поведения опытного тьютора, применяющего метод Сократа. Анализ полученного фактического материала позволили Коллинсу и Стивену выявить две глобальные цели, определяющие два типа диалоговых эпизодов в процессе обучения: диагностика и коррекция [29].

Диагностика определяется как процесс, позволяющий проследить связь между множеством «поверхностных ошибок» и глубоким непониманием

материала. Возможность коррекции базируется на способности вначале классифицировать ошибки, а затем выбирать релевантную корректирующую стратегию. Коллинс и Стивен выявили пять типов «концептуальных ошибок», используемых тьютором-человеком для корректировки обучающей стратегии: от простой ошибки в факторе, которая может быть скорректирована простым утверждением, до ошибочного способа ведения дедуктивных умозаключений, который может быть скорректирован путём дополнительного обучения. Следует отметить, что отмеченные свойства целенаправленного обучающего поведения человека-тьютора были представлены Коллинсом и Стивеном в виде теории, но не были воплощены в очередную версию *WHY*.

Расширение концепции сценарной модели предметной области базировалось на критике сценарно-ориентированного способа моделирования предметной области. По мнению авторов, сценарий в большей степени пригоден для управления диалоговым процессом и в меньшей степени для описания ментальной модели изучаемой предметной области. Ментальная модель определяется как внутренняя, доступная сознанию, модель реальной физической системы, используемая индивидом для построения логических умозаключений и предсказания поведения реальной физической системы. Ментальная модель одной и той же физической системы имеет вертикальную структуру. Стивен и Коллинс обнаружили, например, четыре типа ментальных моделей метеорологических процессов [30].

- На планетарном уровне обычно используется климатическая модель, которая описывает глобальные водные и воздушные течения и их влияние на климат.
- С климатической моделью связана модель циркуляции воды, которая описывает абсорбцию воды воздушными массами, её транспортировку и возвращение на поверхность в виде осадков.
- Модель циркуляции воды связана с макрофункциональной моделью парообразования такими факторами, как температура и влажность.
- Наконец, существует микромодель парообразования, которая учитывает столкновение молекул воды и их испарение с поверхности.

Множественность ментальных моделей предметной области изменяет точку зрения на понимание сложной системы и обучение сложной системе. Понимание сложной системы в большой степени зависит от способности использовать различные ментальные модели и переходить от одной модели к другой. Для каждой из моделей Стивен и Коллинз предложили шкалу, отражающую степень её усвоения и определили обучение как процесс последовательных приближений ментальной модели обучаемого к модели реальной физической системы. Из такого понимания процесса обучения следует стратегия обучения, основанная на постоянном сопоставлении модельных предсказаний и фактов реальной физической системы, что является основной техникой метода Сократа.

Несоответствие, обнаруженное в процессе диагностики, должно служить причиной корректировки ментальной модели пользователя, а поскольку между ментальными моделями существует иерархическая упорядоченность, обучение может рассматриваться также как процесс диагностики и коррекции ментальных моделей пользователя на всё более глубоких уровнях.

Интеллектуальные тьюторы второго поколения, которые разрабатываются, начиная с конца 90-х годов прошлого века и по настоящее время, обладают следующими отличительными чертами: главенствующая роль диалога в процессе передачи знаний; использование гибкого естественно языкового диалога; использование диалога со смешанными ролями; интеграция с технологией World Wide Web.

В [31–33] описан проект интеллектуальной тьюторской системы CIRCSIM-TUTOR, предназначенной для приобретения навыков решения задач в области сердечно-сосудистых заболеваний и регулирования артериального давления. Система способна поддерживать вопросно-ответный диалог на естественном языке. Из опубликованных материалов следует, что в процессе диалога спрашивающей стороной является, как правило, тьютор.

Диалоговый тьютор *SCoT* (*Spoken Conversational Tutor*), проект которого предложен в Центре по изучению языка и информации Станфордского университета, является также представителем второго поколения интеллектуальных тьюторов [34, 35]. Архитектура и поведение *SCoT* отличаются тем, что в них сделан акцент на разделение проблемно-независимых и проблемно-ориентированных функций системы. В тьюторе используется типичный для современных интеллектуальных тьюторов архитектурный элемент — диалоговый менеджер общего назначения, способный поддерживать естественно-языковой диалог со смешанными ролями [36, 37]. Идея диалогового менеджера развивается и в других проектах, например, таких, как *CycleTalk* [38], *RMT* (*Research Methods Tutor*) [39].

Одним из перспективных направлений развития интеллектуальных тьюторов второго поколения является адаптация их архитектуры и поведения к технологии World Wide Web. В [40] описан интеллектуальный тьютор *CALAT*, на примере которого предлагается возможная архитектура и распределение функций между серверной и клиентской стороной Web-ориентированного интеллектуального тьютора.

1.2.2. Обучающие среды с интеллектуальной поддержкой

Серия программ, под общим наименованием *SOPHIE*, описанных в работах Брауна и его коллег [41–45], открывает принципиально новый подход к компьютерному обучению. Разработчики *SOPHIE* не ставили перед собой цель разработать «искусственный тьютор-проводник», ведущий пользователя через

материал предметной области с учетом его когнитивной индивидуальности и уже имеющихся знаний. Скорее наоборот, *SOPHIE* представляет собой «интеллектуальную среду» для обучения путём экспериментирования, в которой пользователь может «перемещаться» по своему усмотрению. Метод обучения, используемый *SOPHIE*, базируется на философии упрощенного конструктивизма и предполагает, что обучение осуществляется в процессе исследования специальной обучающей среды. Пользователь, экспериментируя со средой, проверяет свои идеи и гипотезы и получает в ответ критические замечания и советы. Предметной областью *SOPHIE* является поиск дефектов в электронных схемах. В работах Брауна и его коллег описано три версии *SOPHIE*. Отличительной особенностью всех версий является совершенный естественно-языковой интерфейс, базирующийся на концепции семантического разбора предложений, предложенного Бартоном [46]. На рис. 1.11 приведен пример фрагмента диалога, иллюстрирующего возможности *SOPHIE* интерпретировать «некорректные» естественно-языковые вопросы.

Student: Now what is the outpt?

SOPHIE: (OUTPT \Rightarrow OUTPUT) THE OUTPUT VOLTAGE IS 1.2 VOLTS. IF YOU ARE NOT SURE WHETHER A VALUE IS CORRECT, CAN ASK "IS THAT VALUE CORRECT?"

После того, как ошибка в слове «outpt» распознана и имеющаяся неопределенность разрешена как «output voltage», SOPHIE информирует студента об одной из своих возможностей.

Student: Is that right?

SOPHIE: IN A WORKING CIRCUIT THE OUTPUT VOLTAGE IS 30.1 VOLTS. THE VALUE IS INCORRECT.

Заключение об анафоре: слово «that» относится к величине предыдущего измерения.

Student: What is the voltage between the current limiting transistor and the constant current source?

SOPHIE: THE VOLTAGE AT №4 IS 2.48 VOLTS.

Слово «between» интерпретируется как ссылка на определенную точку, в которой измеряется напряжение по отношению к земле.

Student: What is it between there and the base of Q6?

SOPHIE: THE VOLTAGE BETWEEN №4 AND №5 IS 1.28 VOLTS.

Слово «it» относится к напряжению, а слово «there» — к №4. Слово «between» в этом случае интерпретируется не так, как в предыдущем вопросе пользователя.

Рис. 1.11. Фрагмент естественно-языкового диалога программы *SOPHIE*

Программа *SOPHIE I*, следуя философии упрощенного конструктивизма, отличается тем, что не проявляет никакой педагогической инициативы и поэтому не хранит ни модели пользователя, ни какой-либо стратегии обучения. Единственное, что делает система в смысле методики обучения, — это фиксирует дефект и предлагает пользователю его обнаружить. Следствием чего является принятое распределение ролей партнеров диалога. Как видно из приведенного на рис. 1.11 фрагмента, распределение ролей зафиксировано и представляет собой полную противоположность *WHY*. Обучаемый задаёт вопросы, а система отвечает на них. Таким образом, цель встроенного в обучающую среду тьютора — формировать осмысленные ответы на текущие вопросы пользователя, рассматривая их как независимые.

Программа *SOPHIE I* отличается от ранее рассмотренных систем коммуникации знаний тем, что использует не одну, а несколько моделей предметной области: (1) симулятор в виде математической модели электронной схемы, (2) процедурную модель в виде набора «интеллектуальных специалистов», использующих модель в ходе логического вывода, и (3) декларативную модель в виде семантической сети фактов. Процедурная модель предназначена для ответов на вопросы пользователя путем проведения соответствующих экспериментов с симулятором. «Специалисты» процедурной модели решают четыре типа задач логического вывода.

Задача первого типа — формирование ответа на гипотетические вопросы пользователя вида: «Что произойдет с таким-то компонентом, если будет произведено такое-то действие?»

Задача второго типа — оценка гипотезы, предлагаемой пользователем.

Задача третьего типа — перечисление всех возможных гипотез, соответствующих проведенным измерениям по соответствующему запросу пользователя.

Задача четвертого типа (считающаяся наиболее сложной) заключается в оценке новых измерений, предлагаемых пользователю. На рис. 1.12 приведен фрагмент диалога *SOPHIE I*, иллюстрирующий способность системы решать перечисленные четыре типа задач логического вывода.

Программа *SOPHIE II* отличается от программы *SOPHIE I* тем, что, кроме «интеллектуальной среды» обучения путем экспериментирования, содержит также компонент, играющий роль «эксперта по поиску неисправностей». Эксперт по поиску неисправностей может продемонстрировать пользователю свою (предположительно эталонную) стратегию рассуждений. Прежде чем привести цепь рассуждений, эксперт может задать пользователю уточняющие вопросы. Таким образом, диалог *SOPHIE II* похож на диалог *SCHOLAR* в том смысле, что роли партнеров не зафиксированы и, следовательно, в процессе обучения вопросы может задавать как система, так и пользователь. Эксперт демонстрирует свой вариант логических умозаключений вне зависимости от того, как реально развивается процесс поиска неисправности.

- Студент: Что может быть неисправно?
- SOPHIE: ВОТ НЕСКОЛЬКО НЕИСПРАВНОСТЕЙ, КОТОРЫЕ ОБЪЯСНЯЮТ НАБЛЮДАЕМЫЕ ИЗМЕРЕНИЯ:
НИЗКИЙ БЭТА У Q3
НИЗКИЙ БЭТА У Q4
ИЛИ R22 ИМЕЕТ НЕВЕРНОЕ СОПРОТИВЛЕНИЕ
- Студент: Какой был бы выходной вольтаж, если бэта Q3 равен 8?
- SOPHIE: ЕСЛИ БЫ БЭТА Q3 БЫЛ РАВЕН 8, ТО ПРИ ДАННЫХ УСТАНОВКАХ ВЫХОДНОЙ ВОЛЬТАЖ БЫЛ БЫ 10.5 ВОЛЬТ. В РАБОТАЮЩЕЙ СХЕМЕ ОН РАВЕН 19.9 ВОЛЬТ.
- Студент: Правильный ли ток эмиттера у Q4?
- SOPHIE: ТОК ЭМИТТЕРА Q4 РАВЕН 61 МИЛЛИАМПЕРУ. В РАБОТАЮЩЕЙ СХЕМЕ НА ЭМИТТЕРЕ Q4–49 МИЛЛИАМПЕР. ЭТО ЗНАЧЕНИЕ, ПОХОЖЕ, НЕПРАВИЛЬНОЕ.
- Студент: Может быть, R22 низкое?
- SOPHIE: ПОДОЙДЁТ, ЕСЛИ Я ИСПОЛЬЗУЮ 25 ОМ ДЛЯ СОПРОТИВЛЕНИЯ R22?
- Студент: Да.
- SOPHIE: ЭТА НЕИСПРАВНОСТЬ СОГЛАСУЕТСЯ СО ВСЕМИ ИЗМЕРЕНИЯМИ, КОТОРЫЕ ВЫ СДЕЛАЛИ.
-

Рис. 1.12. Фрагмент диалога *SOPHIE I*, иллюстрирующий способность программы решать четыре типа задач логического вывода

Таким образом, эксперт, встроенный в программу *SOPHIE II*, не в состоянии «отслеживать логику обучаемого» в индивидуальном процессе поиска неисправностей. Задача преодоления этого недостатка была поставлена при разработке версии *SOPHIE III*, однако из имеющихся публикаций неясно, была ли эта цель достигнута.

1.2.3. Интеллектуальные компьютерные репетиторы

Серия программ под общим наименованием *WUSOR* является представителем отдельного класса систем коммуникации знаний — интеллектуальных компьютерных репетиторов.

Главная цель, которая ставится перед интеллектуальным компьютерным репетитором, заключается в поддержании обучающей активности пользователя при выполнении задания путем трансформации совершаемых им ошибок в процедурный опыт. Репетитор не вмешивается в процесс решения задачи до тех пор, пока не обнаруживает ошибочные действия. Таким образом, репетитор

осуществляет непрерывное и незаметное отслеживание активности пользователя и прерывает его работу только тогда, когда действия пользователя не совпадают с действиями, ожидаемыми системой. Искусство репетиторства заключается в нахождении оптимального момента прерывания активности пользователя, а также в нахождении корректной формулировки подсказки. Опубликован ряд педагогических принципов компьютерного репетиторства, предложенных Бартоном и Брауном [47].

Программы *WUSOR* являются репетиторами компьютерной игры *WUMPUS*, известной тем, что она часто используется как предметная область при исследовании и изучении процесса логических умозаключений в прикладных системах искусственного интеллекта [48,49]. В процессе игры игрок проходит через последовательность пещер в лабиринте. Игрока поджидают различные опасности: чудовище *WUMPUS*, бездонные ямы и летучие мыши, которые подхватывают игрока и роняют его в пещеру случайным образом. Как только игрок попадает в новую пещеру, он получает список близлежащих пещер и набор признаков, позволяющих сделать вывод о возможной опасности: сквозняк или писк, предупреждающие о яме или летучей мыши в одной из смежных пещер; плохой запах, говорящий о том, что на расстоянии не более чем в две пещеры находится *WUMPUS*. Игрок перемещается путем выбора одной из смежных пещер. Для победы в игре необходимо выстрелить одной из пяти стрел в логово *WUMPUSa*. Игра проиграна, если игрок проваливается в яму, или попадает в логово с *WUMPUSом*, или безрезультатно израсходует все стрелы. Решение о том, какая из окружающих пещер является безопасной и, следовательно, может быть выбрана следующей, требует от игрока проведения логических умозаключений.

Первая версия программы *WUSOR-I* включает два модуля, названные «эксперт» и «советчик» [50]. Модуль «эксперт» содержит описание предметной области в виде набора правил продукции, связывающих признаки и опасности и позволяющих получить прогноз об опасности или безопасности смежных пещер. Модуль «советчик» вмешивается в игру всякий раз, когда пользователь делает неоптимальный выбор следующей пещеры, и предоставляет пользователю набор соответствующих объяснений. *WUSOR-I* не диагностирует состояние знаний пользователя и не адаптирует свое вмешательство соответственно.

Главное отличие программы *WUSOR-II* от программы *WUSOR-I* — это наличие модели пользователя, базирующейся на теории оверлея (overlay theory) Кара и Гольдштейна [51]. Оверлей Кара и Гольдштейна является классической парадигмой и часто используется при моделировании пользователя (студента) в интеллектуальных тьюторских системах. Теория оверлея предполагает оценку знаний пользователя путём сравнения его поведения с поведением эксперта. При этом принимается, что знания пользователя являются подмножеством знаний эксперта. Существует ряд трудностей при применении оверлейной модели.

Например, оверлейная модель неприменима, если предметная область допускает множество парадигм решения задачи и, следовательно, может быть представлена множеством экспертов. На рис. 1.13 приведен фрагмент диалога пользователя с *WUSOR-II*.

Программа *WUSOR-III* отличается от программы *WUSOR-II* ещё большим акцентом на представление знаний, ориентированных на конкретного пользователя. При проектировании *WUSOR-III* была предложена концепция генетического графа (genetic graph) [52]. Генетический граф является графическим представлением эволюционных отношений между фрагментами знаний и базируется на эволюционной теории интеллекта Пиаже [53]. Узлы генетического графа соответствуют элементарным навыкам, представленным индивидуальным правилом, а ветви — их эволюционным отношениям, таким как обобщение или аналогия. В *WUSOR-III* модель пользователя представлена оверлеем узлов генетического графа, включающих правильные и неправильные правила, а индивидуальная история пользователя представлена оверлеем ветвей генетического графа.

WUSOR: ПРИВЕТ, МЭРИ. ИМЕЕТСЯ 15 ПЕЩЕР, 2 ЛЕТУЧИЕ МЫШИ, 2 ЯМЫ И 1 ВАМПУС. СЕЙЧАС ТЫ В ПЕЩЕРЕ 15 И СОСЕДНИЕ ПЕЩЕРЫ — 4, 14, И 0. БРРР! ТУТ СКВОЗНЯК. ТЫ РЯДОМ С ЯМОЙ. ЧТО ЗА ВОНЬ! ГДЕ-ТО ПОБЛИЗОСТИ ВАМПУС! КУДА ТЕПЕРЬ?

Студент: 4

WUSOR: СЕЙЧАС ТЫ В ПЕЩЕРЕ 4 С СОСЕДЯМИ 15, 14 И 2 БРРР! ТУТ СКВОЗНЯК. ПИСК! ЛЕТУЧАЯ МЫШЬ ГДЕ-ТО РЯДОМ. КУДА ТЕПЕРЬ?

Студент: 14

WUSOR: МЭРИ, НЕТ НЕОБХОДИМОСТИ ТАК СИЛЬНО РИСКОВАТЬ С ЯМАМИ. ОДНА ИЗ ПЕЩЕР, 2 ИЛИ 14, СОДЕРЖИТ ЯМУ. СУЩЕСТВУЕТ МНОЖЕСТВО ПРИЗНАКОВ, ЧТО ЯМА В ПЕЩЕРЕ 14, ЧТО ДЕЛАЕТ ВЕСЬМА ВЕРОЯТНЫМ НАЛИЧИЕ ЯМЫ В ПЕЩЕРЕ 14. МЕНЕЕ ВЕРОЯТНО, ЧТО ЯМА В ПЕЩЕРЕ 0. ТАКИМ ОБРАЗОМ, МЭРИ, МОЖЕТ БЫТЬ, МЫ ИССЛЕДУЕМ ПЕЩЕРУ 0? НЕ ХОЧЕШЬ ЛИ ТЫ ОТКАЗАТЬСЯ ОТ СВОЕГО ХОДА?

Рис. 1.13. Фрагмент диалога с программой *WUSOR-II*

С точки зрения проблематики диалогового поведения, генетический граф является одним из немногих подходов, который может рассматриваться как унифицированная сетевая (сценарная) модель, включающая как стратегию обучения, так и стратегию диагностики и предоставляющая средства персонализированного подхода к обучению.

Серия программ под общим наименованием «когнитивный тьютор», разработанных в университете Карнеги Мелон (США), представляет собой один из наиболее известных примеров интеллектуальных компьютерных репетиторов. Главным идеологом этих программ является Джон Андерсон, разработавший несколько версий унифицированной теории познания, известной под наименованием *ACT (Adaptive Control of Thought)* [54–56]. Когнитивные тьюторы первоначально разрабатывались с целью проверки ключевых идей *ACT*, главной из которых является убеждение в том, что процедурные знания человек хранит в виде системы продукционных правил, упорядоченных иерархией целей. Когнитивные тьюторы разработаны для обучения в нескольких предметных областях: программирование на алгоритмических языках (LISP, Pascal), элементарная математика (геометрия, алгебра) [57–62] и др.

Стратегия репетиторства, воплощенная в когнитивных тьюторах, называется «трассировка модели». Когнитивный тьютор хранит эталон процедурных знаний, которые должен освоить пользователь, в виде последовательности шагов-действий. В том случае, если фактическая последовательность шагов-действий пользователя совпадает с эталонной, то когнитивный тьютор никак не вмешивается в действия пользователя. Если действия пользователя отклоняются от эталонных, то тьютор реагирует выводом сообщения, содержащего расшифровку ошибки, и рекомендации относительно дальнейших действиях. Как при описании эталона, так и при описании возможных отклонений используется система продукционных правил, упорядоченных иерархией целей.

На рис. 1.14 приведен пример продукционного правила, используемого в LISP тьюторе.

IF	цель — кодирование функции с табличным входным параметром, и функция должна иметь доступ к каждому элементу таблицы, и структура таблицы произвольно сложная
THEN	кодируйте cond для обеспечения car-cdr рекурсии и устанавливайте подцелью кодирование случаев остановки и кодируйте случаи рекурсии.

Рис. 1.14. Пример продукционного правила, используемого в LISP тьюторе

Одной из особенностей когнитивных тьюторов как систем коммуникации знаний является отсутствие явного вербального диалога с пользователем. Интерфейс программы разделяет экран на две части, одна из которых является полем действий пользователя, а другая — полем для вывода сообщений тьютора. На рис. 1.15 приведено несколько начальных шагов решения задачи по

программированию функции при помощи языка программирования LISP «под надзором» LISP тьютора [41].

В работах последних лет концепция когнитивных тьюторов развивается, главным образом, в направлении использования ACT-R для моделирования более тонких, направленных на передачу процедурных знаний элементов поведения студента при решении задачи [63, 64].

Окно тьютора
<p>Определите функцию с именем <code>pal</code> и списком параметров, которая возвращает палиндром. Палиндром — это список, который одинаково читается вперёд и назад. Например, <code>(pal '(a b c))</code> возвращает <code>(a b c c b a)</code>.</p>
Окно с кодом
<pre>(defun <name> <parameters> <body>)</pre>
Окно тьютора
<p>Помните, что вы пытаетесь получить параметр <code>list</code> вашей функции. Вам не следует вызывать функцию <code>list</code>. Если вы думаете использовать <code>list</code> как параметр, то это плохая идея, потому что вы можете спутать имя функции с именем параметра.</p>
Окно с кодом
<pre>(defun pal (list) <process></pre>
Окно тьютора
<p>Да, вам понадобится использовать функцию <code>REVERSE</code>, но не сейчас. Помните, что вам необходимо создать новый список, который объединяет исходный список и перевёрнутый список. Поэтому вначале вам необходима функция объединения списков.</p>
Окно с кодом
<pre>(defun pal (origlist) (reverse)</pre>

Рис. 1.15. Начальные шаги работы с LISP тьютором, иллюстрирующие специфику его интерфейса

Несмотря на различие в философии передачи знаний в интеллектуальных компьютерных тьюторах, обучающих средах с интеллектуальной поддержкой и интеллектуальных компьютерных репетиторах, их объединяет общая идея передачи знаний. Все перечисленные системы характеризуются следующими четырьмя базовыми компонентами:

- экспертные знания о предметной области (модель предметной области);
- модель пользователя;
- экспертные знания о методе обучения (стратегия обучения);
- интерфейс «обучающая система — пользователь».

Модель предметной области служит двум целям. С одной стороны, она представляет собой источник предметных знаний, «порционно» предоставляемых пользователю, а с другой — желаемый стандарт знаний пользователя (или финальную цель обучения), используемый для оценки его текущих знаний.

Модель пользователя предназначена, главным образом, для того, чтобы предсказывать наблюдаемое поведение пользователя с целью его интерпретации. Идеальная модель пользователя должна делать точные предсказания поведения любого пользователя в любом контексте предметной области. Например, в виде множества возможных реакций на текущий обучающий эпизод. С моделью пользователя часто связывается понятие диагностического процесса, под которым обычно понимается адаптация модели пользователя на основании данных, полученных в процессе диалога.

Стратегия обучения определяет, какой обучающий эпизод и в какой момент времени должен быть передан пользователю. В наиболее общем случае стратегия обучения не кодируется жестко, а выводится из текущей ситуации и метода, хранящегося в виде набора принципов или правил. Дидактические решения принимаются на основе предсказания реакции пользователя, генерируемой моделью пользователя, его реальной реакцией и отношениями между обучающими эпизодами, хранимыми в модели предметной области.

Стратегии обучения варьируются по отношению к степени «свободы», предоставляемой пользователю, и могут быть упорядоченно расположены между двумя полюсами: от полного контроля всех действий пользователя и выполнения «лоцманских» функций при проведении пользователя через обучающие эпизоды (тьютор), до полной свободы навигационной активности пользователя и выдачи рекомендаций по его запросу (обучающая среда).

Интерфейс в приведенных выше системах варьируется вокруг вариантов ограниченного естественного языка. Таким образом, авторы отмеченных систем, как правило, рассматривают диалог как сугубо вербальный процесс. Графические элементы включаются в интерфейс только для презентации обучающих эпизодов в обучающих средах с интеллектуальной поддержкой.

Рис. 1.16 иллюстрирует отношения между отмеченными четырьмя базовыми компонентами систем коммуникации знаний.

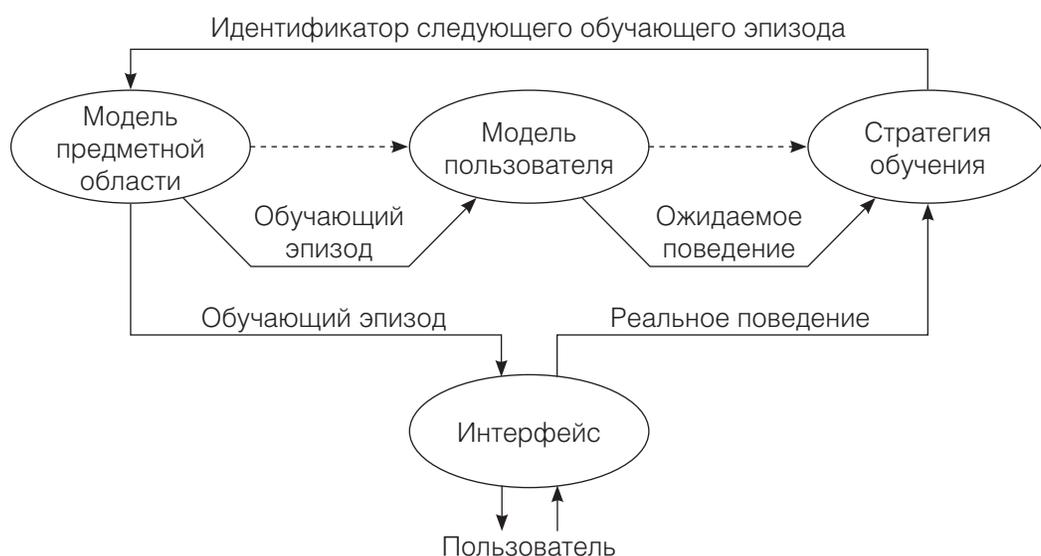


Рис. 1.16. Основные компоненты системы коммуникации знаний и отношения между ними

Внешнее, наблюдаемое поведение системы коммуникации знаний представляет собой целенаправленную последовательность транзакций, где под транзакцией понимается однократный обмен порциями информации/знаний между пользователем и системой.

Целью последовательности транзакций является максимально полная и эффективная передача знаний от системы к пользователю. В случае интеллектуального компьютерного тьютора цель определяет тьютор, в случае обучающей среды — пользователь, а в случае компьютерного репетитора цель определяется репетитором.

Отдельная транзакция может быть описана действиями пользователя и системы следующим образом:

- пользователь опосредованно, при помощи интерфейса воспринимает текущий обучающий эпизод, переданный ему системой коммуникации знаний;
- текущий обучающий эпизод передается также в модель пользователя, которая формирует его ожидаемую реакцию (ожидаемое поведение);
- на основании информации, воспринятой из обучающего эпизода, а также имеющихся знаний о предметной области, пользователь формирует реакцию (реальное поведение);
- стратегия обучения на основании ожидаемой и реальной реакций пользователя формирует идентификатор следующего обучающего эпизода, который передается в модель предметной области;
- модель предметной области формирует следующий обучающий эпизод и передает его в модель пользователя и самому пользователю.

2 ЕСТЕСТВЕННЫЙ ДИАЛоговый ПРОЦЕСС

Естественный диалог — это нерегламентированный, свободный диалог или разговор, в котором участвуют люди, а не искусственные системы. Естественный диалоговый процесс противопоставляется искусственному диалоговому процессу, примеры которого рассмотрены в первом разделе. Искусственный диалог проще естественного и регламентирован набором правил и ограничений, которые должен соблюдать пользователь. Целью настоящего раздела является рассмотрение когнитивных основ, построение онтологического базиса естественного диалогового процесса. Во второй части раздела рассмотрены модели диалоговой транзакции в контексте представления знаний.

2.1. Когнитивные основы естественного диалогового процесса

Одним из участников как искусственного, так и естественного диалогового процесса является человек. Поэтому при построении описательной или формализованной модели диалогового процесса важным является её адекватность процессам восприятия и переработки информации, которые присущи человеку. В том случае, если модель диалогового процесса построена на основе когнитивной модели, соответствующей системе восприятия человека, то от искусственных диалоговых агентов можно ожидать наследования гибкости и универсальности естественного диалога.

Среди множества моделей, предлагаемых когнитивной психологией и имеющих отношение к диалоговому процессу, целесообразно остановиться на моделях, описывающих диалог на уровне, инвариантном к процессу генерации вопросов или ответов участников диалога, поскольку генерация вопросов и ответов в значительной степени зависит от предметной области конкретного диалога, его цели и характера задачи, решаемой в процессе диалога. Иными словами, когнитивная модель диалогового процесса, полезная в методологическом и

прикладном аспектах, должна описывать некоторую «логическую машину диалога», работа которой инвариантна к предметной области.

Современная когнитивная психология, исповедующая принципы информационной переработки, выделяет следующие когнитивные системы, имеющие отношение к диалоговому процессу: сенсорная система, система восприятия/перцепции, система фокусировки внимания, система категоризации и система памяти [65, 66].

Экспериментальные данные показывают, что в восприятии участвуют два направленных навстречу друг другу ментальных процесса. Восходящий процесс имеет направление от сенсорной системы к долговременной памяти, а нисходящий процесс направлен от долговременной памяти к сенсорной системе [67].

Нейсер [68] предложил модель, интегрирующую отмеченные процессы и названную циклом восприятия. Модель предполагает, что процесс восприятия является итерационным, и описывает действия, соответствующие отдельной итерации. С точки зрения Нейсера, когнитивная структура, являющаяся одной из ключевых для процесса восприятия, — это набор схем, включающий схему, которая должна быть воспринята в текущей итерации. Этот набор схем будем называть, для краткости, гипотетическим набором схем. Здесь «схема» — это обобщенный термин, под которым в когнитивной психологии понимается какое-либо представление порции знаний.

На рис. 2.1 изображена циклическая модель восприятия Нейсера. Гипотетический набор схем совместно с перцептивным исследованием определяет информацию, которая должна быть воспринята и категоризована. Гипотетический набор схем встроен в более обширную когнитивную структуру, имеющую наименование когнитивная карта и впервые предложенную Толманом [69]. Простейшее определение когнитивной карты — это ментальное отображение фрагмента воспринимаемого окружающего мира.

Восприятие является конструктивным процессом в том смысле, что на каждой итерации конструируется новый гипотетический набор схем. Образование очередного гипотетического набора схем порождает процесс перцептивного исследования окружающей среды с целью установления степени адекватности гипотетического набора схем набору реальных стимулов.

Процесс исследования предполагает перемещение сенсорных органов, головы и всего тела. Наиболее ярким примером перцептивного исследования является ощупывание предмета при его тактильном восприятии. Процесс перцептивного исследования завершается тем, что сенсорная система фокусирует внимание на конкретном наборе стимулов окружающей среды. Сенсорная система формирует информационные массивы, которые затем воспринимаются как реальный набор схем. Воспринятый реальный набор схем является причиной модификации гипотетического набора. Как было отмечено выше, суть процесса модификации текущего гипотетического набора схем заключается в конструировании последующего гипотетического набора.

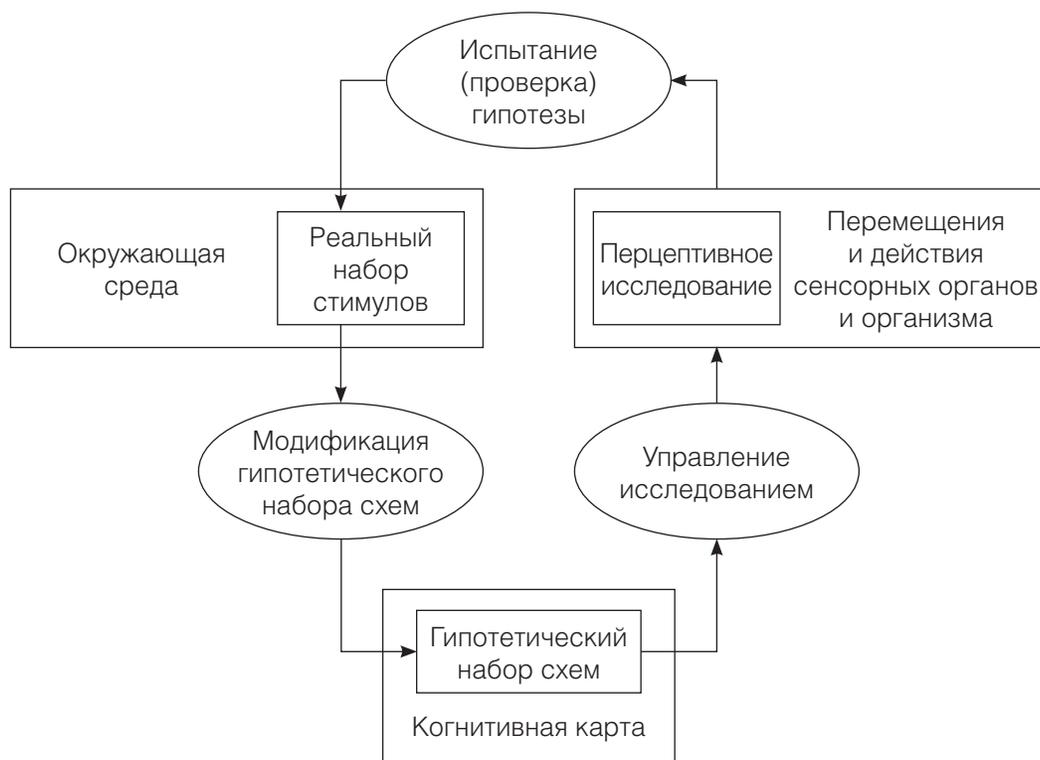


Рис. 2.1. Циклическая модель восприятия Нейсера

Отметим, что гипотетический набор схем формируется из ранее воспринятых и запомненных схем, составляющих когнитивную карту. Таким образом, цикл Нейсера предполагает, что процесс восприятия детерминирован двумя факторами: окружающей средой и прошлым опытом восприятия.

Цикл Нейсера хорошо моделирует процесс рутинного восприятия, под которым понимается процесс последовательной фокусировки сенсорной системы на внешних, не представляющих угрозу (не подозрительных) стимулах. Восприятие подозрительных стимулов, очевидно, требует модификации цикла Нейсера. Например, поскольку обработка подозрительных стимулов должна выполняться быстро, то, наверное, подозрительные стимулы должны прерывать рутинную перцепцию на любой её стадии.

По отношению к любому из участников диалога (как спрашивающему, так и отвечающему), диалоговый процесс аналогичен процессу перцептивного взаимодействия человека с окружающей средой. Отличие заключается в том, что в процессе диалога главные компоненты сенсорной системы человека — зрительный и слуховой анализаторы — подключены не к «естественной» окружающей

среде, а к «искусственной», формируемой потоками зрительных и звуковых стимулов, генерируемых противоположным агентом диалога. Таким образом, в диалоговом процессе реальная окружающая среда подменяется искусственной. Однако очевидно, что восприятие и последующая переработка как стимулов, порождаемых искусственной средой (диалоговым агентом), так и стимулов, генерируемых естественной окружающей средой, осуществляется по одним и тем же «правилам и законам» и одной и той же цепью подсистем переработки информации человеком.

Отмеченная аналогия цикла Нейсера и диалогового процесса, а также независимость модели Нейсера от предметной области восприятия послужили причиной выбора циклической модели восприятия Нейсера в качестве основы для построения когнитивного цикла диалогового процесса [70–92].

Когнитивный цикл диалогового процесса, приведенный на рис. 2.2, моделирует, главным образом, поведение активного (спрашивающего) диалогового агента с фиксированной ролью.

Модель предполагает, что диалоговое поведение является итерационным и на каждой итерации активный диалоговый агент выполняет когнитивные действия, приведенные на рис. 2.2.

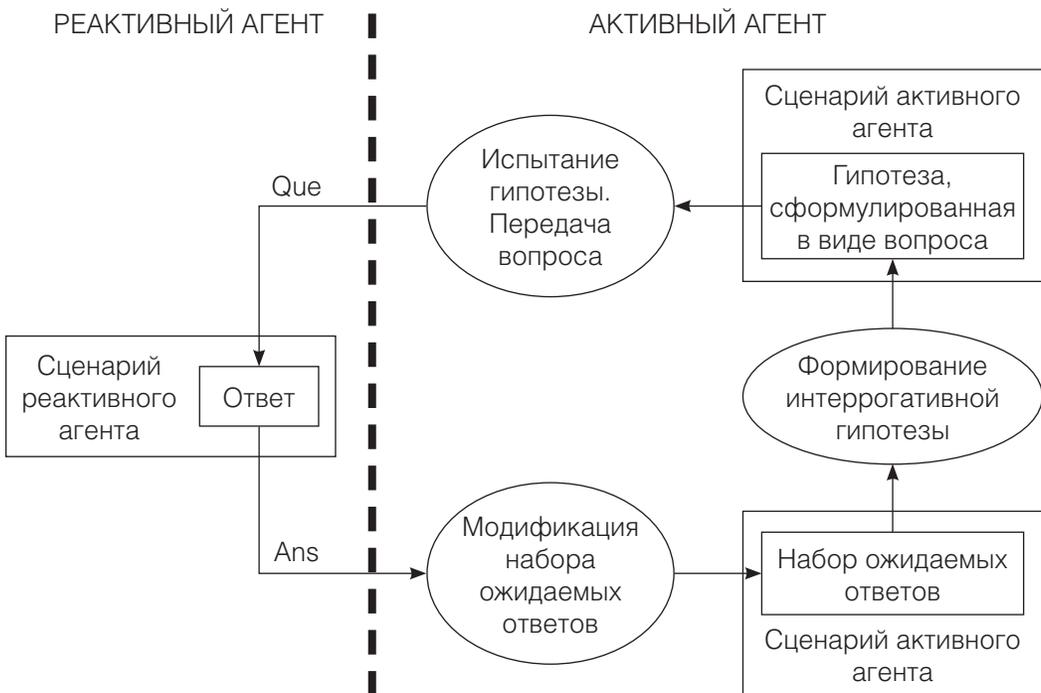


Рис. 2.2. Когнитивный цикл диалогового процесса

Будем считать, что итерация начинается с момента, когда активный агент оперирует относительно небольшим набором ответов, названным набором ожидаемых ответов. Предполагается, что реальный ответ, генерируемый реактивным агентом, совпадает с одним из ответов, входящих в набор ожидаемых ответов. Набор ожидаемых ответов является частным случаем гипотетического набора схем цикла Нейсера (см. рис. 2.1). Таким образом, одной из основных когнитивных структур, которой оперирует активный агент, является схема-ответ, или ментальное представление структуры ответа.

Набор ожидаемых ответов встроен в более обширную когнитивную структуру — сценарий активного агента, являющуюся аналогом когнитивной карты Толмана. Неявно предполагается, что сценарий активного агента включает все необходимые для данного диалога схемы-ответы и что набор ожидаемых ответов — это некоторое подмножество ответов из сценария, релевантное данной диалоговой транзакции. Мы не интересуемся происхождением сценария активного агента и считаем, что он воплощает его опыт диалогового общения.

После восприятия и распознавания реального ответа, полученного от реактивного агента, осуществляется модификация набора ожидаемых ответов, смысл которой — подготовка следующей итерации. Формируется новый набор ожидаемых ответов, включающий ответ, ожидаемый на следующей итерации. В процессе модификации текущий набор ожидаемых ответов может быть заменён полностью или частично.

Восприятие и распознавание реального ответа порождает два когнитивных процесса. Один из процессов — это описанный только что процесс модификации набора ожидаемых схем, а другой — процесс генерации очередного вопроса. Ясно, что ответ детерминируется вопросом, и то, какой реальный ответ будет получен от реактивного агента в последующей итерации, определяется вопросом, сформированным активным агентом в данной итерации.

Ранее неоднократно отмечалось, что одним из отличительных свойств диалогового процесса является его целенаправленность. За внешним процессом обмена вопросами и ответами всегда скрывается процесс достижения цели. Для активного агента под целью можно понимать или получение от реактивного агента некоторого факта в виде ожидаемого целевого ответа (примером может служить диалог — допрос подозреваемого), или реализация диалогового процесса по некоторой целевой «траектории» (примером может служить диалог студента и тьютора). Целенаправленный характер диалога подразумевает, что вопросы активного агента формируются не произвольным образом, а целенаправленно. Управлять этим процессом активный агент может только путём построения интеррогативной гипотезы (гипотезы, имеющей структуру вопросительного предложения), формулировки релевантного вопроса и испытания гипотезы путём передачи её реактивному агенту.

Иллюстрацией сказанного может служить диалог с программой *WHY*, приведенный на рис. 1.10. Во фрагменте диалога, приведенном ниже, присутствуют правила (Rule 1 и Rule 2), на основании которых *WHY* (активный агент) формирует интеррогативные гипотезы.

WHY: КАК ВЫ ДУМАЕТЕ, В АМАЗОНСКИХ ДЖУНГЛЯХ СИЛЬНЫЕ
ИЛИ СЛАБЫЕ ДОЖДИ?

(Rule 1: Ask for a prediction about a particular case.)

Студент: Сильные дожди

WHY: ПОЧЕМУ В АМАЗОНИИ СИЛЬНЫЕ ДОЖДИ?

(Rule 2: Ask for prior cause.)

Схема-вопрос, являющаяся ментальным представлением структуры вопроса, наряду со схемой-ответом является одной из основных когнитивных структур, которыми оперирует активный агент. Предполагается, что схемы-вопросы, необходимые для данного диалога, также встроены в сценарий активного агента.

Концепция когнитивной карты и её частного случая сценария диалога ясно указывает на то, что, по крайней мере, активный агент-человек, имеющий опыт ведения диалога, не «вычисляет» очередной вопрос с помощью некоторого супералгоритма, а, скорее, разыскивает его в памяти вопросов, используя в качестве метода доступа к памяти вопросов метод достижения цели диалога. Таким образом, метод доступа к памяти вопросов является хранилищем метода достижения цели диалога, или диалоговым методом решения задачи. В приведенном фрагменте диалога с *WHY* диалоговый метод решения задач представлен в виде двух правил (Rule 1 и Rule 2). Концепты память вопросов и диалоговый метод решения задачи входят в объем понятия «сценарий» активного агента и детализируют его структуру.

Как для естественного, так и для искусственного диалоговых процессов важным является понятие протокола диалога. Диалоги, которые можно встретить в диалогической литературе, представлены именно протоколами. С точки зрения понятия диалогового протокола, сценарий диалога можно рассматривать как структуру, способную породить множество конкретных протоколов. Таким образом, будем считать, что протокол диалога является одной из возможных реализаций диалогового процесса, а все его возможные реализации представлены сценарием.

Как интроспекция, так и анализ диалоговых процессов, описанных в литературе, обнаруживают, что имеет место не взаимно-однозначная связь между конкретным ответом реактивного агента и последующим вопросом, генерируемым активным агентом. Иными словами, в различных диалоговых итерациях на один и тот же ответ активный агент может генерировать различные вопросы. Для учета отмеченной неоднозначности предлагаются следующие три принципа.

Принцип «глубины» диалога. Принцип означает, что диалоговый сценарий при формировании вопроса должен учитывать как воспринятый ответ, так и номер диалоговой транзакции. В различных транзакциях на один и тот же ответ могут быть сформированы разные вопросы.

Принцип «истории» ответов. Принцип означает, что диалоговый сценарий при формировании вопроса учитывает как воспринятый текущий ответ, так и ранее воспринятые ответы. В одной и той же транзакции на один и тот же ответ могут быть сформированы разные вопросы в зависимости от того, какие ответы были получены в предыдущих транзакциях.

Принцип «истории» вопросов. Принцип означает, что диалоговый сценарий при формировании вопроса учитывает как воспринятый ответ, так и ранее сформированные вопросы. В одной и той же транзакции на один и тот же ответ могут быть сформированы разные вопросы в зависимости от того, какие вопросы были сформированы в предыдущих транзакциях.

Диалоговый цикл, приведенный на рис. 2.2, хорошо описывает диалог, который уместно назвать *гармоничным диалогом*. Под гармоничным диалогом понимается такой диалог, когда оба агента удовлетворены своими ролями и не хотят их изменять. Гармоничный диалог соответствует циклу Нейсера для случая рутинного восприятия. Однако гармоничный диалог не всегда возможен. Каким образом можно учесть возможность нарушения гармоничного диалога и изменения текущих ролей диалоговых агентов? Ответ на этот вопрос может быть следующим. Если считать, что инициатором смены ролей обычно выступает реактивный агент, то признаком момента смены ролей является передача активному агенту сообщения, имеющего статус вопроса. Поэтому одним из способов учёта возможности смены ролей может быть включение в список ожидаемых ответов «детектора вопроса». Детектор вопроса может быть включён в каждый набор ожидаемых ответов (тогда агенты могут сменить роли в любой транзакции) или только в некоторые наборы ожидаемых ответов (тогда роли агентов могут быть изменены только в некоторых, заранее определенных транзакциях).

2.2. Онтологический базис естественного диалогового процесса

При изложении предыдущего материала использовались некоторые, интуитивно понятные концепты, характеризующие диалоговый процесс, например: диалоговый агент, активный диалоговый агент, реактивный диалоговый агент, сценарий диалога, протокол диалога и др. Однако, для изложения последующего материала нам понадобится более обширный и более определенно сформулированный онтологический базис естественного диалогового процесса.

Построение онтологического базиса естественного диалога предполагает анализ некоторого количества протоколов естественных диалоговых процессов, отражающих их логическое и операционное многообразие. Существует множество источников таких протоколов. Например, протоколы допросов подозреваемых. Однако, более богатым и всем доступным источником протоколов естественных диалогов является диалогическая литература. Одним из наиболее значимых источников протоколов естественных диалогов такого рода являются диалоги Платона [93]. Наверное, никто не усомнится в совершенстве искусственного диалогового агента, способного вести такие же диалоги, на которые был способен Сократ.

Концепты, формирующие онтологический базис естественного диалога, будем иллюстрировать на примере небольшого фрагмента одного из диалогов Платона, названного Протагор. Фрагмент диалога Протагор, приведенный ниже, представлен в форме, немного отличающейся от оригинала. Незначительные изменения и добавления сделаны с целью его структуризации, для облегчения ссылок на элементы этого текста. Так, например, отмечены начало и конец каждой транзакции, а также явно отмечены субъекты диалогового общения. Диалог происходит между Сократом (**С**) и Гиппократом (**Г**) перед их встречей с Протагором.

----- Начало транзакции 1. **С** — активный агент -----

- С.** Так мы идём платить деньги Протагору, потому что он софист?
- Г.** Конечно.

----- Конец транзакции 1. Начало транзакции 2 -----

- С.** А если бы спросили тебя ещё и вот о чём: «Сам-то ты кем намерен стать, раз идёшь к Протагору?»
- Г.** Если сообразоваться с прежде сказанным, то ясно, что я собираюсь стать софистом.

----- Конец транзакции 2. Начало транзакции 3 -----

- С.** А тебе не стыдно было бы, клянусь богами, появиться среди эллинов в виде софиста?
- Г.** Клянусь Зевсом, стыдно, Сократ, если говорить то, что я думаю.

----- Конец транзакции 3. Начало транзакции 4 -----

- С.** Но пожалуй, Гиппократ, ты полагаешь, что у Протагора тебе придётся учиться иначе, подобно тому как учился ты у учителя грамоты, игры на кифаре или гимнастики? Ведь каждому из этих предметов ты учился не как будущему своему мастерству, а лишь ради своего образования, как это подобает частному лицу и свободному человеку.
- Г.** Конечно, мне кажется, что Протагорово обучение скорее такого рода.

----- Конец транзакции 4. Начало транзакции 5 -----

- С.** Так сам-то ты знаешь, что собираешься делать, или тебе это неясно?
- Г.** О чём это ты?

----- Конец транзакция 5. Начало транзакции 6. Обмен ролями.

Далее **С** — реактивный агент -----

- С.** Ты намерен предоставить попечение о твоей душе софисту, как ты говоришь; но, право, я бы очень удивился, если бы ты знал, что такое софист. А раз тебе это неизвестно, то ты не знаешь и того, кому ты вверяешь свою душу и для чего — для хорошего или дурного.
- Г.** Я думаю, что знаю.

----- Конец транзакции 6. Начало транзакции 7. Обмен ролями.

Далее **С** — активный агент -----

- С.** Так скажи, что такое софист, по-твоему?
- Г.** Я полагаю, что по смыслу этого слова, он — знаток в мудрых вещах.

----- Конец транзакции 7. Начало транзакции 8-----

- С.** Да ведь это можно сказать и про живописцев, и про строителей: они тоже знатоки в мудрых вещах; но если бы кто-нибудь спросил у нас, в каких именно мудрых вещах знатоки живописцы, мы бы сказали, что в создании изображений; и в других случаях ответили бы так же. А вот если бы кто спросил, чем мудр софист, что бы мы ответили? В каком деле он наставник?
- Г.** А что если бы мы так определили его, Сократ: это тот, кто наставляет других в искусстве красноречия?

----- Конец транзакции 8. Начало транзакции 9. Обмен ролями.

Далее **С** — реактивный агент -----

- С.** Может быть мы и верно бы сказали, однако недостаточно, потому что этот наш ответ требует дальнейшего вопроса: если софист делает людей искусными в речах, то о чём эти речи? Кифарист, например, делает человека искусным в суждениях о том, чему он его научил, — то есть об игре на кифаре. Не так ли?
- Г.** Да.

----- Конец транзакции 9. Начало транзакции 10. Обмен ролями.

Далее **С** — активный агент -----

- С.** Допустим. Ну а софист, в каких речах он делает искусным? Не ясно ли, что в речах о том, в чём он и сам сведущ?
- Г.** Похоже на то.

----- Конец транзакции 10. Начало транзакции 11 -----

- С.** А в чём же софист и сам сведущ, и ученика делает сведущим?
- Г.** Клянусь Зевсом, не знаю, что тебе ответить.

----- Конец транзакции 11. Начало транзакции 12 -----

- С.** Как же так? Знаешь, какой опасности ты собираешься повергнуть свою душу? Ведь когда тебе бывало нужно вверить кому-нибудь своё тело и

было неизвестно, пойдёт ли это на пользу или во вред, ты и сам немало раздумывал, верить его или не верить, и друзей и домашних призывал на совет и обсуждал это дело целыми днями. А когда речь зашла о душе, которую ты ведь ставишь выше, чем тело, потому что от того, будет она лучше или хуже, зависит, хорошо или дурно пойдут все твои дела, ты ни с отцом, ни с братом и ни с кем из нас, твоих друзей, не советовался, верить ли тебе или не верить свою душу этому пришлому чужеземцу. Лишь вчера ввечеру, по твоим словам, услышав о нём, ты уже сегодня идёшь спозаранку, не поразмыслив и не посоветовавшись о том, нужно ли верить ему себя или нет, и сразу готов потратить и собственные деньги, и деньги друзей, как будто ты уже дознался, что тебе нужно непременно сойтись с Протагором, которого, как ты говоришь, ты не знаешь и не разговаривал с ним никогда. Ты называешь его софистом, а что такое софист, оказывается, совсем не ведаешь, хоть и собираешься верить себя ему.

Г. Так оно и выходит, Сократ, как ты говоришь.

----- Конец транзакции 12. Начало транзакции 13 -----

С. А что, Гиппократ, не будет ли наш софист чем-то вроде торговца или разносчика тех припасов, которыми питается душа? По-моему, во всяком случае, он таков.

Г. Но чем же питается душа, Сократ?

----- Конец транзакции 13. Обмен ролями. Далее **С** — реактивный агент -----

2.2.1. Диалоговый агент и его роли

Участников диалогового процесса будем называть *диалоговыми агентами*. Диалоговые агенты формируют *диалоговую транзакцию*, которая представляет собой элементарный завершённый акт обмена знаниями между диалоговыми агентами. Часто, при сугубо вербальном диалоге, транзакция представляет собой вопросно-ответную пару. Хотя участников диалога может быть сколь угодно много, диалоговую транзакцию всегда формируют только два диалоговых агента. Так, например, в Платоновском диалоге Протагор в различные моменты времени транзакции формируют следующие пары агентов: Сократ-Гиппократ, Сократ-Протагор, Сократ-Каллий и др.

Следует различать *транзакцию протокола* диалогового процесса и *транзакцию сценария* диалогового процесса. В структуре транзакции протокола, например, в случае вопросно-ответной пары, имеется один вопрос и один ответ. Однако ясно, что на один вопрос в потенции имеется множество ответов, которые учтены в сценарии в виде набора ожидаемых ответов. Поэтому в транзакции в случае вопросно-ответной пары имеется один вопрос и множество ожидаемых ответов. В настоящем разделе мы будем оперировать, главным образом, понятием транзакции протокола.

Диалоговый агент в процессе диалогового взаимодействия со своим партнером может исполнять одну из двух альтернативных ролей: роль *активного диалогового агента* либо роль *реактивного диалогового агента*.

Активный агент — это спрашивающий агент. Та часть диалоговой транзакции, которую формирует активный агент по отношению к реактивному агенту, имеет статус вопроса. Главная побудительная причина диалогового агента исполнять роль активного агента заключается в нехватке знаний, необходимых для продолжения диалога и/или достижения цели. Агент исполняет роль активного диалогового агента в том случае, когда ему необходимы дополнительные знания, которые он предполагает получить от своего партнёра по диалогу.

Реактивный агент — это отвечающий агент, и та часть диалоговой транзакции, которую он формирует, имеет статус ответа по отношению к активному агенту. Диалоговый агент, исполняя роль реактивного агента, предоставляет своему партнёру некоторые знания, релевантные запрашиваемым.

В приведенном фрагменте диалога Протагор после того, как Гиппократ заявил, что он знает, кто есть софист (транзакция номер шесть), Сократу для продолжения диалога необходимы дополнительные знания, которыми обладает Гиппократ, а именно, что понимает Гиппократ под словом «софист». Поэтому в следующей транзакции (транзакция номер семь), исполняя роль активного агента, Сократ запрашивает у Гиппократа требуемые знания.

С. Так скажи, что такое софист, по-твоему?

Г. Я полагаю, что по смыслу этого слова, он — знаток в мудрых вещах.

В приведенном выше фрагменте диалога Протагор агенты (Сократ и Гиппократ) неоднократно обмениваются ролями. Вначале, в транзакциях с первой по пятую, Сократ исполняет роль активного агента, а Гиппократ — роль реактивного. Затем агенты несколько раз обмениваются ролями.

Инициатором смены ролей, как правило, является реактивный агент. Активный агент обычно удовлетворён своей ролью и после смены ролей старается восстановить свою активность при первой возможности.

2.2.2. Межличностный и внутренний диалоги

В том случае, когда транзакцию формируют две личности, между которыми осуществляется диалоговое взаимодействие, будем говорить, что имеет место *межличностный диалог*. Однако диалоговая транзакция может быть сформирована и одной личностью, попеременно исполняющей роль активного или реактивного агентов. Такой диалог назовём *внутренним диалогом*. Вербальный внутренний диалог часто называют монологом, а ментальный — размышлением.

В случае межличностного диалога, знания, которые запрашивает активный агент, хранятся в долговременной памяти реактивного агента, а знания, на которые ссылается реактивный агент в своём ответе — в долговременной памяти активного агента. Межличностный диалог возможен только в том случае, когда активный и реактивный агенты обладают одними и теми же, или примерно одними и теми же знаниями. В случае внутреннего диалога и запрашиваемые, и получаемые знания хранятся в долговременной памяти одного и того же агента.

Большая часть приведенного фрагмента диалога Протагор иллюстрирует межличностный диалог, однако в нём присутствует и внутренний диалог. В тринадцатой транзакции Сократ ведёт диалог «сам с собой», и то, что он говорит, можно представить следующим образом:

- С. (активный):** А что, Гиппократ, не будет ли наш софист чем-то вроде торговца или разносчика тех припасов, которыми питается душа?
- С. (реактивный):** По-моему, во всяком случае, он таков.

2.2.3. Транзакция диалогового протокола и ее компоненты

Диалоговый процесс может быть записан на каком-либо носителе информации в виде *протокола диалога*, представляющего собой последовательную цепь *диалоговых транзакций протокола*. Характерной особенностью естественного диалога, которая отличает его от других форм интерактивного взаимодействия, например, от работы человека с прикладной системой баз данных, является наличие сильной логической зависимости не только между компонентами отдельной транзакции, что очевидно, но и между смежными звеньями цепи транзакций. Именно наличие такой логической зависимости превращает набор отдельных диалоговых транзакций в целенаправленный интеллектуальный процесс. Каждая диалоговая транзакция протокола состоит из двух компонентов, которые будем называть *информационными посылками*.

Информационная посылка активного агента имеет статус вопроса. Слово «статус» здесь означает, что информационная посылка активного агента не обязательно является вопросом в грамматическом смысле. Это запрос на получение знаний, который может принимать различные формы. Форма информационной посылки не влияет ни на цель, которую пытается достигнуть активный агент при помощи отдельной транзакции, ни на цель, которую он пытается достигнуть при помощи всего диалогового процесса. Для активного агента важным является получение доступа к запрашиваемым знаниям, а не форма запроса, при помощи которого осуществляется этот доступ. Очевидно, что в естественном диалоге активный агент обладает свободой выбора формы

доступа к знаниям реактивного агента. Иными словами, доступ к одним тем же знаниям активный агент может получить при помощи различных информационных посылок. В ряде эпистемологических публикаций эта естественная свобода выбора активным агентом формы доступа к знаниям считается проблемой и называется «проблема конвергентных знаний». Так, например, в [94] различные вопросы, предполагающие одинаковый ответ, называются конвергентными, а знания, ассоциированные с конвергентными вопросами — конвергентными знаниями. В приведенном фрагменте диалога Протагор Сократ, исполняя роль активного агента, может различным образом формировать свои информационные послылки для доступа к одним и тем же знаниям Гиппократу. Например, транзакция номер тринадцать могла бы выглядеть следующим образом:

- С. Скажи мне Гиппократ, а не похож ли софист на продавца того, чем питается душа?
- Г. Но чем же питается душа, Сократ?

Информационные послылки активного агента могут варьироваться от простого «да/нет» вопроса, до продолжительного внутреннего диалога. Однако, даже в том случае, когда информационная посылка активного агента представляет собой некоторые его продолжительные рассуждения, она всё равно имеет статус вопроса по отношению к реактивному агенту. Поэтому уместно говорить о *степени интеррогативности информационной посылки активного агента*.

Часто информационную посылку активного агента можно понимать как *поисковое предписание*, необходимое для получения доступа к знаниям реактивного агента, или как некоторую *ссылку*, обеспечивающую доступ к области долговременной памяти реактивного агента, хранящей запрашиваемые знания.

Информационная посылка реактивного агента по отношению к активному агенту имеет статус ответа. Эту информационную посылку можно также в ряде случаев понимать как *поисковое предписание*, необходимое для получения доступа к знаниям активного агента, или как некоторую *ссылку* на участок долговременной памяти активного агента, где хранятся запрашиваемые знания. В дальнейшем, с целью сокращения длины предложений, вместо словосочетания «информационная посылка» будем использовать слово «посылка».

В приведенном фрагменте диалога Протагор одной из наиболее простых транзакций, с точки зрения структуры её посылок, является транзакция под номером один.

- С. Так мы идём платить деньги Протагору, потому что он софист?
- Г. Конечно.

В этой транзакции активный агент Сократ передаёт реактивному агенту Гиппократу посылку, обеспечивающую Сократу доступ к знаниям, которыми обладает Гиппократ, и которые отсутствуют у Сократа. Сократ не знает, намерен ли Гиппократ платить деньги Протагору только на основании того, что Протагор является софистом. Эта посылка имеет высокую степень интеррогативности и в протоколе диалога представлена одним вопросительным предложением.

В двенадцатой транзакции посылка Сократа, выступающего в роли активного агента, имеет более сложную структуру. Она представляет собой его рассуждения которым, однако, присуща некоторая степень интеррогативности.

- С.** Как же так? Знаешь, какой опасности ты собираешься повергнуть свою душу? Ведь когда тебе бывало нужно верить кому-нибудь своё тело и было неизвестно, пойдёт ли это на пользу или во вред, ты и сам немало раздумывал, верить ли ему или не верить, и друзей и домашних призывал на совет и обсуждал это дело целыми днями. А когда речь зашла о душе, которую ты ведь ставишь выше, чем тело, потому что от того, будет она лучше или хуже, зависит, хорошо или дурно пойдут все твои дела, ты ни с отцом, ни с братом и ни с кем из нас, твоих друзей, не советовался, верить ли тебе или не верить свою душу этому пришлому чужеземцу. Лишь вчера ввечеру, по твоим словам, услышав о нём, ты уже сегодня идёшь спозаранку, не поразмыслив и не посоветовавшись о том, нужно ли верить ему себя или нет, и сразу готов потратить и собственные деньги, и деньги друзей, как будто ты уже дознался, что тебе нужно непременно сойтись с Протагором, которого, как ты говоришь, ты не знаешь и не разговаривал с ним никогда. Ты называешь его софистом, а что такое софист, оказывается, совсем не ведаешь, хоть и собираешься верить себя ему.
- Г.** Так оно и выходит, Сократ, как ты говоришь.

Сократ хочет узнать, согласен ли Гиппократ с его выводом, заключающимся в том, что нельзя верить душу человеку, с которым ты не знаком и которого называют софистом, не зная, что означает это слово, поскольку душе может быть нанесён вред так же, как он может быть нанесён телу. Посылка Сократа представляет собой внутренний диалог, начинающийся с вопроса. Сократ задаёт вопрос: «Знаешь, какой опасности ты собираешься повергнуть свою душу?», затем эксплицирует его и отвечает на него. Можно предположить, что внутренний диалог Сократа имеет целью уточнить характер запрашиваемых знаний и что без этих уточнений его посылка не обеспечит ему доступ именно к тем знаниям, которые необходимы.

Из сказанного следует, что диалоговая транзакция является структурным элементом диалогового процесса, и имеет вопросно-ответную (интеррогативную) природу.

2.2.4. Типы информационных посылок активного агента

Посылки типа «поисковое предписание» и посылки типа «задача». Ранее мы интерпретировали посылку активного агента, как поисковое предписание, детерминирующее область памяти реактивного агента, в которой находятся запрашиваемые знания. Такая интерпретация предполагает, что реактивный агент уже обладает запрашиваемыми знаниями. Однако, это только один из возможных случаев. Можно сформулировать примеры посылок активного агента, запрашивающих знания, которые отсутствуют в памяти реактивного агента и требуют от него выполнения некоторой ментальной работы или ментального усилия с участием ресурса внимания с целью формирования запрашиваемых знаний. Поэтому по отношению к участию ресурса внимания реактивного агента в формировании ответа разделим посылки активного агента на два класса: *посылки типа «поисковое предписание» и посылки типа «задача».*

Для формирования знаний, запрашиваемых при помощи посылки типа «задача», реактивный агент должен решить задачу, ассоциированную с этой посылкой. Ответом является вариант решения, полученный реактивным агентом. Рассмотрим следующий пример посылки активного агента.

Сократ родился в 469 году до нашей эры. Сколько лет было бы Сократу сегодня?

Ясно, что, скорее всего, знания, запрашиваемые приведенной посылкой, не хранятся в памяти реактивного агента в готовом виде, и требуется использование ментального ресурса для их формирования.

Посылка типа «поисковое предписание» предполагает, что в момент получения посылки в памяти реактивного агента уже имеются запрашиваемые знания, а структурные элементы посылки позиционируют память на требуемый участок. Для формирования ответа на посылку типа «поисковое предписание» не требуется использование ресурса внимания. Примером может служить следующая посылка активного агента.

Как Вас зовут?

В классе посылок типа «поисковое предписание» можно выделить подклассы. Например, может случиться так, что посылка активного агента типа «поисковое предписание» будет запрашивать знания, которые, с одной стороны, отсутствуют у реактивного агента, а с другой — он не может сформировать их путем решения задачи. Такие посылки могут поставить реактивного агента в тупиковую ситуацию, для выхода из которой может понадобиться смена ролей. Например, посылка

Почему Вы скрываете, что Вы мужчина?

может поставить в тупик женщину, которая не является мужчиной. Назовем описанный подкласс посылок типа «поисковое предписание» *тупиковыми посылками типа «поисковое предписание»*.

Разделение посылок активного агента на посылки типа «поисковое предписание» и посылки типа «задача» не следует рассматривать как постоянное и абсолютное. При некоторых условиях посылка, относившаяся ранее к классу посылок типа «задача» может перейти в класс посылок типа «поисковое предписание». Если задача, порождаемая посылкой типа «задача», решается впервые или очень редко (настолько редко, что процесс забывания превалирует), то для её решения, согласно гипотезе Канемана, выделяется ресурс внимания [95]. В случае многократного и частого ответа на одни и те же или близкие посылки типа «задача» решение запоминается, для формирования ответа более не требуется механизм внимания, а посылка типа «задача» превращается в посылку типа «поисковое предписание». Если, например, реактивный агент впервые получает посылку

Сократ родился в 469 году до нашей эры. Сколько лет было бы Сократу сегодня?

то она, скорее всего, должна рассматриваться как посылка типа «задача», поскольку реактивный агент не держит в памяти знания о том, сколько лет было бы Сократу сегодня. Для получения этих знаний необходимо решить арифметическую задачу. Однако, если реактивный агент получает эту посылку часто и многократно, то он, в конце концов, запоминает результат решения (гипотетический возраст Сократа), и отмеченная посылка переходит в класс посылок типа «поисковое предписание».

Неопределенность посылок типа «поисковое предписание». Анализ естественных диалогов позволяет обнаружить, что почти всегда посылка типа «поисковое предписание» обладает некоторой степенью неопределённости по отношению к классу запрашиваемых знаний. Неопределенность выражается в том, что в посылке отсутствуют сведения о том, к какому точно классу относятся запрашиваемые знания. Чем более неопределенной является посылка активного агента, тем большей свободой обладает реактивный агент при выборе класса знаний для формирования ответа. Поэтому по отношению к точности указания класса запрашиваемых знаний все посылки типа «поисковое предписание» можно разделить, по крайней мере, на два подкласса: неопределённые и определённые. Это грубая классификация, поскольку она не учитывает степени неопределённости посылки. Более точная классификация должна учитывать степень неопределенности, характеризующую количество классов, к которым могут относиться запрашиваемые знания. Очевидно, что чем больше это количество, тем выше степень неопределенности посылки типа «поисковое предписание».

Неопределённая посылка типа «поисковое предписание» означает, что ответная посылка реактивного агента может принадлежать одному из нескольких классов ответов. Отвечая на неопределённую посылку типа «поисковое предписание», реактивный агент может использовать одну из двух стратегий: (1) по своему усмотрению выбрать класс, к которому принадлежит ответ; (2) с целью уменьшения или снятия неопределённости, вместо ответа сформулировать уточняющий вопрос. Последний случай означает обмен ролями между диалоговыми агентами, при котором реактивный агент берет на себя роль активного агента. Таким образом, получение реактивным агентом неопределённой посылки типа «поисковое предписание» может рассматриваться как одна из причин смены ролей в диалоговом процессе. Очевидно, что процесс уточнения неопределённой посылки типа «поисковое предписание» может быть итерационным, если реактивный агент воспринимает новую посылку как неопределённую. Например, посылка активного агента

Кто живёт за этой дверью?

является неопределённой, поскольку запрашиваемые знания могут принадлежать одному из нескольких классов: (1) знания о возрасте жильца; (2) знания о внешности жильца, и т. д. Реактивный агент, получив такую посылку и желая уточнить, какие знания запрашивает активный агент, может, изменив роль, вместо ответа передать своему партнёру следующую уточняющую посылку:

Вы хотите знать имя жильца, который живёт за этой дверью, либо что-то другое?

Если последующая посылка активного агента сформулирована в виде

Не могли бы Вы его описать?,

то эту посылку реактивный агент опять может интерпретировать как неопределённую посылку, поскольку она предполагает, что запрашиваемые знания принадлежат одному из нескольких классов.

Определённая посылка типа «поисковое предписание» означает, что запрашиваются знания, принадлежащие только одному классу, например:

Назовите фамилию человека, который живёт за этой дверью.

Посылки типа «ли» и посылки типа «какой». Получив неопределённую посылку типа «поисковое предписание», реактивный агент, в конечном итоге, всегда отвечает на определённую посылку. Он приходит к этому либо путем смены роли и уточнения класса знаний, к которому принадлежит ответ, либо произвольным выбором этого класса. Каждая определённая посылка

типа «поисковое предписание» содержит *структурный компонент*, который задает класс запрашиваемых знаний. В приведенном примере определенной посылки

Назовите фамилию человека, который живёт за этой дверью.

этот класс знаний представляет собой множество фамилий.

Таким образом, один из структурных компонентов посылки типа «поисковое предписание» предназначен для того, чтобы детерминировать класс запрашиваемых знаний. Белман и Стил в своей версии логики вопросов и ответов называют эту часть вопроса «*субъектом вопроса*» [96]. Следуя Белману и Стилу, будем компонент определенной посылки типа «поисковое предписание», детерминирующий класс запрашиваемых знаний, называть *субъектом поискового предписания*.

Субъект поискового предписания может детерминировать класс, в котором имеется конечное количество экземпляров, либо класс, который содержит бесконечное количество экземпляров. Назовем поисковые предписания, субъекты которых детерминируют конечный класс, *ли(whether)-посылками*, а поисковые предписания, субъекты которых детерминируют классы с бесконечным количеством экземпляров — *какой(which)-посылками*.

В первой транзакции, приведенного фрагмента диалога Протагор Сократ, являясь активным агентом, передает Гиппократу следующую посылку:

Так мы идём платить деньги Протагору, потому что он софист?

Если предположить, что Гиппократ имеет заранее сформированное мнение относительно того, по какой причине он хочет платить деньги Протагору, то приведенная посылка является посылкой типа «поисковое предписание». Далее, поскольку ответ Гиппократа принадлежит только одному классу ответов, то эта посылка является определенной. Этот единственный класс ответов является конечным и состоит только из двух экземпляров: (1) утвердительный ответ и (2) отрицательный ответ. Поэтому, в соответствии с только что введенной классификацией, мы можем классифицировать рассматриваемую информационную посылку Сократа как *ли-посылку*.

Множество примеров какой-посылок можно найти в [96]. Например,

Какова температура замерзания воды по Фаренгейту при нормальных условиях?

Если предположить, что реактивный агент знает ответ, то это — определенная посылка типа «поисковое предписание», поскольку запрашиваемые знания принадлежат одному классу, экземпляры которого представляют собой температуры замерзания жидкостей при нормальных условиях. Количество

экземпляров этого класса настолько велико, что можно считать эту посылку — какой-посылкой.

В ряде случаев, субъект ли-посылки может быть задан перечислением экземпляров соответствующего класса. Примером может быть субъект следующей ли-посылки, заимствованной из [96].

Курение табака — это порок, добродетель, причуда, сумасбродство или панацея от всех невзгод?»

В приведенном примере субъект ли-посылки задан следующим списком: «порок, добродетель, причуда, сумасбродство, панацея от всех невзгод».

На рис. 2.3 рассмотренные типы посылок активного агента представлены в виде диаграммы классов.

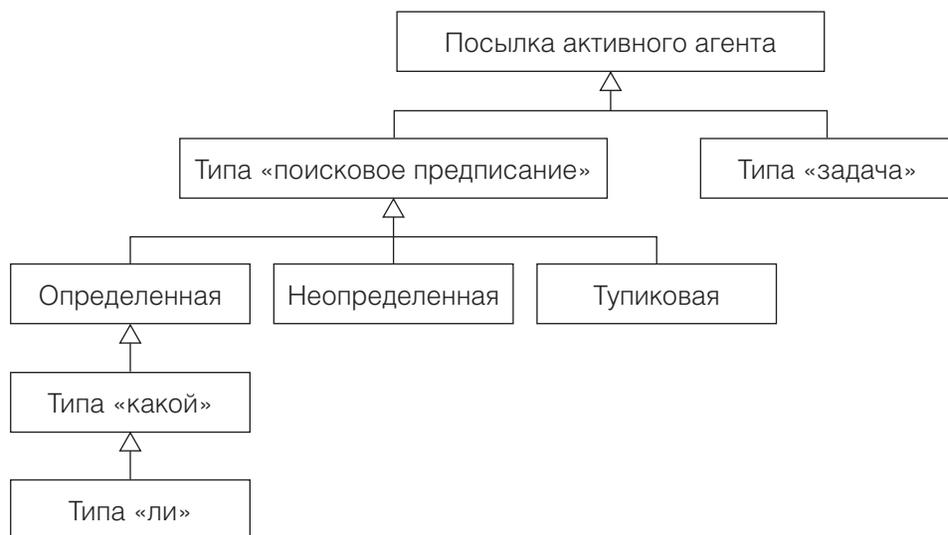


Рис. 2.3. Типы посылок активного агента.

На рис. 2.3 посылки типа «какой» и «ли» находятся в отношении суперкласс-подкласс. Поскольку эти типы посылок отличаются только мощностью субъекта поискового предписания, то класс какой-посылка является более широким и должен рассматриваться как подкласс класса ли-посылка. Поэтому в дальнейшем мы не будем пользоваться термином «элементарный вопрос».

Классификация информационных посылок активного агента, приведенная на рис. 2.3, не претендует на полноту и может быть продолжена, если, например, рассматривать подклассы посылок типа «задача» или подклассы неопределенных посылок типа «поисковое предписание».

2.2.5. Типы информационных посылок реактивного агента

Прямые и не прямые посылки. Посылку реактивного агента мы понимаем как поисковое предписание, адресованное активному агенту и необходимое для получения доступа к запрашиваемым знаниям, или как некоторую ссылку на участок долговременной памяти активного агента, где хранятся запрашиваемые знания. Реактивный агент, формируя свою посылку, может непосредственно сослаться на запрашиваемые знания. Такую посылку будем называть *прямой посылкой реактивного агента*. Примером транзакции, в которой посылка реактивного агента непосредственно ссылается на запрашиваемые знания, может служить следующая.

Активный: Какова температура замерзания воды по Фаренгейту при нормальных условиях?»
Реактивный: Температура замерзания воды при нормальных условиях равна 32 градусам по шкале Фаренгейта.

Однако реактивный агент, формируя свою посылку, может сослаться на запрашиваемые знания опосредованно. Такую посылку будем называть *непрямой посылкой реактивного агента*. Приведенную выше транзакцию несложно представить в форме, когда посылка реактивного агента ссылается на запрашиваемые знания опосредовано.

Активный: Какова температура замерзания воды по Фаренгейту при нормальных условиях?
Реактивный: Температура замерзания жидкостей при нормальных условиях по Фаренгейту указана в приложении.

Кодированные посылки. В примерах естественного диалога часто встречаются случаи, когда реактивный агент сокращает размер своей прямой посылки при помощи кодирования, или замены ее на компактный код, смысл которого определяется контекстом посылки активного агента. Так, например, в первом из только что приведенных примеров реактивный агент вместо прямой посылки

Температура замерзания воды при нормальных условиях равна 32 градусам по шкале Фаренгейта,

мог бы передать ее код, например, в виде числа, соответствующего запрашиваемой температуре.

Активный: Какова температура замерзания воды по Фаренгейту при нормальных условиях?»
Реактивный: 32 градуса.

В первой транзакции приведенного фрагмента диалога Протагор

С: Так мы идём платить деньги Протагору, потому что он софист?

Г: Конечно.

слово «конечно», возвращаемое Гиппократом в качестве прямой посылки, является кодом, заменяющим посылку «Да, мы идем платить деньги Протагору, потому что он софист».

Таким образом, приведенные примеры транзакций позволяют разделить прямые посылки реактивного агента на два класса: *кодированные посылки* и *некодированные посылки*.

К вопросу о типологии информационных посылок реактивного агента относится вопрос об *истинности или ложности этих посылок*.

Проблема истинности информационной посылки реактивного агента при моделировании естественного диалога не может решаться так просто, как это принято в современной эпистемологии и эротетической логике, где утверждается, что ответ может быть либо истинным, либо ложным и третьего не дано. Такое абсолютное понимание проблемы истинности/ложности ответа, по всей видимости, связано с тем, что в отмеченных науках анализу подвергаются изолированные вопросно-ответные пары, а не диалоги. Целеориентированное диалоговое взаимодействие в виде логически связанных транзакций *предопределяет относительность понятия истинность или ложность ответа*. Реактивный агент в ходе диалогового процесса решает свою проблему и формирует ответы, исходя не только из содержания вопросов, но и из той цели, которой он старается достичь в ходе диалога. По отношению к цели реактивного агента можно сформулировать примерный список правил, которыми он может руководствоваться при конструировании ответа на определенный вопрос. Этот список может состоять, например из следующих правил: (1) если вопрос не затрагивает нежелательные для меня предметы, то я искренне отвечаю на него; (2) если вопрос затрагивает нежелательные для меня предметы, то в ответе я скрываю/искажаю часть того, что знаю; (3) если вопрос затрагивает нежелательные для меня предметы, то в ответе я скрываю/искажаю всё, что знаю. Для того, чтобы при ответе на вопрос реактивный агент мог руководствоваться одним из приведенных выше правил, он должен прийти к нему в ходе рассуждений, предваряющих ответ. Во время этих рассуждений реактивный агент оценивает целесообразную степень полноты/искренности ответа по отношению к цели реактивного агента. Такие рассуждения являются опцией, которой может воспользоваться опытный и осторожный реактивный агент или не воспользоваться неопытный и темпераментный реактивный агент. В общем случае проблема истинности или ложности информационной посылки реактивного агента связана с проблемой моделирования «неискренности» реактивного агента, которая рассмотрена в подразделе 3.4.

2.3. Диалоговая транзакция в контексте представления знаний

Диалоговая транзакция является «молекулой» диалогового процесса. Построение искусственного диалогового агента, способного поддерживать естественный диалоговый процесс на основе когнитивного цикла, приведенного на рис. 2.2, невозможно без исследования внутренней организации диалоговой транзакции. Важными являются вопросы о логической структуре диалоговой транзакции, характере знаний, ассоциированных с отдельной транзакцией, их представлением и моделированием.

2.3.1. Знания, ассоциированные с диалоговой транзакцией

Последующее изложение базируется на некотором предположении относительно естественного диалога, заключающееся в том, что вместо полномасштабного естественного диалога рассматривается его упрощенный вариант, называемый *эротетическим диалогом* [97, 98].

В эротетическом диалоге информационные посылки активного агента ограничены относительно простыми типами. Это определенные посылки типа поисковое предписание с ли- или какой-субъектом поискового предписания. Иными словами, в эротетическом диалоге информационные посылки активного агента имеют стопроцентную степень интеррогативности и представляют собой вопросы в лингвистическом смысле слова, а информационные посылки реактивного агента — ответы на эти вопросы. Эротетический диалог — это шаг в направлении упрощения естественного диалога, который, однако, представляется целесообразным по следующим причинам. Во-первых, естественный диалог, как правило, можно конвертировать в эротетический, сохранив при этом исходные цель и метод, во-вторых, исследование эротетического диалога может базироваться на более или менее формализованных теориях и, в-третьих, результаты, полученные при исследовании эротетического диалога, могут быть хорошей основой для их обобщения на полномасштабный естественный диалог.

В эротетическом диалоге знания, запрашиваемые активным агентом и возвращаемые реактивным агентом, являются декларативными. Поэтому модель структуры эротетической транзакции в контексте представления знаний должна основываться на некоторой приемлемой концепции представления декларативных знаний. Хотя система декларативных знаний человека, по всей видимости, едина и подобна ткани, в которой близлежащие части сцеплены между собой, вопросы и ответы оперируют с отдельными небольшими частями или порциями системы декларативных знаний человека. Поэтому нам понадобятся средства представления только порции декларативных знаний, ассоциированных с эротетической транзакцией.

Эпистемологи, часто рассматривающие знания в контексте отдельной эротетической транзакции, обычно не используют термины «декларативные» и «процедурные» знания, а также декларативно-процедурную дихотомию знаний. Однако из контекста ряда публикаций ясно, что так называемые знания-that и знания-wh — это не что иное, как эпистемологическое наименование порций декларативных знаний, участвующих в эротетической транзакции [94]. Эти порции знаний представлены в виде естественноречевых предложений, являющихся ответами на that- или wh-вопросы, соответственно.

Wh-вопрос — это класс вопросов, обобщающий шесть подклассов: (1) who/кто-вопрос; (2) what/что-вопрос; (3) when/когда-вопрос; (4) where/где-вопрос; (5) how/как-вопрос и (6) why/почему-вопрос. Знания, выражаемые ответами на вопросы из перечисленных классов, формируют шесть классов декларативных знаний, рассматриваемых эпистемологами: (1) кто-знания; (2) что-знания; (3) когда-знания; (4) где-знания; (5) как-знания и (6) почему-знания. Эпистемологическое понимание знаний-wh можно проиллюстрировать примерами следующих шести естественноречевых предложений: (1) “Я знаю, *кто* есть Протагор”, (2) “Я знаю, *что* он будет говорить”, (3) “Я знаю, *когда* он начнёт беседу”, (4) “Я знаю, *где* будет беседа”, (5) “Я знаю *как* Протагор начнёт выступление” и (6) “Я знаю, *почему* его аргументы некорректны”. Приведенная классификация отражает и иллюстрирует явную пропозициональную ориентацию эпистемологического представления знаний, ассоциированных с эротетической транзакцией.

2.3.2. Тернарное представление декларативных знаний

В эпистемологии и других дисциплинах, изучающих эротетические транзакции, декларативные знания, участвующие в транзакции, представляются в виде пропозиций и, следовательно, элементарная порция декларативных знаний рассматривается как предложение. Общепринятый способ формализации такого представления знаний заключается в трансляции естественноречевых предложений в предложения, записанные в онтологическом базисе и нотации логики первого порядка. Например, класс что-знаний, в общем случае, может быть представлен предложением

$$(\exists x) \text{ а знает, что } (b = x)$$

Моделирование эротетической транзакции, в контексте представления декларативных знаний, может осуществляться средствами с более унифицированной онтологией, чем формальная система логики первого порядка. Под унифицированной онтологией здесь понимается онтология, основанная не на лингво-зависимых переменных (как в логике первого порядка), а на лингвонезависимых

фундаментальных сущностях декларативных знаний. В этом смысле привлекательным является язык тернарного описания (ЯТО), первоначально предложенный Уёмовым для формального описания параметрического варианта общей теории систем, а затем развитый им и его учениками в самостоятельную неклассическую логику [99–101].

Исходной сущностью в ЯТО является *объект*. Объект, в зависимости от его места в структуре знаний, существует в одной из трёх форм: *объект-вещь*, *объект-свойство* и *объект-отношение*.

Категории «вещь» и «свойство» имеют в ЯТО традиционный смысл, а смысл категории «отношение» несколько отличается от традиционного. Общепринято использовать понятие «отношение» для обозначения взаимовлияния нескольких вещей, т. е. говорить об отношении между вещами. Например, между двумя мужчинами может существовать отношение «отцовство». В контексте ЯТО под отношением понимается то, что образует вещь, т. е. говорится об отношении на вещи или отношении, имеющем место в вещи. Иными словами, ЯТО-отношение — это, в каком-то смысле, другое наименование внутренней организации вещи.

Бинарная ассоциация объекта-вещи с объектом-свойством порождает два прототипа для представления сущностей в ЯТО.

1. Наименование первого прототипа: «*вещь, обладающая свойством*», а формальная нотация имеет вид $(*)^*$.
2. Наименование второго прототипа: «*свойство, присущее вещи*», а формальная нотация имеет вид $(*)^*$.

Бинарная ассоциация объекта-вещи с объектом-отношением порождает ещё два прототипа:

3. Наименование третьего прототипа «*вещь, в которой имеет место отношение*», а формальная нотация имеет вид $*(*)$.
4. Наименование четвертого прототипа «*отношение, имеющее место в вещи*», а формальная нотация имеет вид $*((*)$.

Для формального представления сущностей в ЯТО используется специальная скобочная нотация. Символ «звёздочка» указывает на знакоместо для символа объекта. Символ объекта-вещи записывается внутри скобок, символ объекта-свойства — справа от скобок, а символ объекта-отношения — слева от скобок.

Ассоциация объекта-вещи с объектом-свойством или с объектом-отношением имеет направление. Если символ объекта-вещи заключён в обычные (одинарные) круглые скобки, то это означает, что ассоциация направлена от объекта-вещи к объекту-свойству или объекту-отношению. Словами это можно выразить как: «вещь, обладающая свойством», или «вещь, в которой имеет место отношение». Ассиметричная (двойная) круглая скобка означает, что ассоциация направлена от объекта-свойства или объекта-отношения к объекту-вещи. Словесная формулировка: «свойство, присущее вещи», или «отношение, имеющее место в вещи».

Отметим, что в публикациях, посвящённых ЯТО, знакоместо внутри скобок используется только для символа объекта-вещи. Это ограничение исключает из набора прототипов следующие два: «отношение, обладающее свойством», и «свойство, присущее отношению».

Объект, в зависимости от степени неопределённости знаний о нём, существует в одной из трёх форм:

- *определённый* (в знакоместо прототипа подставляется символ t),
- *неопределённый* (в знакоместо прототипа подставляется символ a),
- *произвольный* (в знакоместо прототипа подставляется символ A).

Категории «вещь, свойство и отношение», а также «определённость, неопределённость и произвольность» независимы и формируют девять классов объектов: (1) определённый объект-вещь, (2) неопределённый объект-вещь, (3) произвольный объект-вещь, (4) определённый объект-свойство, (5) неопределённый объект-свойство, (6) произвольный объект-свойство, (7) определённый объект-отношение, (8) неопределённый объект-отношение, (9) произвольный объект-отношение.

Подставляя в приведенные выше прототипы вместо символа звёздочка символы t , a или A , получим множество возможных моделей порций декларативных знаний в онтологическом базисе ЯТО. Ниже приведены несколько моделей.

Модель

$$t(a) \tag{2.1}$$

представляет знания относительно некоторой определённой вещи, которая обладает каким-то (неопределённым) свойством. Направление ассоциации от вещи к свойству.

Модель

$$(t)a \tag{2.2}$$

представляет знания относительно какого-то (неопределённого) свойства, которое присуще некоторой определённой вещи. Направление ассоциации от свойства к вещи.

Модель

$$(a)A \tag{2.3}$$

представляет знания относительно некоторого произвольного свойства, которое присуще какой-то (неопределённой) вещи. Направление ассоциации от свойства к вещи.

Модель

$$a(t) \tag{2.4}$$

представляет знания относительно некоторой определённой вещи, в которой имеет место какое-то (неопределённое) отношение. Направление ассоциации от вещи к отношению.

Поскольку имеется четыре прототипа, каждый из которых заполняется объектами из девяти классов, то общее количество моделей элементарных порций декларативных знаний в онтологическом базисе ЯТО равно тридцати шести.

2.3.3. Логическая структура диалоговой транзакции

Логическую структуру посылки активного агента в эротетической транзакции протокола, в контексте ЯТО-представления декларативных знаний, будем рассматривать как развитие теории эротетической логики, предложенной Белнапом и Стилом [96].

Теория Белмана и Стила ориентирована исключительно на естественноречевые эротетические транзакции протокола, что отразилось как на их классификации, так и на тех формализмах, которые предлагаются для моделирования логической структуры транзакций. Теория не является универсальной по отношению ко всем возможным естественноречевым вопросам и охватывает лишь те, которые допустимы в эротетическом диалоге.

Важным элементом концептуального базиса теории Белнапа и Стила является понятие прямой посылки реактивного агента, или прямого ответа, согласно терминологии Белмана и Стила. Прямой ответ понимается Белнапом и Стилом как естественноречевое утверждение, которое может быть оценено как истинное или ложное. Теория Белнапа и Стила ориентирована исключительно на прямые посылки реактивного агента, поэтому в дальнейшем под словом «ответ» мы всегда будем подразумевать прямую посылку реактивного агента.

Приписывая прямому ответу истинность или ложность, Белнап и Стил абсолютизируют понятие истинности ответа, утверждая, что с любым ответом необходимо ассоциировать одну из альтернативных оценок «истина» или «ложь». Абсолютизация понятия истинности или ложности ответа является важным элементом философии Белнапа и Стила, которая нашла отражение в предложенных ими формально-логических моделях эротетической транзакции. Однако, такая абсолютизация является упрощением реальной ситуации. Она предполагает, что имеется некоторая эталонная/абсолютная система знаний вне сознания активного или реактивного агентов, по отношению к которой оценивается истинность прямого ответа.

В общем случае, понятие истинности или ложности ответа относительно и определяется по отношению к знаниям того или иного диалогового агента. Это становится очевидным, если рассматривать эротетическую транзакцию не саму по себе, а в контексте диалогового процесса. В случае межличностного диалога, вопросы и ответы формируются различными агентами, обладающими различными знаниями и преследующими различные цели. Ответ формируется реактивным агентом, и его истинность или ложность должны рассматриваться по

отношению как к системе знаний реактивного агента, так и к цели, которую он преследует. Поскольку система знаний и цели агентов могут не совпадать, то ответ, который является ложным для активного агента, может быть вполне истинным для реактивного. Например, в примере вопроса о курении табака, приведенном в 2.2.4, отношение к курению табака активного агента может не совпадать с отношением к курению реактивного агента. Активный агент может считать, что курение — это порок, а реактивный — что это добродетель.

Белнап и Стил постулируют, что логическая структура любого вопроса активного агента может быть представлена двумя компонентами:

- субъектом вопроса и
- требованием вопроса.

Субъект вопроса активного агента явно или неявно представляет некоторое множество утверждений (элементов субъекта). Утверждения понимаются в смысле ассерторических суждений, в которых констатируется наличие или отсутствие у предмета некоторого признака. В субъекте могут быть как истинные, так и ложные утверждения. Таким образом, Белнап и Стил абсолютизируют понятие истинности или ложности не только для прямого ответа, но и для утверждений субъекта вопроса активного агента. Реактивный агент формирует свой ответ путем выбора некоторого количества элементов субъекта вопроса активного агента. Например, вопрос активного агента

Является ли стекло жидкостью при 70° F?

предполагает, что субъект вопроса состоит из двух элементов

Стекло является жидкостью при 70° F. и
Стекло не является жидкостью при 70° F.

В приведенном примере множество элементов субъекта вопроса конечно. Несложно найти примеры, в которых субъект вопроса состоит из бесконечного множества элементов. Например, рассмотренный ранее вопрос

Какова температура замерзания воды по Фаренгейту при нормальных условиях?

предполагает бесконечно много элементов в субъекте

Температура замерзания воды при нормальных условиях равна 0° F.
Температура замерзания воды при нормальных условиях равна 1° F.
и так далее.

Требование вопроса активного агента необходимо для специфицирования того, в каком виде должен быть представлен ответ, формируемый реактивным

агентом из субъекта вопроса. Для иллюстрации того, что в логической структуре вопроса, кроме субъекта, должен присутствовать компонент, названный требованием, Белнап и Стил приводят примеры нескольких вопросов, имеющих одинаковые субъекты, но предполагающие различные ответы.

Вопрос 1.

Какие простые числа находятся между 10 и 20?

Вопрос 2.

Приведите пример простого числа, находящегося между 10 и 20

Оба вопроса обладают одним и тем же субъектом, в качестве которого можно, например, рассматривать множество целых чисел, находящихся между числами 10 и 20. Ответ на первый вопрос — это список простых чисел, а ответ на второй — это одно из простых чисел, выбранных реактивным агентом в качестве примера.

Как видно из приведенных примеров, одному вопросу активного агента, в общем случае, соответствует некоторое количество ответов реактивного.

Белнап и Стил предлагают специфицировать требование вопроса активного агента при помощи трех компонентов:

- *спецификация размера выборки,*
- *спецификация полноты,*
- *спецификация различения.*

Под выборкой понимается подмножество элементов субъекта, переходящих в ответ. Спецификация размера выборки детерминирует количество элементов субъекта, которые должны войти в выборку и перейти в ответ. Размер выборки специфицируется путём указания нижнего и верхнего пределов для количества элементов.

Спецификация полноты дополняет спецификацию размера выборки и специфицирует долю истинных элементов субъекта, переходящих в ответ, по отношению ко всему множеству истинных элементов субъекта. Иными словами, это требование того, какая часть от всех истинных элементов субъекта должна перейти в ответ.

Спецификация различения индицирует, необходимо ли в ответе указывать, являются ли элементы субъекта реально или номинально различными. Реальные элементы субъекта — это некоторые сущности мира, а номинальные элементы субъекта — их имена. Необходимость этого компонента объясняется необходимостью различать в ответе различные имена для одной и той же сущности. Например, число «семь» может иметь два разных имени: арабская цифра «7» или римская цифра «VII».

Таким образом, требование вопроса включает некоторое количество спецификаций, среди которых обязательной является спецификация размера выборки.

Требование любого вопроса должно обязательно включать эту спецификацию. Спецификации полноты и различения являются необязательными и включаются в требование только некоторых вопросов. Таким образом, требование вопроса может быть представлено одной, двумя или тремя спецификациями. Формально, требование вопроса можно представить при помощи одного из следующих четырёх выражений

$$s \wedge c \wedge d, s \wedge c, s \wedge d, s, \quad (2.5)$$

где s — спецификация размера выборки;

c — спецификация требования полноты;

d — спецификация требования различения.

Задача обязательного компонента требования вопроса (спецификация размера выборки) заключается в указании мощности подмножества элементов субъекта, переходящих в ответ.

Спецификация полноты имеет смысл в том случае, когда субъект может включать как истинные, так и ложные утверждения. В том случае, когда все элементы субъекта — истинны, то спецификация требования полноты автоматически задаётся спецификацией размера выборки.

Спецификация различения имеет смысл тогда, когда элементы субъекта представлены именами (например, закрытыми терминами в смысле логики первого порядка), среди которых допускаются различные имена, именующие одну и ту же сущность. Если, при конструировании субъекта вопроса запретить использование различных имён для одной и той же сущности, то отпадает необходимость в спецификации различения.

Как было отмечено, в эротетическом диалоге вопрос активного агента ограничен определенной посылкой типа поисковое предписание с ли- или какой-субъектом. Это ограничение полностью соответствует теории Белмана и Стила, в которой вопросы классифицируются по отношению к мощности субъекта. Субъект вопроса может детерминировать либо (1) конечное или небольшое и обозримое количество элементов, либо (2) бесконечное или очень большое количество элементов. В первом случае соответствующие вопросы называются ли-вопросами, а во-втором, — какой-вопросами.

Формально логическую модель ли- или какой-вопроса Белнап и Стил называют интеррогативной формулой, которая, в общем случае, записывается в виде

$$Que \stackrel{\text{def}}{=} ? Subj, Req, \quad (2.6)$$

где: $Subj$ — субъект вопроса;

Req — требование вопроса;

$?$ — признак интеррогатива.

С учётом трёхкомпонентной структуры требования, интеррогативная формула имеет вид

$$Que \stackrel{\text{def}}{=} ?Subj, \langle s c d \rangle. \quad (2.7)$$

В случае ли-вопроса интеррогативная формула имеет вид

$$Whether - Que \stackrel{\text{def}}{=} ? \langle A_1, \dots, A_n \rangle, Req, \quad (2.8)$$

где $\langle A_1, \dots, A_n \rangle$ — субъект ли-вопроса, или *ли-субъект*.

Ли-субъект представляет собой список предложений, относительно которого вводится несколько ограничений. Во-первых, предполагается, что среди элементов списка нет повторяющихся, а во-вторых, предполагается, что список может включать как элементы, представляющие собой простые предложения, так и элементы, представляющие собой конъюнкции простых предложений. При этом хотя бы один из конъюнктов не должен являться элементом списка.

Частным случаем ли-вопроса является вопрос типа да-нет. Субъект вопроса типа да-нет состоит из двух простых взаимно противоположных утверждений, и его интеррогатив записывается в виде

$$Yes/No - Que \stackrel{\text{def}}{=} ? \langle A, \bar{A} \rangle, Req. \quad (2.9)$$

Рассмотрим теперь, каким образом Белнап и Стил предлагают формально представлять субъект и интеррогативную формулу какой-вопроса. Вначале — несколько примеров. Для вопроса

Какое целое положительное число является наименьшим простым числом, большим, чем 45?

субъект может быть задан при помощи следующих двух предложений с переменной

x является наименьшим простым числом, большим, чем 45
x является положительным целым числом

Первое из приведенных предложений названо *матрицей*, а второе — *категорным условием*. Матрица представлена при помощи естественно-языкового предложения, в которое включён символ переменной, а категорное условие представляет собой ограничение, ограничивающее возможные значения переменной. Заменяя в матрице переменную x её значениями, удовлетворяющими категорному условию, можно получить сколь угодно много элементов субъекта.

В матрице может быть несколько переменных и в этом случае необходимо сформулировать несколько категорных условий. Например, субъект вопроса

Какие мальчики являются братьями каких девочек?

может быть задан следующими предложениями:

x является братом у	(матрица)
x есть мальчик	(первое категорное условие)
y есть девочка	(второе категорное условие)

В обоих примерах какой-вопросов значения переменных представляют собой имена сущностей, имеющих категориальный статус вещи. Это подсказывает иной подход к формально-логическому моделированию вопроса и, в частности, иной способ представления субъекта. Нотация, предлагаемая Белнапом и Стиллом для формально-логического моделирования вопросов, сильно привязана к естественному языку. Она предполагает, что элементами субъекта являются естественныеязыковые предложения. Однако, не следует исключать случаи, когда элементами субъекта являются не символные образы или изображения. Такого типа субъекты часто используются в вопросах, используемых в психологических тестах. Для того, чтобы уйти от привязки к естественному языку и сделать формально-логические модели вопросов более универсальными, целесообразно воспользоваться концептуальным базисом ЯТО, рассмотренным в 2.3.2. Полученные в рамках отмеченного концептуального базиса формально-логические модели вопросов и ответов приобретают статус моделей представления знаний в эротетических транзакциях.

Таким образом, согласно Белнапу и Стиллу, субъект какой-вопроса можно представить тройкой

$$\langle X, g, A(X) \rangle, \quad (2.10)$$

где X — множество переменных $\{x_1, \dots, x_n\}$, используемых в матрице;

g — категорное отображение;

$A(X)$ — матрица.

Категорное отображение ставит в соответствие элементам множества X категорные условия. Отношение между X и набором категорных условий не взаимно однозначное. Если некоторой переменной не поставлено в соответствие категорное условие, то это означает, что переменная никак не ограничена, может принимать произвольные значения и называется категорно-свободной.

Вместо категорного отображения g в субъекте какой-вопроса можно явно указать список категорно-ограниченных и категорно-свободных переменных

$$\langle C_1 x_1, \dots, C_r x_r, x_{r+1}, \dots, x_n // A(X) \rangle, \quad (2.11)$$

где $C_1 x_1, \dots, C_r x_r$ — список категорно-ограниченных переменных;

x_{r+1}, \dots, x_n — список категорно-свободных переменных;

$A(X)$ — матрица.

Последняя нотация субъекта какой-вопроса может быть использована для записи интеррогативной формулы какой-вопроса в виде

$$\text{Which} - \text{Que} \stackrel{\text{def}}{=} ? \langle C_1 x_1, \dots, C_r x_r, x_{r+1}, \dots, x_n // A(X) \rangle, \text{Req} \quad (2.12)$$

Идея представления субъекта какой-вопроса в виде матрицы и списка категорных условий обладает тем недостатком, что допускает многозначность трансляции формально-логической модели одного и того же вопроса в естественноречевые предложения. Отмеченная многозначность обусловлена тем, что матрица субъекта и, следовательно, соответствующий ей список категорных условий могут быть сформированы множеством различных способов. Это легко продемонстрировать на примере вопроса о простых числах, лежащих в диапазоне от 10 до 20:

Какие простые числа находятся между 10 и 20?

Интеррогатив этого вопроса имеет вид

$$\text{Which} - \text{Que} \stackrel{\text{def}}{=} ? \langle Cx // A(x) \rangle, \text{Req}, \quad (2.13)$$

где категорное условие Cx и матрица $A(x)$ могут быть представлены различным образом. Например:

(x есть целое // x является простым числом, находящимся между 10 и 20),
 (x есть простое // x находится между 10 и 20).

Избавиться от этого недостатка можно, если конструировать субъект не из лексических элементов, а из элементов, инвариантных к лексике естественного языка. Хорошим кандидатом являются такие категории, как вещь, свойство и отношение.

Ответ на любой ли- или какой-вопрос представляет собой часть субъекта вопроса, выделенную из всего субъекта в соответствии с требованием вопроса. Часть субъекта, переходящая в ответ, может быть представлена в виде конъюнкции либо в виде списка элементов субъекта. Это эквивалентные представления, поскольку из списка предложений выводится их конъюнкция (правило введения конъюнкции). Количество элементов списка или количество конъюнктов — это размер выборки, который задаётся спецификацией размера выборки. Таким образом, логическая структура ответа может быть записана в виде

$$\text{Ans} \stackrel{\text{def}}{=} (\bigwedge_{i=1}^{i=p} A_i) \wedge c \wedge d, \quad (2.14)$$

где A_i — элемент субъекта вопроса;

p — количество элементов субъекта вопроса, переходящих в ответ (размер выборки);

c — необязательная спецификация полноты;

d — необязательная спецификация различения.

2.3.4. Модели диалоговой транзакции в контексте тернарного представления декларативных знаний

Используем онтологический базис и формулы ЯТО, а также теорию Белмана и Стила для синтеза моделей эротетической транзакции в контексте ассоциированных с ней декларативных знаний [102, 103].

Логическую структуру посылки активного агента в эротетической транзакции, в контексте ЯТО-представления декларативных знаний, будем рассматривать как развитие базовой интеррогативной формулы Белнапа и Стила, постулирующей, что внутренняя логическая структура вопроса детерминируется субъектом вопроса и требованием вопроса:

$$Que = Subj, Req. \quad (2.15)$$

Важным элементом философии Белнапа и Стила, лежащей в основе их теории, является концепция субъекта вопроса. Из этой концепции следует, что те вопросы, модель которых включает субъект, например, вопросы типа «поисковое предписание», не только предопределяют возможные ответы, но и содержат их. Это означает, что субъект вопроса можно мыслить как ответ с некоторой степенью неопределённости.

Субъект вопроса является ключевым компонентом эротетической транзакции. Реактивный агент при конструировании ответа занимается, по сути, тем, что трансформирует неопределённые знания, на которые указывает субъект вопроса, в более определённые знания, ассоциированные с ответом. Иными словами, реактивный агент понижает степень неопределённости знаний субъекта вопроса до уровня, приемлемого для ответа.

Будем моделировать порции декларативных знаний, на которые указывает субъект вопроса, следующими альтернативными моделями:

$$K_{subj} = (t)a, \quad (2.16)$$

$$K_{subj} = (a)t, \quad (2.17)$$

$$K_{subj} = a(t), \quad (2.18)$$

$$K_{subj} = t((a)). \quad (2.19)$$

Модели (2.16)–(2.19) получены из четырёх рассмотренных ранее ЯТО-прототипов порций декларативных знаний для случая ассоциации определённого и неопределённого ЯТО-объектов. Имеется несколько причин выбора моделей (2.16)–(2.19) для представления порции декларативных знаний. Во-первых, субъекты всех примеров вопросов, приведенных в [94, 96], могут быть представлены одной из моделей (2.16)–(2.19).

Во-вторых, модели (2.16)–(2.19) являются именно теми моделями (из тридцати шести возможных, упомянутых в 2.3.2.), которые соответствуют идее о том, что реактивный агент при конструировании ответа трансформирует неопределённые знания субъекта вопроса в определённые знания ответа. Один из объектов в моделях (2.16)–(2.19) является неопределённым, и более конкретные знания именно о нём активный агент ожидает получить от реактивного агента.

Поясним модели (2.16)–(2.19) на содержательном уровне.

- Модель $K_{subj} = (t)a$ представляет знания о конкретной определённой вещи, которой присуще некоторое неопределённое свойство. Вопрос с таким субъектом формируется активным агентом в том случае, когда он хочет узнать, *какими свойствами обладает данная вещь*.
- Модель $K_{subj} = (a)t$ представляет знания об определённом свойстве, которое присуще некоторой неопределённой вещи. Вопрос с таким субъектом формируется активным агентом в том случае, когда он хочет узнать, *каким вещам присуще данное свойство*.
- Модель $K_{subj} = a(t)$ представляет знания о конкретной определённой вещи, в которой имеет место некоторое неопределённое отношение. Вопрос с таким субъектом формируется активным агентом в том случае, когда он хочет узнать, *какое отношение имеет место в данной вещи*.
- Модель $K_{subj} = t(a)$ представляет знания о конкретном определённом отношении, которое имеет место в некоторой неопределённой вещи. Вопрос с таким субъектом формируется активным агентом в том случае, когда он хочет узнать, *в каких вещах имеет место данное отношение*.

Недостатком моделей (2.16)–(2.19), с точки зрения инжиниринга программных систем, основанных на знаниях, является их плохая приспособленность для отображения в релевантные структуры данных. Эти модели могли бы быть полезными для целей разработки программных систем в случае, если бы мы нашли способ их трансформации в типы или структуры данных современных систем программирования. Речь идёт, в первую очередь, о даталогической интерпретации неопределённого объекта.

Одной из возможных даталогических интерпретацией неопределённости является множественность. Неопределённый объект можно понимать как множество определённых объектов, а мощность этого множества — как меру его неопределённости. Тогда понижение степени неопределённости объекта эквивалентно уменьшению мощности соответствующего множества. Неопределённый объект превращается в абсолютно определённый в том случае когда мощность множества становится равной единице или когда множество представлено одним определённым объектом.

С учётом такой интерпретации неопределённости, мы можем заменить неопределённые объекты списками определённых объектов и представить модели (2.16)–(2.19) виде:

$$K_{subj} = \langle \text{объект-вещь} \rangle \{ \text{список объектов-свойств} \}, \quad (2.20)$$

$$K_{subj} = \langle \text{объект-свойство} \rangle \{ \text{список объектов-вещей} \}, \quad (2.21)$$

$$K_{subj} = \langle \text{объект-вещь} \rangle \{ \text{список объектов-отношений} \}, \quad (2.22)$$

$$K_{subj} = \langle \text{объект-отношение} \rangle \{ \text{список объектов-вещей} \}. \quad (2.23)$$

Модели (2.20)–(2.23) являются даталогическими аналогами моделей (2.16)–(2.19). Эти модели следует рассматривать как представление декларативные знания, передаваемые реактивному агенту посредством субъекта вопроса. Модели не привязаны к конкретному естественному языку, поскольку базируются на онтологии ЯТО.

Дополним модели (2.20)–(2.23) двумя дополнительными моделями, которые непредставимы в нотации ЯТО в силу того, что внутри скобок разрешается записывать только символ объекта-вещи. Представим эти модели формулами (2.24) и (2.25).

$$K_{subj} = \langle \text{объект-свойство} \rangle \{ \text{список объектов-отношений} \}, \quad (2.24)$$

$$K_{subj} = \langle \text{объект-отношение} \rangle \{ \text{список объектов-свойств} \}. \quad (2.25)$$

Поскольку субъект вопроса является, по сути, ответом с некоторой степенью неопределенности и реактивный агент при конструировании ответа понижает эту степень до уровня, приемлемого для ответа, (детерминируется требованием вопроса *Req* в формуле (2.14), то модели порций декларативных знаний, ассоциированных с ответом, должны быть аналогичны (2.20–2.25). Отличие моделей декларативных знаний, ассоциированных с ответом, заключается в мощности множества (списка) объектов. Представим их в виде

$$K_{ans} = \langle \text{вещь} \rangle \text{ОБЛАДАЕТ СВОЙСТВАМИ} \{ \text{список свойств} \}, \quad (2.26)$$

$$K_{ans} = \langle \text{свойство} \rangle \text{ПРИСУЩЕ ВЕЩАМ} \{ \text{список вещей} \}, \quad (2.27)$$

$$K_{ans} = V \langle \text{вещи} \rangle \text{ИМЕЮТ МЕСТО ОТНОШЕНИЯ} \{ \text{список отношений} \}, \quad (2.28)$$

$$K_{ans} = \langle \text{отношение} \rangle \text{ИМЕЕТ МЕСТО В ВЕЩАХ} \{ \text{список вещей} \}, \quad (2.29)$$

$$K_{ans} = \langle \text{свойство} \rangle \text{ПРИСУЩЕ ОТНОШЕНИЯМ} \{ \text{список отношений} \}, \quad (2.30)$$

$$K_{ans} = \langle \text{отношение} \rangle \text{ИМЕЕТ МЕСТО В СВОЙСТВАХ} \{ \text{список свойств} \}. \quad (2.31)$$

Модели (2.26)–(2.31) представляют декларативные знания, которые реактивный агент возвращает активному агенту в своих ответах на вопросы активного агента с субъектами, декларативные знания которых представлены моделями (2.20)–(2.25).

Полученные модели можно относительно легко проиллюстрировать большим количеством примеров транзакций, построенных на основе ли-вопросов, приведенных в [96].

Рассмотрим транзакцию.

Активный: Какова температура замерзания воды по Фаренгейту при нормальных условиях?

Реактивный: При нормальных условиях температура замерзания воды равна 32°F.

Субъект вопроса в приведенной транзакции обеспечивает доступ к декларативным знаниям, представимым моделью (2.20)

$$K_{subj} = \langle \text{вода при нормальных условиях} \rangle \\ \{ \text{температуры замерзания жидкостей по шкале Фаренгейта} \}.$$

Модель представляет знания относительно данной определённой вещи с именем «вода при нормальных условиях», которая обладает некоторым неопределённым свойством — какой-то температурой замерзания. Активный агент, формулируя вопрос с таким субъектом, хочет перейти от неопределённости к определённости и получить доступ к знаниям реактивного агента относительно конкретного свойства «воды при нормальных условиях». Неопределённость в знаниях, на которые указывает субъект, выражается множеством, состоящим из конечного, но достаточно большого количества альтернатив. Знания, ассоциированные с ответом реактивного агента, вполне определённы и представимы моделью (2.26)

$$K_{ans} = \langle \text{вода при нормальных условиях} \rangle \text{ ОБЛАДАЕТ СВОЙСТВОМ} \\ \{ \text{замерзает при 32°F} \}.$$

Рассмотрим вопрос.

Какие простые числа лежат между 10 и 20?

Субъект этого вопроса обеспечивает доступ к декларативным знаниям, представимым моделью (2.21)

$$K_{subj} = \langle \text{быть простым числом} \rangle \{ 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 \}.$$

Модель представляет знания относительно данного определённого свойства с именем «быть простым числом», которое присуще некоторой неопределённой вещи — какому-то числу в диапазоне от 10 до 20. Активный агент, формулируя вопрос с таким субъектом, хочет перейти от неопределённости к определённости и узнать у реактивного агента, каким конкретным вещам присуще свойство «быть простым числом». Неопределённость в знаниях, на которые указывает субъект, выражается множеством, состоящим из одиннадцати элементов. Знания, ассоциированные с ответом реактивного агента, представимы моделью (2.27)

$$K_{ans} = \langle \text{быть простым числом} \rangle \text{ ПРИСУЩЕ ЧИСЛАМ} \{ 11, 13, 17, 19 \}.$$

В качестве следующего примера рассмотрим одну из транзакций приведенного в начальной части 2.2 фрагмента диалога Протагор. Фрагмент этого диалога не является исключительно эротетическим, поскольку информационные посылки активного агента не во всех транзакциях представлены вопросами. Однако, ряд транзакций этого диалога являются эротетическими. Рассмотрим первую из них.

С: Так мы идём платить деньги Протагору, потому что он софист?

Г: Конечно.

Субъект вопроса Сократа в этой транзакции обеспечивает доступ к декларативным знаниям, представимым моделью (2.20), которая представляет знания о данной определённой вещи «Сократ-и-Гиппократ», обладающей некоторым неопределённым свойством — готовностью или неготовностью платить деньги Протагору на том основании, что он софист. Сократ, формулируя вопрос с таким субъектом, хочет перейти от неопределённости к определённости и узнать у Гиппократа, каким конкретным свойством обладает вещь Сократ-и-Гиппократ. Сократ ожидает получить некоторый ответ и, исходя из своих ожиданий, конструирует субъект вопроса таким образом, чтобы неопределённость в знаниях, на которые указывает субъект, представлялась списком, состоящим только из нескольких альтернатив.

$$K_{subj} = \langle \text{Сократ-и-Гиппократ} \rangle \\ \{ \langle \text{давать деньги Протагору} \rangle, \langle \text{не давать деньги Протагору} \rangle \}.$$

Ответ Гиппократа является указателем на порцию декларативных знаний, выражаемых моделью (2.25)

$$K_{ans} = \langle \text{Сократ-и-Гиппократ} \rangle \text{ ОБЛАДАЕТ СВОЙСТВОМ} \\ \{ \langle \text{давать деньги Протагору} \rangle \}.$$

Эротетический диалог является одним из типов естественного диалога, которым могут оперировать искусственные диалоговые агенты, демонстрирующие антропоморфное поведение. Транзакции эротетического диалога ограничены вопросно-ответными парами и поэтому внешне эротетический диалог не настолько разнообразен, как ничем не ограниченный естественный диалог, однако он может достигать тех же целей и решать те же проблемы, что и полномасштабный естественный диалог.

Модели логической структуры транзакции эротетического диалога целесообразно строить, исходя из структуры знаний, ассоциированных с транзакцией. В этом случае модели не только способствуют лучшему пониманию внутренней сущности диалоговой транзакции, но и имеют прагматическую ценность, поскольку могут служить теоретической основой для проектирования базы знаний диалогового агента. Модели, представленные формулами (2.19)–(2.30), отражают внутреннюю логическую сущность эротетической транзакции в контексте представления знаний.

3 ЭРОТЕТИЧЕСКИЙ ДИАЛОГОВЫЙ ПРОЦЕСС

Модели, описанные в настоящем разделе, являются формальными или, по крайней мере, формализованными и моделируют упрощенный эротетический диалоговый процесс. При разработке моделей ставилась прагматическая цель — синтез универсального искусственного диалогового агента, способного участвовать в абстрактном эротетическом диалоге, не привязанном к конкретной проблемной области.

Согласно принятой договоренности, в эротетическом диалоге информационные посылки активного агента ограничены относительно простыми типами. Это определенные посылки типа поисковое предписание с ли- или какой-субъектом, среди которых отсутствуют тупиковые. Иными словами, в эротетическом диалоге информационные посылки активного агента имеют стопроцентную степень интеррогативности и могут быть представлены вопросами в лингвистическом смысле слова, а информационные посылки реактивного агента — ответами на эти вопросы. Предположим также, что диалог является не только эротетическим, но и гармоничным. Иными словами, роли диалоговых агентов зафиксированы, и агенты удовлетворены принятым распределением ролей. Предположим также, что агенты обладают знаниями, достаточными для достижения цели диалога, а реактивный агент всегда дает правдивые ответы.

Принятые допущения являются еще одним шагом в направлении упрощения естественного диалога. Однако, ограниченный таким образом естественный диалог, тем не менее, обладает способностью решать ряд практических задач. Для решения многих практических задач не нужна вся гибкость и универсальность естественного диалога. Как показано в первом разделе, гармоничный эротетический диалог достаточен для построения ряда диалоговых приложений. Он, например, достаточен для решения задачи обучения при помощи интеллектуального компьютерного тьютора.

3.1. Автоматная модель

Автоматная модель представляет собой конечно-автоматную интерпретацию когнитивного цикла диалогового процесса, рассмотренного в 2.1 и изображенного на рис. 2.2. Как было отмечено в 2.1, наличие в структуре когнитивного цикла компонента, названного сценарием диалога, позволяет активному агенту не синтезировать очередной вопрос с использованием внутренних ментальных ресурсов, а разыскивать его в памяти вопросов, используя в качестве метода доступа к памяти вопросов метод достижения цели диалога. Поэтому введем в структуру активного диалогового агента *память вопросов*, хранящую декларативные знания активного агента в виде информационных посылок-вопросов, а также *метод доступа к памяти вопросов*, обеспечивающий доступ к поименованным элементам памяти вопросов.

Автоматная модель представляет собой модель метода доступа к памяти вопросов и может рассматриваться также как модель метода достижения цели диалога, или диалоговым методом решения задачи. Необходимым условием функционирования автоматной модели является выполнение следующих условий: (1) цель диалогового процесса может быть достигнута за конечное количество транзакций, и к началу решения задачи детерминированы все вопросы и ответы, необходимые для формирования этих транзакций; (2) модификация набора ожидаемых ответов всегда порождает новый набор, отличающийся от текущего.

Автоматная модель базируется на определении автомата Мили [104] и, в общем случае, может быть описана следующей пятеркой:

$$DiAM = (QUE, ANS, T, f, g). \quad (3.1)$$

В формуле (3.1) *DiAM* обозначает диалоговый метод доступа и является аббревиатурой, полученной из слов *Dialogue Access Method*.

$$QUE = \{Que_i\}; i = 1, \dots, k \quad (3.2)$$

обозначает множество идентификаторов (имен) вопросов активного агента, которые, согласно условию (1), достаточны для формирования любой транзакции. Метод доступа оперирует только именами вопросов, которые следует понимать как символические адреса вопросов в памяти вопросов. Спецификации вопросов находятся в памяти вопросов, имеющей прямой доступ к своим адресуемым элементам. Таким образом, для получения доступа к спецификации какого-либо вопроса, хранящейся в памяти вопросов, необходимо передать ей имя вопроса. Множество *QUE* не может хранить спецификации вопросов по следующим причинам, во-первых, субъекты некоторых вопросов могут включать не только символьную, но и графическую информацию, и во-вторых, один и тот же вопрос может многократно использоваться в одном сценарии.

$$ANS = \{Ans_i\}; i = 1, \dots, l \quad (3.3)$$

обозначает множество ответов реактивного агента, которые, согласно условию (1), достаточны для формирования любой транзакции. Множество ANS может хранить реальные спецификации ответов, поскольку часто они представлены в кодированной форме.

$$T = \{T_i\}; i = 1, \dots, m \quad (3.4)$$

обозначает множество стабильных состояний автомата и соответствует множеству имен транзакций диалогового сценария. Напомним, что в отличие от транзакции диалогового протокола, транзакция диалогового сценария включает один вопрос и все потенциально возможные ответы. Стабильное состояние можно рассматривать как состояние ожидания активным агентом ответа реактивного агента или как состояние ожидания реактивным агентом вопроса активного агента.

$$f: T \times ANS \rightarrow T \quad (3.5)$$

обозначает функцию перехода, переводящую сценарий в новое стабильное состояние (в новую транзакцию сценария) в зависимости от имени текущей транзакции и ответа, полученного от реактивного агента.

$$g: T \times ANS \rightarrow QUE \quad (3.6)$$

обозначает функцию выхода, детерминирующую имя очередного вопроса в зависимости от имени текущей транзакции и текущего ответа реактивного агента.

Функции перехода и выхода соответствуют классическому определению автомата Мили. При моделировании метода доступа к памяти вопросов более естественной является функция транзакции. Поэтому в дальнейшем вместо двух отображений f и g будем пользоваться одним

$$\varphi: T \times ANS \rightarrow T \times QUE. \quad (3.7)$$

Функция транзакции детерминирует пару: имя очередного вопроса и имя следующей транзакции в зависимости от текущего ответа и имени текущей транзакции.

При выполнении транзакции (переходе из предыдущего стабильного состояния в последующее) автомат

$$DiAM = (QUE, ANS, T, \varphi) \quad (3.8)$$

выполняет следующую последовательность действий:

- воспринимает текущий ответ;
- распознает текущий ответ;
- детерминирует имя следующего вопроса.

Если никак не структурировать множество ANS , то это означает, что для распознавания текущего ответа $DiAM$ должен при выполнении каждой транзакции оперировать всеми элементами множества ANS . Для обширных сценариев мощность этого множества может быть значительной.

В текущей транзакции активный агент ожидает не любой, произвольный ответ, из множества ANS , а только некоторое количество ожидаемых ответов, которые детерминируются субъектом и требованием текущего вопроса. Поэтому для того, чтобы уменьшить мощность множества ответов, с которым на каждой транзакции оперирует $DiAM$, введем в модель множество ответов, распознаваемых на i -й транзакции. Это множество содержит только те

$$Ans \in ANS,$$

которые ожидаются на i -й транзакции, необходимы для реализации диалогового метода и, следовательно, должны быть распознаны. Все остальные $Ans \in ANS$ отнесём к классу ответов, не распознаваемых на i -й транзакции. Введем обозначения

R^i — множество ответов, распознаваемых на i -й транзакции;

NR^i — множество ответов, не распознаваемых на i -й транзакции.

К множеству не распознаваемых на i -й транзакции ответов, безусловно, относятся все ответы, принадлежащие множеству ANS и не принадлежащие множеству R^i , т. е.

$$NR^i = ANS - R^i. \quad (3.9)$$

Множество NR^i можно рассматривать и как более обширное, например, как любые мыслимые ответы реактивного агента, не принадлежащие множеству R^i , т. е.

$$NR^i = U - R^i, \quad (3.10)$$

где U универсум ответов, т. е. множество ответов, которые может породить реактивный агент. Ясно, что $ANS \subseteq U$.

Естественно предположить, что появление любого ответа из множества NR^i приводит к порождению автоматом $DiAM$ имени одного того же вопроса. Ибо в противном случае это означало бы, что ответы относятся к множеству распознаваемых. Поэтому в функции транзакции φ для каждой транзакции все множество нерасознаваемых ответов можно моделировать только одним нерасознаваемым ответом.

Проведенные рассуждения позволяют уточнить структуру набора ожидаемых ответов в когнитивном цикле диалогового процесса, рассмотренного в 2.1 и изображенного на рис. 2.2 и считать, что на каждой диалоговой транзакции набор ответов, которые ожидает активный агент, может включать: (1) класс распознаваемых ответов и (2) класс нерасознаваемых ответов (моделируемый одним ответом).

Рассмотрим пример, иллюстрирующий конечно-автоматную модель метода доступа к памяти вопросов. Пример преследует две цели: во-первых, попробовать оценить, насколько предложенная конечно-автоматная модель полезна при конструировании диалогового метода доступа к памяти вопросов, и, во-вторых, обеспечить переход к последующей сетевой модели диалогового метода доступа. Пример включает четыре транзакции и отражает логику доступа к последующему вопросу активного агента как реакцию на распознаваемые или нераспознаваемые ответы реактивного агента.

На транзакции 1 активный агент формирует вопрос с именем Que_1 и ожидает получить следующие ответы реактивного агента:

- Ans_1 — распознаваемый ответ, означающий запрос на получение твёрдой копии текущего вопроса. После выполнения запроса отсутствует переход к следующей транзакции.
- Ans_3 — распознаваемый ответ, означающий продолжение диалога, переход к транзакции 3 и формирование вопроса с именем Que_3 .
- Ans_4 — распознаваемый ответ, означающий продолжение диалога, переход к транзакции 4 и формирование вопроса с именем Que_4 .
- Ans_5 — нераспознаваемый ответ (любой ответ, отличный от Ans_1, Ans_3, Ans_4). Получение этого ответа означает, что необходимо продолжить диалог, перейти к транзакции 2 и сформировать вопрос с именем Que_2 .

На транзакции 2 активный агент формирует вопрос с именем Que_2 , представляющий собой переформулированный вопрос с именем Que_1 . На транзакции 2 активный агент ожидает получить тот же набор ответов, что и на транзакции 1, а также ответ Ans_2 , являющийся запросом на возвращение на транзакцию 1 и повторение вопроса с именем Que_1 .

Метод доступа к памяти вопросов для описанного диалога может быть описан следующим автоматом:

$$QUE = \{Que_1, Que_2, Que_3, Que_4\}, \quad (3.11)$$

$$ANS = \{Ans_1, Ans_2, Ans_3, Ans_4, Ans_5\}, \quad (3.12)$$

$$T = \{t_1, t_2, t_3, t_4\} \quad (3.13)$$

$$\left. \begin{array}{l} \varphi(t_1, Ans_1) \rightarrow (t_1, Que_1) \\ (t_1, Ans_3) \rightarrow (t_3, Que_3) \\ (t_1, Ans_4) \rightarrow (t_4, Que_4) \\ (t_1, Ans_5) \rightarrow (t_2, Que_2) \\ (t_2, Ans_1) \rightarrow (t_2, Que_2) \\ (t_2, Ans_2) \rightarrow (t_1, Que_1) \\ (t_2, Ans_3) \rightarrow (t_3, Que_3) \\ (t_2, Ans_4) \rightarrow (t_4, Que_4) \end{array} \right\} \quad (3.14)$$

Отображение φ , или функция транзакции, представленная формулами (3.14), с одной стороны, описывает логику доступа к памяти вопросов, и в этом смысле может рассматриваться как модель метода доступа к памяти вопросов, а с другой — логику диалогового метода решения некоторой задачи.

Отображение φ целесообразно представить в табличной форме, поскольку табличное представление можно использовать в инструментальной программе конструирования сценариев эротетических диалогов. На экране монитора практичнее заполнять таблицу, чем писать формулы. В табл. 3.1 содержится та же информация, что и в формулах (3.14).

Таблица 3.1

**Табличное представление функции транзакции
для примера, описанного в тексте**

Имя текущей транзакции	Ответ	Имя вопроса	Имя следующей транзакции
t_1	Ans_1	Que_1	t_1
	Ans_3	Que_3	t_3
	Ans_4	Que_4	t_4
	Ans_5	Que_2	t_2
t_2	Ans_1	Que_2	t_2
	Ans_2	Que_1	t_1
	Ans_3	Que_3	t_3
	Ans_4	Que_4	t_4
	Ans_5	Que_2	t_2

Табличное представление функции транзакции удобно также тем, что позволяет легко перейти к его графическому представлению в форме диаграммы состояний. На рис. 3.1 приведено графическое представление функции транзакции в виде диаграммы состояний. В диаграмме состояний, приведенной на рис. 3.1, множество вершин соответствует множеству транзакций, а каждая ветвь — одному из возможных вариантов развития диалога, определяемого ответом реактивного агента. Ветви помечены парами

$$Ans_i, Que_j,$$

где Ans_i — один из ответов, ожидаемых на текущей транзакции;

Que_j — имя вопроса, формируемого на текущей транзакции.

Как было отмечено ранее, функция транзакции может рассматриваться как способ описания логики диалогового метода решения задачи, и в этом

смысле диаграмма состояний, пример которой приведен на рис. 3.1, может рассматриваться как диаграмма, моделирующая целенаправленное поведение активного агента.

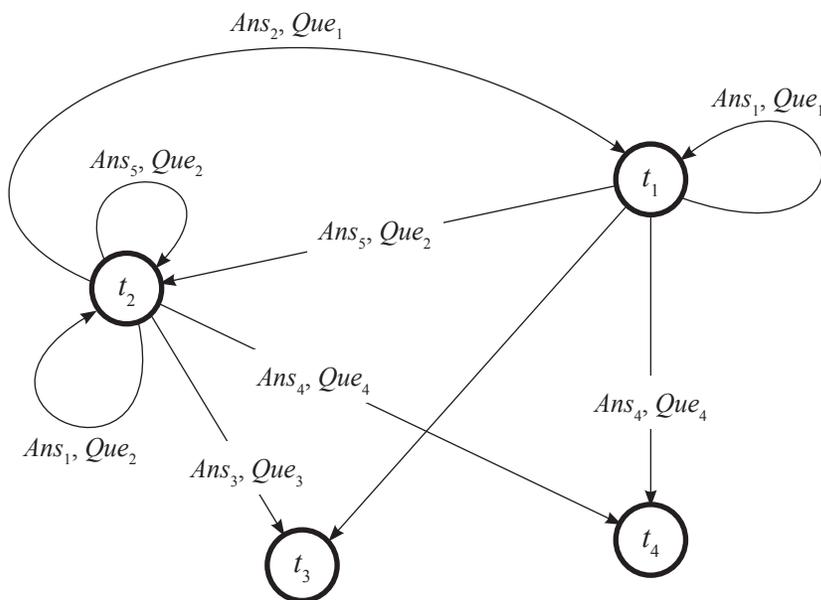


Рис. 3.1. Диаграмма состояний диалогового метода доступа для примера, описанного в тексте

3.2. Петри-модель

Автоматная модель, приведенная в 3.1, имеет свою сферу применимости и может использоваться для специфицирования сценариев некоторых прикладных диалоговых систем. Так, например, при помощи этой модели легко описывается метод внутреннего программирования Краудера [105], используемый при проектировании систем программированного обучения. Однако модель обладает недостатком, заключающимся в сложности её отображения в программный код.

Эффективным подходом к компьютерной реализации диалоговых систем является даталогический подход (data-driven approach). Даталогический подход противопоставляется алгоритмическому подходу и для случая программирования диалогового метода доступа предполагает отображение функции транзакции не в программный код, а в базу данных [106–110]. Решающим преимуществом даталогического подхода является возможность легкой модификации

диалогового метода доступа при помощи полноэкранных средств редактирования данных. Таким образом, при диалоговом подходе сценарий диалогового приложения представляет собой базу данных, при синтезе сценария исключается процесс компиляции, а синтез сценария может осуществляться силами его автора, который, как правило, не обладает специальными знаниями в области программирования.

С позиций диалогового подхода, формализм, принятый для описания диалогового метода доступа, необходимо отождествлять с концептуальной моделью некоторой базы данных. Как отмечено в [111], проблематика концептуальной модели баз данных связана с таким представлением модели, которое, с одной стороны, наиболее естественно отражает предметную область (в нашем случае диалоговый метод доступа к памяти вопросов), а, с другой, — может поддерживаться компьютерными средствами. Иными словами, необходим формализм, который легко трансформируется в схему данных. Опыт проектирования сценариев диалоговых приложений позволяет сделать вывод, что отмеченным свойством в большей степени обладает не диаграмма состояний диалогового метода доступа, пример которой приведен на рис. 3.1, а граф, моделирующий метод доступа в виде сети Петри [112]. Петри-модель метода доступа к памяти вопросов, так же, как и автоматная модель, предполагает выполнение двух условий: (1) цель диалогового процесса может быть достигнута за конечное количество транзакций, и к началу решения задачи детерминированы все вопросы и ответы, необходимые для формирования транзакций; (2) модификация набора ожидаемых ответов в любом случае порождает набор, отличающийся от текущего.

Петри-модель диалогового метода доступа опишем следующей четверкой:

$$DiAM = (QUE, ANS, NextQue, NextAns). \quad (3.15)$$

В формуле (3.15)

$$QUE = \{Que_i\}; i = 1, \dots, n \quad (3.16)$$

обозначает комплект имен всех вопросов активного агента, которые, согласно условию (1), достаточны для достижения цели диалогового процесса. Комплект (3.16) составлен из элементов множества (3.2) и представляет собой набор, в котором, в отличие от множества, допускает присутствие нескольких экземпляров одного и того же элемента [112]. Если $\#(x, B)$ — функция, которая определяет количество экземпляров элемента x в комплекте B , то мощность комплекта имен вопросов определяется формулой $n = |QUE| = \sum \#(Que_i, QUE)$. Поскольку каждая транзакция сценария включает один вопрос, который может повторяться в различных транзакциях, то число n определяет количество транзакций в сценарии.

$$ANS = \{Ans_i\}; i = 1, \dots, p \quad (3.17)$$

обозначает комплект ожидаемых ответов реактивного агента, составленный из элементов множества (3.3). Мощность комплекта ответов определяется формулой $p = |ANS| = \sum \#(Ans_p, ANS)$. Комплекты имен вопросов и ответов не пересекаются, т. е. имеет место $QUE \cap ANS = 0$.

$$NextQue : ANS \rightarrow QUE \quad (3.18)$$

обозначает функцию имен последующих вопросов, которая задает отображение комплекта ожидаемых ответов в комплект имен вопросов. Специфика межличностного диалога, заключающаяся в том, что активный агент на каждой транзакции формирует и передает реактивному агенту только один вопрос, накладывает на функцию $NextQue$ следующее ограничение. Для каждого ответа из комплекта ожидаемых ответов функция имен последующих вопросов детерминирует имя одного-единственного вопроса из комплекта вопросов.

$$NextAns : QUE \rightarrow ANS \quad (3.19)$$

обозначает функцию последующих ответов, которая задает отображение комплекта имен вопросов в комплект ожидаемых ответов.

Для иллюстрации введенных формул представим пример, описанный в 3.1 в виде Петри-модели.

$$QUE = \{Que_1, Que_2, Que_3, Que_4\}, \quad (3.20)$$

$$ANS = \{Ans_1, Ans_2, Ans_3, Ans_4, Ans_5\}, \quad (3.21)$$

$$\left. \begin{aligned} NextAns(Que_1) &= \{Ans_1, Ans_3, Ans_4, Ans_5\}, \\ NextAns(Que_2) &= \{Ans_1, Ans_2, Ans_3, Ans_4, Ans_5\}, \\ NextAns(Que_3) &= \{\}, \\ NextAns(Que_4) &= \{\}, \end{aligned} \right\} \quad (3.22)$$

$$\left. \begin{aligned} NextQue(Ans_1) &= \{Que_1\}, \\ NextQue(Ans_2) &= \{Que_1\}, \\ NextQue(Ans_3) &= \{Que_3\}, \\ NextQue(Ans_4) &= \{Que_4\}, \\ NextQue(Ans_5) &= \{Que_2\} \end{aligned} \right\} \quad (3.23)$$

Формулы (3.22) и (3.23) можно интерпретировать как табличное задание функций $NextAns$ и $NextQue$ соответственно.

Правая часть i -й строки функции $NextAns$ задает множество ответов реактивного агента, ожидаемых на i -й транзакции

$$NextAns(Que_i) = R^i \cup NR^i. \quad (3.24)$$

Правая часть i -й строки функции $NextQue$ состоит из одного элемента комплекта QUE , что является следствием отмеченного ограничения межличностного

диалога, заключающегося в том, что для каждого ответа из комплекта ожидаемых ответов функция имен последующих вопросов детерминирует имя одного-единственного вопроса из комплекта вопросов.

На рис. 3.2 приведен граф Петри-модели диалогового метода доступа, соответствующий рассмотренному примеру. При графической интерпретации Петри-модели диалогового метода доступа комплект имен вопросов *QUE* и комплект ответов *ANS* представлены множествами позиций (кружки) и переходов (прямоугольники) соответственно. Множество ожидаемых на *i*-ом шаге ответов $R^i \cup NR^i$ представлено совокупностью переходов инцидентных *i*-й позиции и соединённых с ней исходящими ветвями.

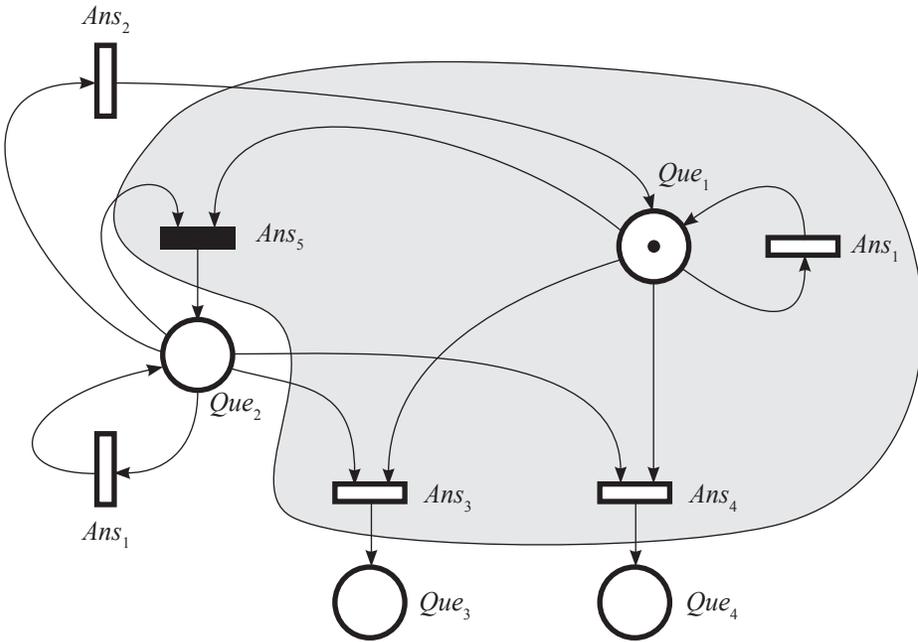


Рис. 3.2. Граф Петри-модели диалогового метода доступа для примера, описанного в тексте. Выделены элементы сети, входящие в объем понятия «транзакция сценария». Зачернённый переход моделирует множество нераспознаваемых ответов

Для Петри-модели транзакция сценария является естественным «строительным блоком» диалогового метода доступа, которая и для *i*-й шага может быть определена как совокупность

$$Que_i, (R^i \cup NR^i). \tag{3.25}$$

При графической интерпретации Петри-модели транзакция сценария — это некоторая позиция и совокупность переходов, соединённых с ней исходящими

ветвями. На рис. 3.2 выделены элементы, объединённые понятием «транзакция сценария».

Граф на рис. 3.2 маркирован единственной фишкой, отмечающей текущую транзакцию. Начальная разметка в этом случае помечает первую транзакцию сценария, с которого начинается реализация диалогового метода. Условием запуска перехода является совпадение ответа, кодируемого данным переходом, и ответа, полученного от реактивного агента. Иными словами, переход запускается и «пропускает» фишку, если от реактивного агента получен ответ, совпадающий с ответом, кодируемым данным переходом.

С точки зрения существующей классификации, сеть Петри, граф которой приведен на рис. 3.2, является ординарной (т. е. все дуги имеют одну и ту же кратность, равную единице) и не автоматной, по крайней мере, согласно определениям, приведенным в [113, 114], поскольку предлагаемая Петри-модель допускает несколько входов для одного и того же перехода.

Сети Петри предоставляют удобные средства для отслеживания динамики моделируемого диалогового процесса в виде маркеров позиций и условий запуска переходов. Поскольку в рассматриваемой Петри-модели имеется единственная фишка, то маркировка может быть представлена вектором, количество компонентов которого равно количеству элементов комплекта имен вопросов или количеству транзакций сценария.

$$t = (t_1, t_2, \dots, t_n). \quad (3.26)$$

Каждый компонент вектора маркировки t принимает значения на двухэлементном множестве $\{0, 1\}$

$t_i = 0$ в том случае, если фишка отсутствует в i -ой позиции, соответствующей i -й транзакции или i -му элементу комплекта *QUE*;

$t_i = 1$ в том случае, если фишка находится в i -ой позиции, соответствующей i -й транзакции или i -му элементу комплекта *QUE*.

Диалоговый процесс сопровождается перемещением фишки по сети без её уничтожения или размножения. Следовательно, в любой момент диалога имеет место $\sum t_i = 1$.

Диалоговый метод доступа, таким образом, характеризуется непрерывно изменяющимся вектором t , который указывает на текущую активную транзакцию. Для сети, приведенной на рис. 3.2, вектор t указывает на транзакцию 1 и имеет следующее значение $t = (1, 0, 0, 0)$.

Для того, чтобы фишка переместилась в последующую позицию, необходимо, чтобы запустился один из переходов транзакции.

Представление диалогового метода доступа в виде сети Петри моделирует его пространственную структуру и не отражает динамики диалогового процесса, задаваемой, в общем случае, когнитивным циклом диалогового процесса, изображенным на рис. 2.2. Дополним статическое описание метода доступа

компонентами, моделирующими его динамику. С этой целью рассмотрим, каким образом в рамках предложенной Петри-модели реализуется когнитивный цикл диалогового процесса. Моделирование цикла диалогового процесса будем рассматривать, исходя из предположения, что функции *NextAns* и *NextQue* представлены таблично.

Отдельная строка таблицы функции *NextAns* соответствует одной транзакции или одной итерации диалогового цикла. Каждая строка этой таблицы включает имя вопроса, ассоциированного с данной транзакцией, за которым следует множество ответов, ожидаемых на данной транзакции. Таким образом, таблица функции *NextAns* состоит из n строк, где n — количество элементов комплекта имен вопросов, которое совпадает с количеством транзакций сценария. Поскольку множество ответов, ожидаемых на данной транзакции, индивидуально для каждой транзакции, то часть ячеек в таблице функции *NextAns* остается незаполненной.

Таблица функции *NextQue* имеет более простую структуру. Она состоит из p строк, где p — количество элементов комплекта ответов, а каждая строка таблицы функции *NextQue* содержит два элемента: ожидаемый ответ и связанный с ним последующий вопрос.

Введем в модель унифицированный процесс, моделирующий когнитивный диалоговый цикл Петри-модели, обозначим его аббревиатурой *DiMC* (*Dialogue Method Cycle*) и представим функцией

$$DiMC(t, NextAns, NextQue). \quad (3.27)$$

DiMC управляет переходом от предыдущей транзакции к последующей, что для Петри-модели эквивалентно открытию одного из переходов транзакции и перемещению фишки. Унифицированность процесса *DiMC* означает, во-первых, что для выполнения любой транзакции необходимо выполнить одну и ту же последовательность действий, а во-вторых, что эта последовательность действий не зависит от предметной области диалогового процесса. На содержательном уровне цикл *DiMC* может быть описан следующим образом.

1. При помощи вектора маркировки t определяется текущая транзакция сценария (номер строки в таблице функции *NextAns*).
2. Определяется и передаётся в память вопросов имя текущего вопроса. Активный агент формирует текущий вопрос, передаёт его реактивному агенту и переходит в состояние ожидания ответа реактивного агента.
3. Активный агент воспринимает текущий ответ реактивного агента.
4. Обрабатывается текущий ответ реактивного агента. Последовательно сравнивается воспринятый ответ со всеми, ожидаемыми на данной транзакции ответами, хранящимися в ячейках выбранной строки таблицы

функции *NextAns*. Для совпавшего ответа (например, *Ans_j*) определяется строка таблицы функции *NextQue*.

- При помощи таблицы функции *NextQue* определяется компонент вектора *t*, который изменяет нулевое значение на единичное, и процесс возвращается к пункту 1.

На рис. 3.3 показана последовательность взаимодействия вектора маркировки *t* и таблиц функций *NextAns* и *NextQue* при выполнении *DiMC*.

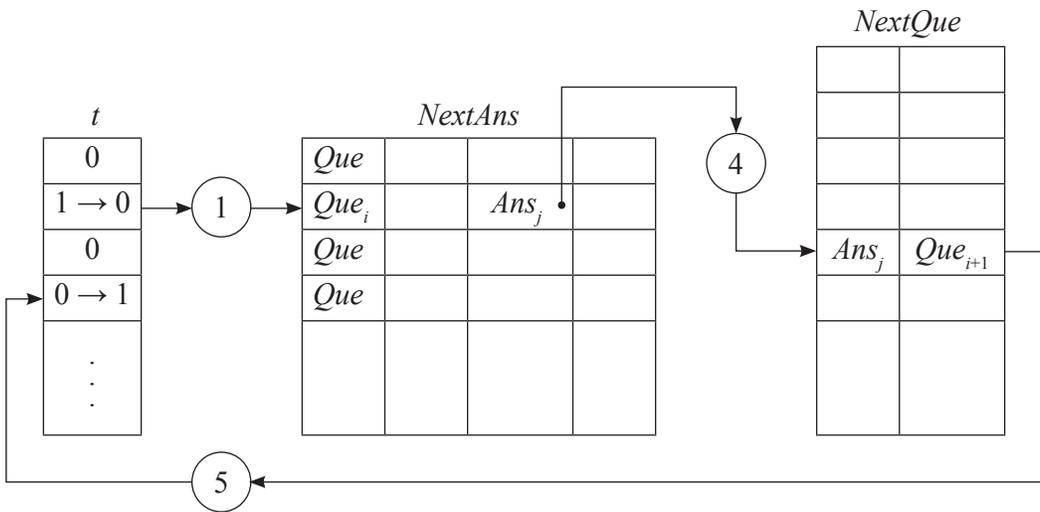


Рис. 3.3. Использование вектора маркировки *t* и таблиц функций *NextAns* и *NextQue* процессом *DiMC*. Номера в кружках соответствуют описанию *DiMC*, приведенному в тексте

Среди действий, выполняемых *DiMC*, есть два действия, требующие более детального рассмотрения и уточнения. Это действия, при помощи которых активный агент осуществляет формирование вопроса и обработку ответа. Поскольку *DiMC* — это унифицированный процесс, который «единообразно» обрабатывает любую транзакцию, то, как формирование вопроса, так и обработка ответа должны быть стандартизованы. Иными словами, *DiMC* должен уметь неким стандартным способом генерировать любой вопрос и обрабатывать любой ответ. Ясно, что даже при достаточно обширном стандарте могут встретиться случаи, когда стандартных средств недостаточно. Например, при генерации вопроса, в том случае, когда он представлен невербально, невозможно учесть все многообразие невербальной презентации информации. Необходимость в нестандартных средствах становится особенно очевидной, когда при обработке ответа необходимо выполнить специфическую численную обработку. Поэтому, кроме базового процесса *DiMC*, в модель целесообразно ввести «внешние», по отношению к *DiMC*, процессы. Эти процессы должны вызываться из *DiMC*

и возвращать ему управление после завершения. Назовем такие процессы процессы-демоны [115] и введём в модель два типа процессов-демонов

QueDemon и *AnsDemon*,

которые могут осуществлять «дообработку» вопроса или ответа на любой транзакции в том случае, когда стандартных средств *DiMC* недостаточно.

3.2.1. Применимость Петри-модели

Рассмотрим сферу применимости предложенной Петри-модели. Под применимостью модели будем понимать разделение (не строгое) задач, решаемых в ходе диалогового процесса, на два класса: легко и сложно реализуемых в рамках этой модели.

Рассмотрим сложнореализуемую, в рамках Петри-модели, задачу, относящуюся к области психодиагностики. Как правило, на первом этапе психодиагностики (в процессе диалогового взаимодействия с субъектом диагностики) формируется первичный набор данных, используемый затем для формирования заключения об интеллектуальных способностях или структуре интеллекта индивида. Примером может служить методика прогрессивных матриц Равена [116], используемая для тестирования интеллектуальных способностей. Отметим, что методика прогрессивных матриц Равена на каждой транзакции задаёт субъекту тестирования вопросы в невербальной форме. Особенностью диалогового процесса с психодиагностическими тестами является то, что последовательность вопросов, передаваемых реактивному агенту (субъект тестирования), формируется активным агентом (методика тестирования) практически без учёта последовательности ответов реактивного агента.

Будем считать для упрощения выкладок, что на первой транзакции активный агент формирует и передаёт реактивному агенту вопрос Que_1 , на который ожидает один из трёх ответов: Ans_1 , Ans_2 и Ans_3 . На второй транзакции, вне зависимости от ранее полученного ответа, активный агент формирует и передаёт реактивному агенту вопрос Que_2 , на который также ожидает один из трёх ответов: Ans_4 , Ans_5 и Ans_6 .

Решение задачи (заключение) формируется в зависимости от комбинации ответов, полученных при реализации диалогового процесса. На рис. 3.4 приведена сеть Петри для первых двух транзакций описанного примера.

На рис. 3.4 нераспознаваемые ответы реактивного агента опущены, чтобы не загромождать рисунок. В рассматриваемом примере диалогового процесса, графическая модель которого представима в виде дерева, сложность реализации диалогового метода проявляется в «комбинаторном взрыве» количества транзакций или в резком увеличении количества транзакций диалога в зависимости

от количества ожидаемых ответов для каждого из вопросов и общего количества вопросов в тесте (глубина дерева).

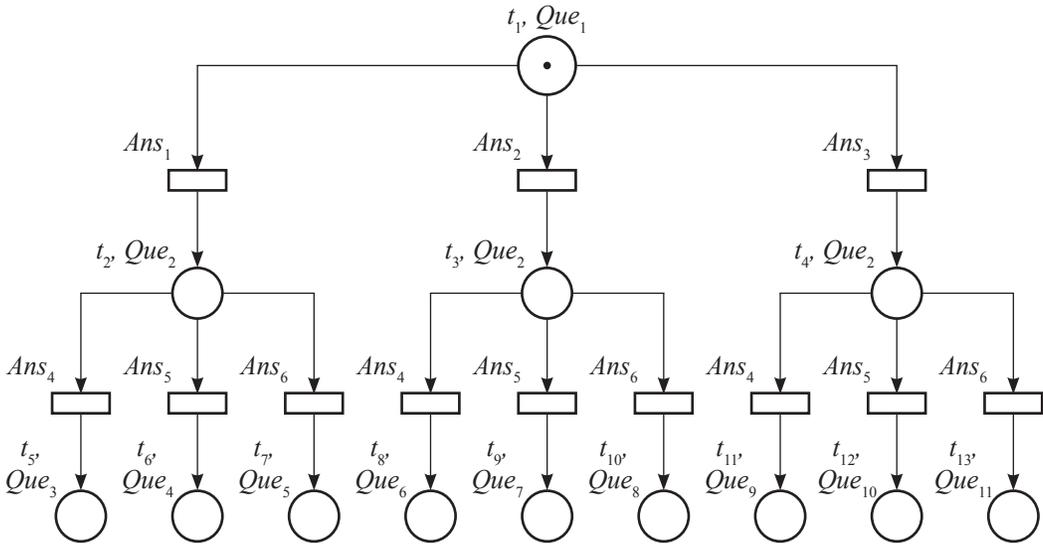


Рис. 3.4. Граф Петри-модели сложнореализуемого диалогового метода для примера, описанного в тексте

Для случая, приведенного на рис. 3.4, количество транзакций сценария может быть подсчитано по формуле:

$$n = 3^0 + 3^1 + 3^2. \quad (3.28)$$

В общем случае, если на каждый вопрос ожидается одинаковое количество ответов α , т. е. для каждой транзакции i имеет место

$$|R^i \cup NR^i| = const = \alpha, \quad (3.29)$$

где $|R^i \cup NR^i|$ — мощность множества ответов, ожидаемых на i -й транзакции, то

$$n = \alpha^0 + \alpha^1 + \dots + \alpha^d + \alpha^{(d+1)} - 1, \quad (3.30)$$

где d — общее количество вопросов в тесте (глубина дерева сетевой модели).

Комбинаторный взрыв при моделировании диалоговой методики прогрессивных матриц Равена средствами Петри-модели иллюстрируется количеством транзакций, необходимых для построения полного сценария. Методика включает 60-матриц (вопросов), на каждый из которых ожидается только два ответа (правильный и неправильный), следовательно, общее количество транзакций сценария может быть посчитано по формуле (3.30) и определяется числом $2^{61} - 1$.

Причина, породившая древовидную форму графа Петри-модели и, как следствие, приведшая к комбинаторному взрыву, заключается в том, что модель позволяет учесть комбинацию ответов, необходимую для получения заключения и достижения цели диалога, только одним способом — формированием в сети уникальных траекторий прохождения фишки.

Кроме комбинаторного взрыва, другим наблюдаемым эффектом сложнореализуемых задач является избыточность описания диалогового метода. Проиллюстрируем это на только что приведенном примере.

$$QUE = \{Que_1, Que_2, Que_2, Que_2, Que_3, Que_4, Que_5, Que_6, Que_7, Que_8, Que_9, Que_{10}, Que_{11}\} \quad (3.31)$$

$$ANS = \{Ans_1, Ans_2, Ans_3, Ans_4, Ans_4, Ans_4, Ans_5, Ans_5, Ans_5, Ans_6, Ans_6, Ans_6\} \quad (3.32)$$

$$\left. \begin{aligned} t_1 \text{ NextAns}(Que_1) &= \{Ans_1, Ans_2, Ans_3\} \\ t_2 \text{ NextAns}(Que_2) &= \{Ans_4, Ans_5, Ans_6\} \\ t_3 \text{ NextAns}(Que_2) &= \{Ans_4, Ans_5, Ans_6\} \\ t_4 \text{ NextAns}(Que_2) &= \{Ans_4, Ans_5, Ans_6\} \\ t_5 \text{ NextAns}(Que_3) &= \{\} \\ t_6 \text{ NextAns}(Que_4) &= \{\} \\ \dots \\ t_{13} \text{ NextAns}(Que_{11}) &= \{\} \end{aligned} \right\} \quad (3.33)$$

$$\left. \begin{aligned} t_2 \text{ NextQue}(Ans_1) &= \{Que_2\} \\ t_3 \text{ NextQue}(Ans_2) &= \{Que_2\} \\ t_4 \text{ NextQue}(Ans_3) &= \{Que_2\} \\ t_5 \text{ NextQue}(Ans_4) &= \{Que_3\} \\ t_6 \text{ NextQue}(Ans_5) &= \{Que_4\} \\ t_7 \text{ NextQue}(Ans_6) &= \{Que_5\} \\ t_8 \text{ NextQue}(Ans_4) &= \{Que_6\} \\ t_9 \text{ NextQue}(Ans_5) &= \{Que_7\} \\ t_{10} \text{ NextQue}(Ans_6) &= \{Que_8\} \\ t_{11} \text{ NextQue}(Ans_4) &= \{Que_9\} \\ t_{12} \text{ NextQue}(Ans_5) &= \{Que_{10}\} \\ t_{13} \text{ NextQue}(Ans_6) &= \{Que_{11}\} \end{aligned} \right\} \quad (3.34)$$

Строки функции *NextAns* и *NextQue* в формулах (3.33, 3.34) проиндексированы компонентами вектора маркировки.

Как видно из выражений (3.31)–(3.34), модель содержит фрагменты избыточного описания. Избыточность описания проявляется в повторяющихся строках в функции *NextQue* и в повторяющихся аргументах функции *NextAns*. Отмеченную избыточность в представлении диалогового метода средствами

Петри-модели будем называть концептуальной избыточностью. Ясно, что концептуальная избыточность модели является негативным свойством, с точки зрения её практического применения, поскольку приводит к избыточности хранимых данных при даталогической реализации модели и исключает применение диалоговых методов при решении задач, приводящих к комбинаторному взрыву.

Одним из способов борьбы с концептуальной избыточностью является введение в модель памяти, хранящей «историю ответов». Наличие такой памяти означает учет сформулированного в разделе 2.1 принципа истории ответов, который означает, что диалоговый метод при формировании вопроса учитывает как только что воспринятый ответ, так и ранее воспринятые ответы.

Рассмотрим влияние памяти истории ответов на концептуальную избыточность Петри-модели на примере рассмотренной диалоговой методики прогрессивных матриц Равена, иллюстрируемой рис. 3.4 и формулами (3.31)–(3.34). Для упрощения изложения будем считать, что модель включает память *Mem*, хранящую только один ответ, полученный на последней транзакции, а содержимое *Mem* автоматически обновляется при переходе к следующей транзакции. Процесс *DiMC* при определении имени следующего вопроса учитывает содержимое *Mem*.

На рис. 3.5 приведен граф Петри-модели диалогового метода для рассматриваемого примера с учетом памяти предыдущего ответа *Mem*.

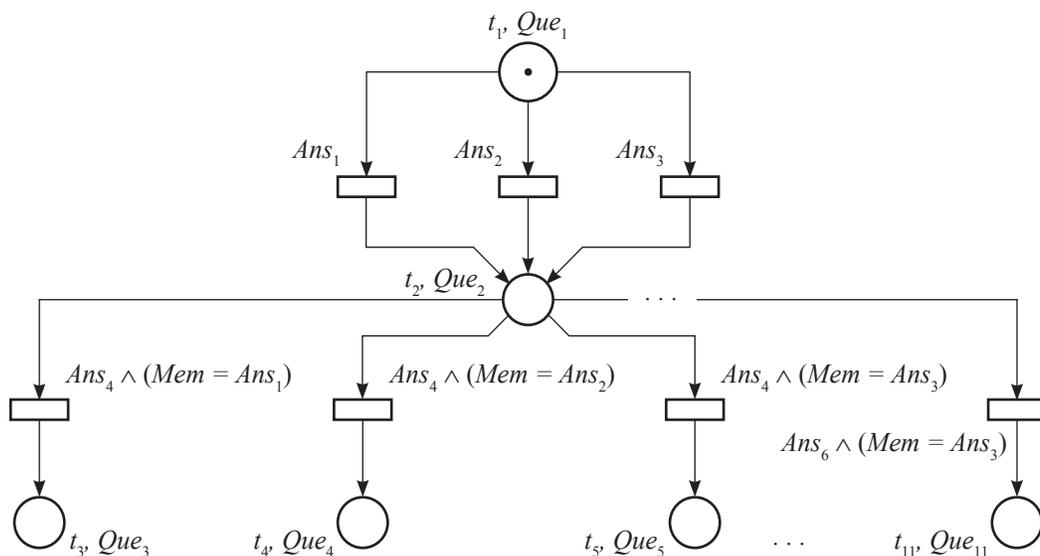


Рис. 3.5. Граф Петри-модели сложнореализуемого диалогового метода при наличии памяти предыдущего ответа *Mem*

Как видно на рис. 3.5, включение в модель памяти предыдущего ответа не требует формирования уникальной траектории для каждого заключения. Уникальными остаются целевые позиции, а траектории вывода имеют общую часть.

Представим теперь рассматриваемую модель аналитически. Для этого переишем выражения (3.31)–(2.34) с учётом того, что условие срабатывания перехода включает содержимое памяти предыдущего ответа и в общем случае формулируется следующим образом: переход срабатывает, если на текущем шаге получен один из ожидаемых ответов и на предыдущем шаге также получен ожидаемый ответ.

$$QUE = \{Que_1, Que_2, Que_3, Que_4, Que_5, Que_6, Que_7, Que_8, Que_9, Que_{10}, Que_{11}\} \quad (3.35)$$

$$ANS = \left. \begin{aligned} &\{Ans_1, Ans_2, Ans_3, \\ &Ans_4 \wedge (Mem = Ans_1), \\ &Ans_4 \wedge (Mem = Ans_2), \\ &Ans_4 \wedge (Mem = Ans_3), \\ &Ans_5 \wedge (Mem = Ans_1), \\ &Ans_5 \wedge (Mem = Ans_2), \\ &Ans_5 \wedge (Mem = Ans_3), \\ &Ans_6 \wedge (Mem = Ans_1), \\ &Ans_6 \wedge (Mem = Ans_2), \\ &Ans_6 \wedge (Mem = Ans_3)\} \end{aligned} \right\} \quad (3.36)$$

$$\left. \begin{aligned} t_1 \text{NextAns}(Que_1) &= \{Ans_1, Ans_2, Ans_3\} \\ t_2 \text{NextAns}(Que_2) &= \{Ans_4 \wedge (Mem = Ans_1) \\ &Ans_4 \wedge (Mem = Ans_2) \\ &Ans_4 \wedge (Mem = Ans_3) \\ &\dots \\ &Ans_6 \wedge (Mem = Ans_3)\} \\ t_3 \text{NextAns}(Que_3) &= \{\} \\ t_4 \text{NextAns}(Que_4) &= \{\} \\ &\dots \\ t_{11} \text{NextAns}(Que_{11}) &= \{\} \end{aligned} \right\} \quad (3.37)$$

$$\left. \begin{aligned} \text{NextQue}(Ans_1) &= \{Que_2\} \\ \text{NextQue}(Ans_2) &= \{Que_2\} \\ \text{NextQue}(Ans_3) &= \{Que_2\} \\ \text{NextQue}(Ans_4 \wedge (Mem = Ans_1)) &= \{Que_3\} \\ \text{NextQue}(Ans_4 \wedge (Mem = Ans_2)) &= \{Que_6\} \\ \text{NextQue}(Ans_4 \wedge (Mem = Ans_3)) &= \{Que_9\} \\ &\dots \\ \text{NextQue}(Ans_6 \wedge (Mem = Ans_3)) &= \{Que_{11}\} \end{aligned} \right\} \quad (3.38)$$

Как видно из формул (2.36)–(2.38), введение в модель памяти предыдущего ответа привело к полному устранению избыточности в комплекте *Que*, который, по сути, превратился в множество, состоящее из неповторяющихся элементов. Однако, при этом не изменилась мощность комплекта *Ans*, а среди ожидаемых ответов появились «сложные» в виде конъюнкции ожидаемых от реактивного агента ответов и содержимого *Mem*.

Таким образом, если предположить, что трудоёмкость реализации комплекта *Que* и распознавания ответов одинаковы, то «общая сложность» диалогового метода осталась без изменения. Она как бы перераспределяется внутри модели. Введение памяти предыдущего ответа упростило описание связей в сети, но усложнило процесс распознавания ответа.

3.3. Сетевая модель

Петри-модель, описанная в 3.2, имеет более широкую сферу применимости, чем автоматная модель, описанная в 3.1. Её преимущество в том, что она учитывает динамику диалогового процесса в виде унифицированного процесса *DiMC*, а также нестандартную обработку вопросов и ответов при помощи уникальных процессов *QueDemon* и *AnsDemon*. Петри-модель не исключает возможности учёта принципа истории ответов реактивного агента. Как показали проведенные исследования, сфера применимости Петри-модели ограничена задачами, диалоговый метод решения которых слабо чувствителен к истории ответов, поскольку в этом случае модель обнаруживает существенную концептуальную избыточность.

3.3.1. Структура модели

Сетевая модель диалогового метода доступа к памяти вопросов развивает главные идеи двух предыдущих моделей, но базируется не на математической структуре (конечный автомат или сети Петри), а на идее схемы базы данных [111]. В предлагаемой модели существенно следующее:

- учитываются два из трёх, сформулированных в разделе 2.1, принципов организации эротетического диалога: принцип глубины диалога и принцип истории ответов реактивного агента;
- развивается даталогический подход и а priori предполагается, что диалоговый метод доступа к памяти вопросов и спецификации вопросов активного агента, хранящиеся в памяти вопросов, являются данными и хранятся в некоторой базе данных;
- развивается сетевое понимание структуры диалогового метода.

Как и ранее, будем предполагать, что сценарий диалогового процесса представляется двумя компонентами: памятью вопросов и диалоговым методом доступа к памяти вопросов *DiAM*.

Будем считать, что *DiAM* имеет сетевую организацию и состоит из связанных между собой узловых элементов. Отдельный узел *DiAM* соответствует одной транзакции сценария. В каждый момент времени активным является только один узел. Определение следующей транзакции и активизацию соответствующего узла осуществляет унифицированный процесс *DiMC*. Связи между узловыми элементами являются направленными и для каждого узла определяют множество возможных последующих транзакций.

Будем считать, что сеть *DiAM* неоднородна в том смысле, что она строится из узлов разных типов. Узлы отличаются назначением, описанием и правилом навигации. *Идея неоднородности сети DiAM является принципиальной* и означает, во-первых, что сеть состоит из разнотипных узловых элементов, а во-вторых, что возможна различная типология узловых элементов. В предлагаемой модели вводится три типа узловых элементов [117].

- Узел прямой навигации (узел типа *Node1*). Это узел, который соответствует транзакции, находящейся внутри сценария и предполагающей продолжение диалога. Для узла типа *Node1* имя следующей транзакции определяется только с учётом принципа глубины.
- Узел условной навигации (узел типа *Node2*). Это узел, который также соответствует транзакции, предполагающей продолжение диалога. Для узла типа *Node2* имя следующей транзакции определяется с учётом принципа глубины и принципа истории ответов.
- Целевой узел (узел типа *Node3*). Это узел, которым завершается диалоговый процесс.

Для каждого узла типов *Node1* и *Node2* существует своё правило навигации, при помощи которого определяется имя следующей транзакции. На рис. 3.6 изображены узлы типов *Node1* и *Node2* и соответствующие им правила навигации. Как видно на рис. 3.6, для узла типа *Node1* каждому ожидаемому на данной транзакции ответу соответствует только одна последующая транзакция с именем *NextNodeInd*, а все нераспознаваемые на данной транзакции ответы моделируются одной связью с последующей транзакцией *NNodeInd*.

Навигация для узла типа *Node1* осуществляется относительно просто. Текущий ответ реактивного агента рассматривается как «закодированное имя» следующей транзакции. Поэтому процесс его распознавания (либо классификации как нераспознанного) рассматривается как процесс табличной перекодировки воспринятого ответа в *NextNodeInd* или *NNodeInd*

Навигация для узла типа *Node2* осуществляется следующим образом. Воспринятый текущий ответ последовательно сравнивается со всеми ожидаемыми ответами, ассоциированными с текущей транзакцией. Если воспринятый

текущий ответ совпал с одним из ожидаемых, то проверяется содержимое памяти одной из предыдущих транзакций. Если на предыдущей транзакции воспринят заданный ответ, то формируется *NextNodeInd*. Если воспринятый текущий ответ не совпал ни с одним из ожидаемых, то формируется *NNodeInd*.

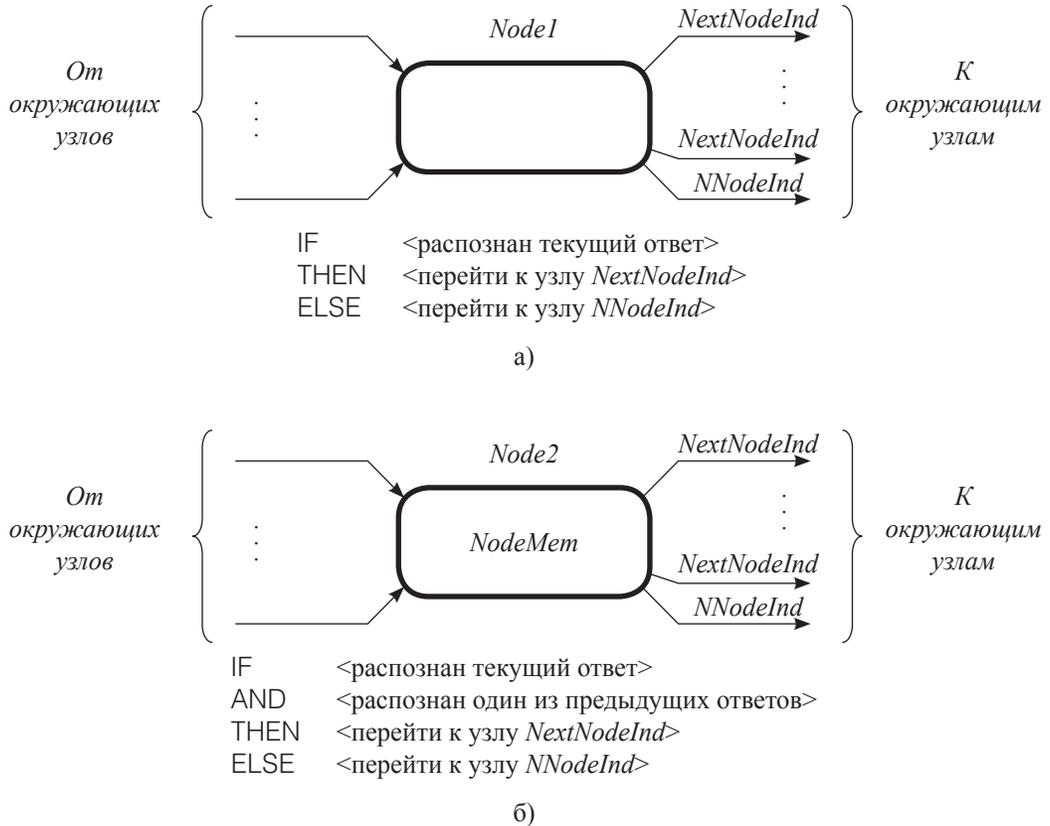


Рис. 3.6. Узловые элементы сетевой модели с прямой навигацией (a) и условной навигацией (b)

Таким образом, правило навигации для узла типа *Node2* предполагает наличие в каждом узле памяти, сохраняющей воспринятый ответ реактивного агента. Обозначим эту память *NodeMem*.

Как было отмечено ранее, введенная классификация узлов не является единственной. При практической реализации модели полезна, например, классификация, учитывающая тип вопроса, ассоциированного с транзакцией (узлом). Важно, что сетевая модель диалогового метода позволяет вводить сколь угодно сложную классификацию узлов, в отличие, например, от Петри-модели, который, являясь двудольным графом, предполагает наличие только двух типов вершин.

Сеть *DiAM* будем отображать в схему реляционной базы данных. Для того, чтобы такое отображение было возможным, необходимо даталогическое представление каждого из введенных типов узлов. На рис. 3.7 в графическом виде приведено даталогическое представление узла типа *Node1*.

На рис. 3.7 использованы следующие обозначения:

TranInd — имя транзакции;

NodeType — тип узла, соответствующий данной транзакции;

QueInd — имя вопроса, ассоциированного с данной транзакцией;

QueDemonInd — имя процесса-демона, обслуживающего вопрос данной транзакции;

CurrAns[1] ... CurrAns[r] — текущие ожидаемые ответы;

AnsDemonInd — имя процесса-демона, обслуживающего ответы данной транзакции.

NextTranInd — имя следующей транзакции для ожидаемого ответа.

NNTranInd — имя следующей транзакции для неожиданного ответа.

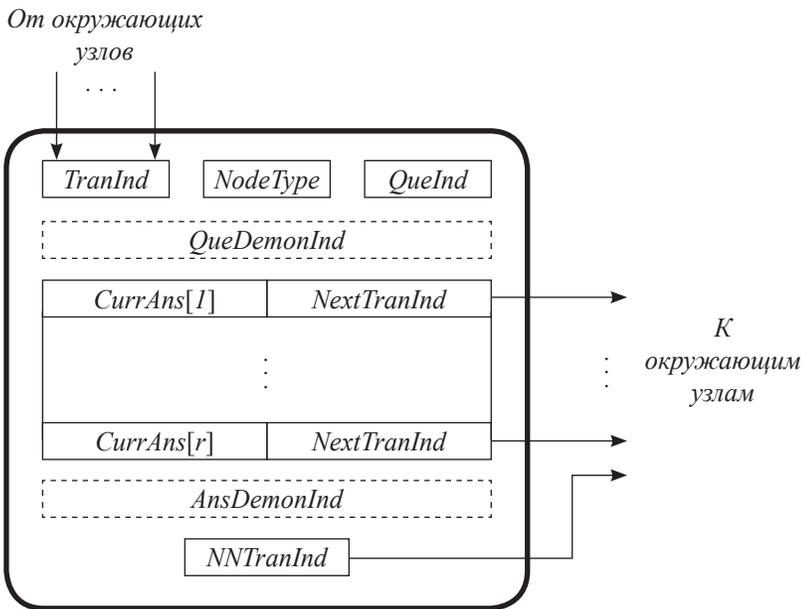


Рис. 3.7. Даталогическое представление узла типа *Node1*. Пунктиром отмечены необязательные описатели

Даталогическое описание узла типа *Node1*, приведенное на рис. 3.7, интерпретируется унифицированным процессом *DiMC*, работа которого может быть прервана одним из процессов-демонов *QueDemon* или *AnsDemon*. Ранее, в 3.2, цикл процесса *DiMC* был описан в терминах Петри-модели.

Для случая сетевой модели цикл *DiMC* представим в виде следующих, последовательно выполняемых, трёх фаз.

Фаза генерации вопроса. В памяти вопросов отыскивается описание вопроса с именем *QueInd*, он интерпретируется и выводится через видео- и аудиоканалы компьютера. Если описание узла включает *QueDemonInd*, то этот процесс загружается и выполняется. После завершения работы процесса *QueDemonInd* управление возвращается *DiMC*.

Фаза восприятия ответа. Воспринимается текущий ответ реактивного агента. Если описание узла включает *AnsDemonInd*, то этот процесс загружается и выполняется. После завершения работы процесса *AnsDemonInd* управление возвращается *DiMC*.

Фаза навигации. Определяется индекс следующего узлового элемента. С учетом введенных обозначений, правило навигации для узла типа *Node1* можно записать в виде

```

IF      <распознан CurrAns[i]>
THEN    <перейти к узлу NextNodeInd>
ELSE                                         (3.39)
IF      <не распознан CurrAns[i]>
THEN    <перейти к узлу NNodeInd>

```

Идея учёта истории ответов для узлов с условной навигацией предполагает учёт одного из ответов воспринятого на одной из предыдущих транзакций. Это означает, что имя следующей транзакции для каждого распознанного текущего ответа должно определяться при условии, что на одной из предшествующих транзакций был воспринят конкретный ответ.

На рис. 3.8 приведено даталогическое представление узла типа *Node2* с учетом принятого ограничения.

На рис. 3.8 введены следующие дополнительные обозначения.

CurrAns[1], ..., *CurrAns*[*r*] — ответы, ожидаемые на текущей транзакции;

NodeMem[*TranInd*] — ответ, воспринятый на предыдущей транзакции *TranInd*.

Интерпретация описания узла типа *Node2* унифицированным процессом *DiMC* отличается от интерпретации узла типа *Node1* только фазой навигации. С учётом введенных обозначений, правило навигации для узла типа *Node2* представим в виде

```

IF      <распознан CurrAns[i]>
AND     <распознано содержимое NodeMem[TranInd]>
THEN    <перейти к узлу NextNodeInd[i, j]>
ELSE                                         (3.40)
IF      <не распознан CurrAns[i]>
THEN    <перейти к узлу NNodeInd>

```

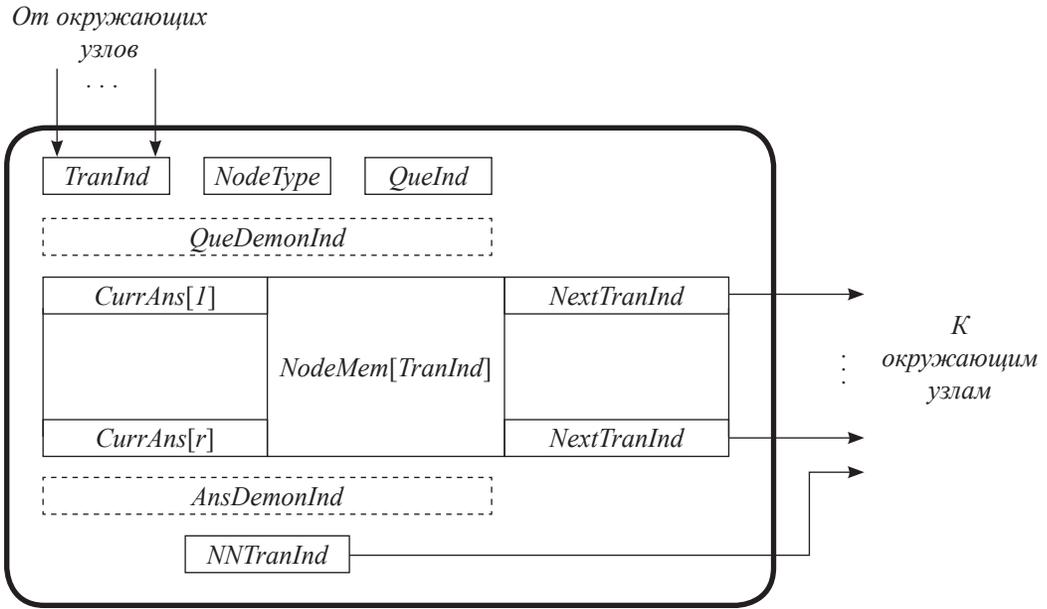


Рис. 3.8. Даталогическое представление узла типа *Node2*.
Пунктиром отмечены необязательные описатели

На рис. 3.9 приведено даталогическое представление узла типа *Node3*. Поскольку узел типа *Node3* является целевым и переход на этот узел эквивалентен завершению работы диалогового метода, то, во-первых, его описание не предполагает перехода к последующим узлам, а, во-вторых, *QueInd* является именем утверждения, а не вопроса.

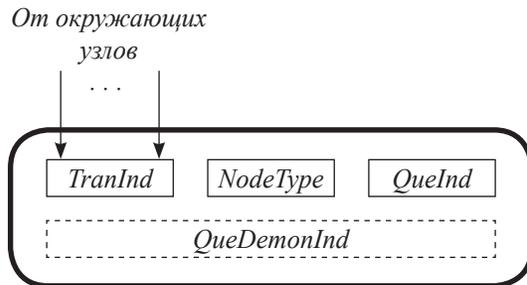


Рис. 3.9. Даталогическое представление узла типа *Node3*

При практической реализации диалогового сценария с методом доступа на основе сетевой модели, достижение узла типа *Node3* означает переход из

DiMC в программную среду более высокого уровня, например, в среду операционной системы.

3.3.2. Аналитическое представление модели

Опишем сеть *DiAM*, состоящую из узловых элементов типа *Node1*, *Node2* и *Node3*, в виде схемы абстрактной реляционной базы данных, используя нотацию, предложенную в [118].

Задача логического проектирования базы данных является, по сути, задачей информационного моделирования и не имеет единственного решения. Задача информационного моделирования сети *DiAM* заключается в описании сети, состоящей из узловых элементов типов *Node1*, *Node2* и *Node3* в виде набора отношений. Ясно, что можно предложить несколько вариантов такого описания. Приведенная база не претендует на оптимальность, а, скорее, иллюстрирует конструктивность подхода.

Будем моделировать сеть *DiAM* следующими пятью отношениями.

$$DiAM = \{Tran, Navig1, Navig2, QueDemon, AnsDemon\} \quad (3.41)$$

где *Tran* — отношение, моделирующее множество транзакций, связанных сетью *DiAM*;

Navig1 — отношение, моделирующее транзакции с узлами типа *Node1* и правилом навигации, описываемым формулой (3.39);

Navig2 — отношение, моделирующее транзакции с узлами типа *Node2* и правилом навигации, описываемым формулой (3.40);

QueDemon — отношение, моделирующее множество процессов-демонов, обслуживающих вопросы;

AnsDemon — отношение, моделирующее множество процессов-демонов, обслуживающих ответы.

Представим каждое из отношений базы (3.41) в виде набора атрибутов. В приведенных ниже формулах (3.42)–(3.46) атрибуты-ключи выделены подчеркиванием, а выражение $dom(Attr)=\{\dots\}$ специфицирует домен соответствующего атрибута.

$$Tran = \{\underline{TranInd}, NodeType, QueInd, NTranInd\} \quad (3.42)$$

Атрибут с именем *TranInd* моделирует имя транзакции. Имена транзакций различны. Домен этого атрибута состоит из *TranQ* имен, где *TranQ* — общее количество транзакций *DiAM*.

Атрибут с именем *NodeType* моделирует тип узлового элемента. Домен этого атрибута $dom(NodeType) = \{1, 2, 3\}$.

Атрибут с именем *QueInd* моделирует имя вопроса соответствующего транзакции. В различных транзакциях могут быть совпадающие имена вопросов.

Домен этого атрибута состоит из $QueQ$ имен, где $QueQ$ — общее количество различных вопросов в памяти вопросов $TranQ \geq QueQ$.

Атрибут с именем $NTranInd$ моделирует имя транзакции, на которую осуществляется переход для случая нераспознаваемого ответа. Домен этого атрибута, так же, как и домен атрибута $TranInd$, состоит из $TranQ$ имен.

Каждая из транзакций $DiAM$ описывается точно одним кортежем отношения $Tran$, поэтому количество кортежей отношения $Tran$ равно $TranQ$.

$$Navig1 = \{TranInd, CurrAns, NextTranInd\} \quad (3.43)$$

Атрибут с именем $TranInd$ в отношении $Navig1$ имеет точно такой же смысл, как и в отношении $Tran$.

Атрибут с именем $CurrAns$ моделирует распознаваемые ответы. Домен этого атрибута $dom(CurrAns) = \{CurrAns_1, \dots, CurrAns_r\}$, где r — наибольшее количество распознаваемых ответов для любой транзакции $DiAM$.

Атрибут с именем $NextTranInd$ моделирует имя транзакции, на которую осуществляется переход для случая распознаваемого ответа. Домен атрибута состоит из $TranQ$ имен.

Каждый распознаваемый ответ на каждой транзакции для узла с прямой навигацией описывается точно одним кортежем отношения $Navig1$. Количество кортежей отношения $Navig1$ не превышает числа $r * TranQ$.

$$Navig2 = \{TranInd, CurrAns, PrevTranInd, PrevAns, NextTranInd\} \quad (3.44)$$

Атрибуты с именами $TranInd$ и $CurrAns$ в отношении $Navig2$ имеют точно такой же смысл, как и в отношении $Navig1$.

Атрибут с именем $PrevTranInd$ моделирует имя предшествующей транзакции для узла с условной навигацией. Домен этого атрибута состоит из $TranQ$ имен, где $TranQ$ — общее количество транзакций $DiAM$.

Атрибут с именем $PrevAns$ моделирует ответ, воспринятый на предшествующей транзакции для узла с условной навигацией. Домен этого атрибута $dom(PrevAns) = \{PrevAns_1, \dots, PrevAns_r\}$, где r — наибольшее количество распознаваемых ответов для любой транзакции $DiAM$.

Атрибут с именем $NextTranInd$ для отношения $Navig2$ имеет такой же смысл, как и для отношения $Navig1$.

Каждый распознаваемый ответ на каждой транзакции для узла с условной навигацией, как и для узла с прямой навигацией, описывается одним кортежем отношения $Navig2$, поэтому количество кортежей отношения $Navig2$ не превышает числа $r * TranQ$.

$$QueDemon = \{TranInd, QueDemonInd\} \quad (3.45)$$

$$AnsDemon = \{TranInd, AnsDemonInd\} \quad (3.46)$$

Отношения *QueDemon* и *AnsDemon* имеют простую структуру. Атрибуты этих отношений ставят в соответствие каждому имени транзакции имя одного из процессов-демонов.

Используя нотацию унифицированного языка моделирования UML, можно аналитическое представление сетевой модели *DiAM* в виде формул (3.41)–(3.46) трансформировать в логическую модель базы данных *DiAM* в виде UML диаграммы классов. На рис. 3.10 приведена диаграмма классов, моделирующая логическую структуру реляционной базы данных сети *DiAM*.

Отметим несколько особенностей диаграммы классов, приведенной на рис. 3.10.

Первая особенность заключается в том, что графические символы классов не содержат отделений с описанием методов. Это связано с тем, что диаграмма классов используется для моделирования логической структуры реляционной, а не объектной базы данных. Однако, плодотворной может быть идея представления сетевой модели *DiAM* в виде объектной базы. Такое представление сетевой модели *DiAM* может объединить даталогический подход к представлению сети *DiAM*, унифицированный процесс *DiMC* и процессы-демоны. В объектной базе функции *DiMC* и процессов-демонов должны быть разделены на отдельные методы и распределены между всеми классами системы, изображенной на рис. 3.10.

Вторая особенность заключается в том, что два отношения типа «ассоциация», установленные между классом *Tran* и классами *Navig1* и *Navig2*, являются альтернативными, поскольку, как это следует из проведенных ранее рассуждений, для произвольной транзакции возможен только один тип навигации.

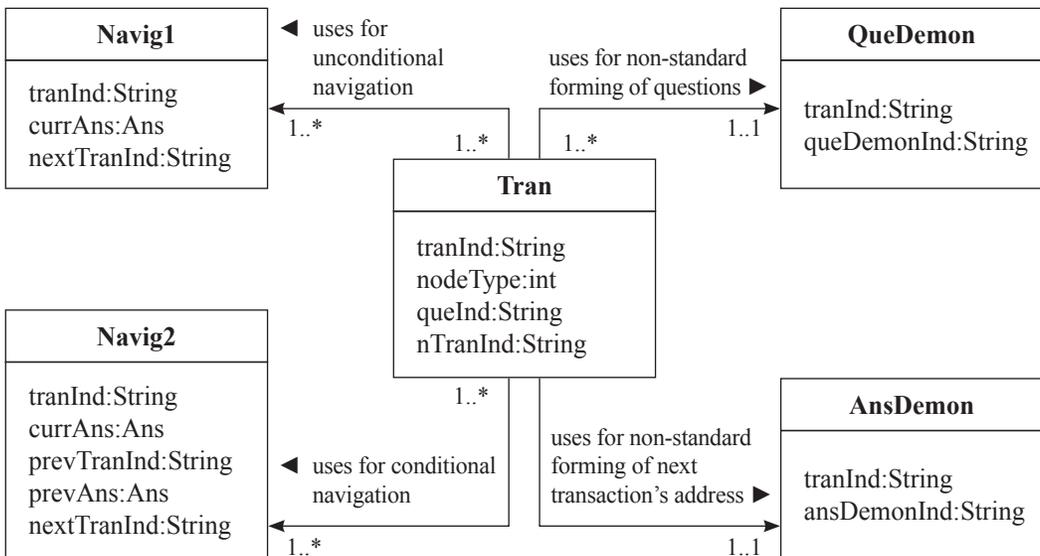


Рис. 3.10. Логическая структура реляционной базы данных сети *DiAM* в виде UML диаграммы классов

В 3.2 проведено исследование сферы применимости Петри-модели на примере сложнореализуемого диалогового метода, относящегося к проблематике психодиагностики. В процессе исследования диалоговый метод был вначале описан без использования памяти истории ответов (формулы (3.31)–(3.34), рис. 3.4), а затем с простейшей памятью истории ответов, хранящей только один ответ, полученный на последней транзакции (формулы (3.35)–(3.38), рис. 3.5). Проведём аналогичное исследование, используя полученную сетевую модель. Рассмотрим два варианта описания сложнореализуемого диалогового метода доступа к памяти вопросов: (1) *DiAM* состоит только из узлов типов *Node1* и *Node3*; (2) *DiAM* состоит из узлов всех трёх типов.

Напомним, что суть сложной реализуемости рассматриваемого примера заключается в следующем. Реактивному агенту передается вопрос Que_1 . На этот вопрос от реактивного агента ожидается один из трех возможных ответов: Ans_1 , Ans_2 , Ans_3 . Вне зависимости от полученного ответа, реактивному агенту передается вопрос Que_2 . На этот вопрос также ожидается три ответа: Ans_4 , Ans_5 , Ans_6 . На последней транзакции, в зависимости от комбинации ответов, полученных на оба вопроса, выбирается одно из девяти заключений. Будем считать, для определенности, что в том случае, когда на любой из вопросов получен нераспознаваемый ответ, то этот вопрос повторяется.

На рис. 3.11 приведено графическое изображение сети *DiAM*, которая состоит только из узлов типов *Node1* и *Node3*, а на рис. 3.12 — её модель в виде отношений *Tran* и *Navig1*. Оба отношения представлены в табличной форме, а для упрощения принято, что процессы-демоны не используются. Отсутствие процессов-демонов означает, что в сценарии используются только те вопросы и ответы, которые могут быть проинтерпретированы стандартным процессом *DiMC*.

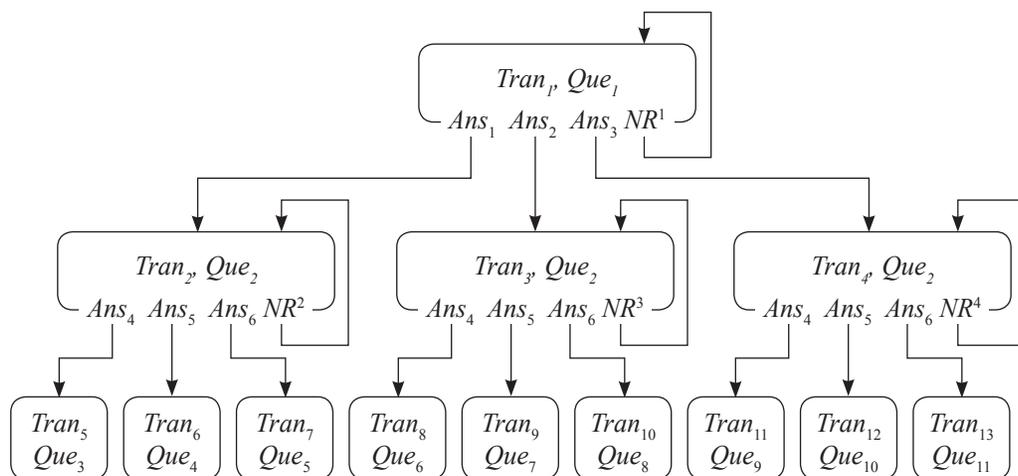


Рис. 3.11. Графическое представление сети *DiAM* для примера сложнореализуемого метода, моделируемого узлами *Node1* и *Node3*

<i>Tran</i>			
<i>TranInd</i>	<i>NodeType</i>	<i>QueInd</i>	<i>NTranInd</i>
1	<i>Node1</i>	1	1
2	<i>Node1</i>	2	2
3	<i>Node1</i>	2	3
4	<i>Node1</i>	2	4
5	<i>Node3</i>	3	<i>отсутствует</i>
6	<i>Node3</i>	4	<i>отсутствует</i>
7	<i>Node3</i>	5	<i>отсутствует</i>
8	<i>Node3</i>	6	<i>отсутствует</i>
9	<i>Node3</i>	7	<i>отсутствует</i>
10	<i>Node3</i>	8	<i>отсутствует</i>
11	<i>Node3</i>	9	<i>отсутствует</i>
12	<i>Node3</i>	10	<i>отсутствует</i>
13	<i>Node3</i>	11	<i>отсутствует</i>

<i>Navig1</i>		
<i>TranInd</i>	<i>CurrAns</i>	<i>NextTranInd</i>
1	<i>Ans₁</i>	2
1	<i>Ans₂</i>	3
1	<i>Ans₃</i>	4
2	<i>Ans₄</i>	5
2	<i>Ans₅</i>	6
2	<i>Ans₆</i>	7
3	<i>Ans₄</i>	8
3	<i>Ans₅</i>	9
3	<i>Ans₆</i>	10
4	<i>Ans₄</i>	11
4	<i>Ans₅</i>	12
4	<i>Ans₆</i>	13

Рис. 3.12. Реляционная модель сети *DiAM* для примера сложнореализуемого метода, моделируемого узлами *Node1*, и *Node3*

Предположим теперь, что один из узлов сети, а именно узел второй транзакции, является узлом с условной навигацией. На рис. 3.13 и 3.14 приведены графическое изображение сети *DiAM* и её модель в виде отношений *Tran*, *Navig1* и *Navig2* соответственно.

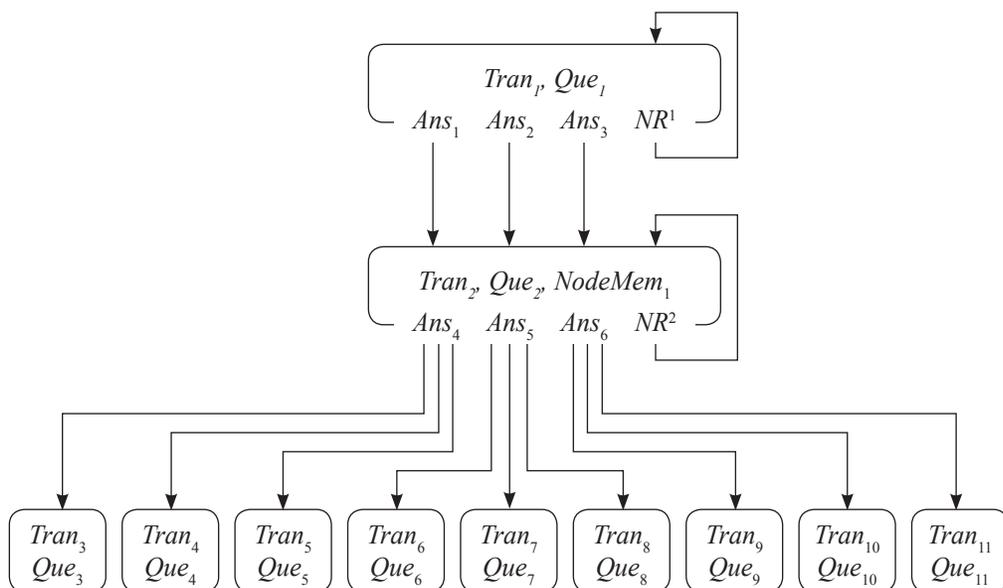


Рис. 3.13. Графическое представление сети *DiAM* для примера сложнореализуемого метода, моделируемого узлами *Node1*, *Node2* и *Node3*

Существует принципиальная возможность двойного описания одного и того же диалогового метода доступа как с использованием узлов с условной навигацией, так и без использования узлов этого типа.

Анализ моделей, приведенных на рис. 3.11–3.14, позволяет сделать следующие выводы. В том случае, когда сеть *DiAM* не содержит узлы типа *Node2*, в сети формируются уникальные, непересекающиеся траектории, соединяющие начальный и целевые узловы элементы. В том случае, когда сеть *DiAM* строится с использованием узлов типа *Node2*, общее количество узлов уменьшается, количество связей остаётся неизменным, а размер базы незначительно возрастает. Следовательно, при разработке диалогового метода решения задач, относящихся к классу психодиагностических тестов, учёт истории ответов и узлов с условной навигацией не упрощает базу данных.

Основным инструментом, позволяющим избежать явления комбинационного взрыва, является использование не памяти истории ответов, а процесса-демона *AnsDemon*, обрабатывающего ответы. Процесс *AnsDemon* позволяет упростить модель, например, следующим образом. Ответам приписываются баллы. Баллы суммируются, и, таким образом, вся история ответов кодируется одним числом. Таким образом, процедура логического вывода заменяется процедурой, которая ставит в соответствие полученной сумме баллов релевантное заключение.

<i>Tran</i>			
<i>TranInd</i>	<i>NodeType</i>	<i>QueInd</i>	<i>NextTranInd</i>
1	<i>Node1</i>	1	1
2	<i>Node2</i>	2	2
3	<i>Node3</i>	3	<i>отсутствует</i>
4	<i>Node3</i>	4	<i>отсутствует</i>
5	<i>Node3</i>	5	<i>отсутствует</i>
6	<i>Node3</i>	6	<i>отсутствует</i>
7	<i>Node3</i>	7	<i>отсутствует</i>
8	<i>Node3</i>	8	<i>отсутствует</i>
9	<i>Node3</i>	9	<i>отсутствует</i>
10	<i>Node3</i>	10	<i>отсутствует</i>
11	<i>Node3</i>	11	<i>отсутствует</i>

<i>Navig1</i>		
<i>TranInd</i>	<i>CurrAns</i>	<i>NextTranInd</i>
1	<i>Ans₁</i>	2
1	<i>Ans₂</i>	2
1	<i>Ans₃</i>	2

<i>Navig2</i>				
<i>TranInd</i>	<i>CurrAns</i>	<i>PrevTranInd</i>	<i>PrevAns</i>	<i>NextTranInd</i>
2	<i>Ans₄</i>	1	<i>Ans₁</i>	3
2	<i>Ans₄</i>	1	<i>Ans₂</i>	4
2	<i>Ans₄</i>	1	<i>Ans₃</i>	5
2	<i>Ans₅</i>	1	<i>Ans₁</i>	6
2	<i>Ans₅</i>	1	<i>Ans₂</i>	7
2	<i>Ans₅</i>	1	<i>Ans₃</i>	8
2	<i>Ans₆</i>	1	<i>Ans₁</i>	9
2	<i>Ans₆</i>	1	<i>Ans₂</i>	10
2	<i>Ans₆</i>	1	<i>Ans₃</i>	11

Рис. 3.14. Реляционная модель сети *DiAM* для примера сложнореализуемого метода, моделируемого узлами *Node1*, *Node2* и *Node3*

3.4. Проблема моделирования «неискренности» реактивного агента

В предложенном онтологическом базисе и моделях диалогового процесса отсутствует компонент, явно учитывающий такое антропоморфное свойство реального диалога, как неискренность реактивного агента. В реальном диалоге ответы реактивного агента не всегда правдивы, особенно в том случае, когда вопросы активного агента касаются нежелательных, запретных или опасных для реактивного агента тем. Как правило, одной из целей реактивного агента в ходе диалогового процесса является уклонение от таких тем и сокрытие информации.

Учет неискренности реактивного агента даёт возможность строить диалоговые методы, позволяющие достигнуть поставленной цели даже в том случае, когда части ответов нельзя доверять, и, следовательно, делает модель более адекватной естественным диалогам, наблюдаемым в реальной жизни.

Для учёта неискренности реактивного агента предлагается использование основных положений теории нечётких множеств [119, 120]. Прикладная ценность теории нечётких множеств заключается не столько в предлагаемом ею определении и формализме нечёткого множества, сколько в интерпретации функции степени принадлежности.

Как отмечается в [119], теория нечётких множеств является следующим (вслед за формальной логикой) шагом в борьбе с нечёткостью и расплывчатостью представления человеческих знаний. Авторы теории нечётких множеств предложили, в качестве основного её постулата, отказаться от принципа классической теории множеств, согласно которому элемент может либо принадлежать, либо не принадлежать множеству. Вместо этого для каждого множества вводится характеристическая функция, которая, будучи непрерывной и принимая значения на сегменте $[0, 1]$, характеризует степень принадлежности элемента множеству.

Таким образом, нечёткое множество определяется как

$$A = \{x, \mu_A(x)\}$$

и представляет собой совокупность упорядоченных пар элементов $x \in X$ и соответствующих им степеней принадлежности

$$\mu_A(x): [0, 1].$$

Трансформируем рассмотренную в 3.1 детерминированную автоматную модель *DiAM* в нечёткую, используя приведенное определение нечёткого множества.

Автоматная модель, описанная в 3.1, включает три детерминированных множества — потенциальных кандидатов на введение нечёткости. Это множество имен вопросов активного агента *QUE*, множество ответов реактивного агента *ANS* и множество стабильных состояний сценария диалога, или множество имен транзакций сценария *T*.

Стабильное состояние t_i представляет собой имя конкретной транзакции, а множество транзакций T является четким, поскольку бессмысленно говорить о степени принадлежности к имени.

В детерминированной модели множество QUE — это множество имен всех вопросов, достаточных для достижения цели диалога. С точки зрения модели, все вопросы Que_i являются равноценными и одинаково приближают диалог к его цели. Однако в естественном диалоговом процессе дело может обстоять иначе, и тот или иной вопрос может (в зависимости от степени своей прямоты) способствовать достижению цели диалога в большей или меньшей степени.

Следовательно, представляется целесообразным следующая интерпретация понятия нечёткости по отношению к множеству вопросов.

Определим множество вопросов QUE как нечёткое следующим образом:

$$QUE = \{Que_j, \mu_Q(Que_j)\} j = 1, \dots, k \quad (3.47)$$

$$\mu_Q: QUE \rightarrow [0, 1], \quad (3.48)$$

где $\mu_Q(Que_j)$ — функция прямоты вопроса Que_j . Численное значение функции прямоты вопроса отражает «близость» текущего вопроса к прямому вопросу.

Прямым вопросом будем называть вопрос, предполагающий в качестве ответа решение задачи. Таким образом, предполагается, что цель диалога может быть достигнута за одну транзакцию при помощи единственного прямого вопроса. Естественно, что при этом активный агент должен абсолютно доверять ответам реактивного агента. В противном случае активный агент при помощи серии непрямых вопросов накапливает некоторое количество фактов, используемых как «исходный материал» для вывода (или расчёта) искомого решения.

Интуитивно ясно, что чем менее прямые вопросы использует активный партнёр, тем дольше будет длиться диалоговый процесс, и наоборот. Ясно также, что одной и той же цели можно достичь за различное количество транзакций в зависимости от степени прямоты используемых вопросов. Поэтому функция прямоты вопроса μ_Q характеризует также «скорость», с которой диалоговый процесс приближается к цели.

Степень прямоты вопроса может трактоваться как «расстояние» между прямым и непрямым вопросами либо как «расстояние» до цели диалога. Таким образом, чем меньше активный агент доверяет ответам реактивного агента, тем менее прямые вопросы он задаёт и, следовательно, тем более длительным должен быть «путь к цели», или фактический диалоговый процесс. Если поведение реактивного агента неодинаково на различных участках диалога, то степень доверия к его ответам тоже различна. Активный агент может учитывать этот факт путем изменения степени прямоты своих вопросов.

В детерминированной модели множество ANS представляет собой множество всех ответов, достаточных для достижения цели диалога. Важной

характеристикой ответов реактивного агента является их правдивость. Активный агент, зная, насколько правдив тот или иной ответ, может варьировать степень прямоты своих вопросов.

Определим множество ответов ANS как нечеткое следующим образом:

$$ANS = \{Ans_j, \mu_A(Ans_j)\} \quad j = 1, \dots, l \quad (3.49)$$

$$\mu_A: ANS \rightarrow [0, 1], \quad (3.50)$$

где $\mu_A(Ans_j)$ — функция ожидаемой правдивости ответа Ans_j . Функция μ_A названа функцией ожидаемой правдивости ответа, поскольку в естественном диалоговом процессе часто сложно точно знать, насколько правдив полученный ответ, и активный агент может лишь оценивать эту величину.

Таким образом, нечёткая автоматная модель диалогового метода доступа описывается семёркой [121, 122]

$$DiAM = (QUE, ANS, T, \mu_Q, \mu_A, f, g), \quad (3.51)$$

где $QUE = \{Que_j, \mu_Q(Que_j)\} \quad j = 1, \dots, k$ — множество не прямых вопросов активного агента;

$ANS = \{Ans_j, \mu_A(Ans_j)\} \quad j = 1, \dots, l$ — множество неправдивых ответов реактивного агента;

$T = \{t_i\} \quad i = 1, \dots, m$ — множество транзакций диалогового сценария;

$\mu_Q: ANS \rightarrow [0, 1]$ — функция прямоты вопроса активного агента;

$\mu_A: ANS \rightarrow [0, 1]$ — функция ожидаемой правдивости ответа реактивного агента;

$f: T \times ANS \rightarrow T$ — функция перехода, переводящая $DiAM$ в новую транзакцию в зависимости от текущей транзакции и ответа, полученного от реактивного агента;

$g: T \times ANS \rightarrow QUE$ — функция выхода, определяющая имя вопроса активного агента в зависимости от текущей транзакции и ответа, полученного от реактивного агента;

Поскольку, как отмечалось выше, значения функций μ_Q и μ_A для соответственно предыдущего ответа и последующего вопроса не независимы, то необходимо ввести дополнительный формализм, учитывающий их корреляцию

$$\mu_A(Ans_j) = k * \mu_Q(Que_{j+1}), \quad k \approx 1. \quad (3.52)$$

Нечёткая автоматная модель диалогового метода доступа позволяет разрабатывать общие принципы или стратегии диалоговых методов решения задач. Приведем несколько примеров таких стратегий.

Стратегия 1. Активный агент строит своё поведение, исходя из предположения, что в ходе диалога степень ожидаемой правдивости ответов реактивного агента остается неизменной и, следовательно, степень правдивости его вопросов также остается неизменной.

Стратегия 2. Активный агент строит своё поведение, исходя из предположения, что в ходе диалога степень ожидаемой правдивости ответов реактивного агента изменяется. Тогда на каждом шаге связь между ожидаемой правдивостью ответа и степенью прямоты очередного вопроса определяется некоторой зависимостью. Например, линейной зависимостью вида (3.52) с коэффициентом, близким к единице. Чем меньше ожидаемая правдивость ответа, тем менее прямым должен быть очередной вопрос и наоборот.

4 ДАТАЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ СТРУКТУРЫ ВОПРОСА

В третьем разделе при разработке моделей диалогового метода доступа к памяти вопросов отмечались преимущества даталогического подхода при практической разработке диалоговых программ.

Даталогический подход к проектированию программ, в отличие от алгоритмического, предполагает, что большая часть предметной области «покрывается» базой данных, а не набором программных функций [123]. При даталогическом подходе сложность программы перераспределяется между сложностью организации базы данных и программными функциями в пользу базы данных. Это позволяет использовать простые и часто стандартизованные алгоритмы для интерпретации данных, хранящихся в базе [124].

Исследование эротетического диалога и разработка его моделей, приведенные в третьем разделе, по сути, были направлены на получение модели, ориентированной на даталогическое проектирование. Предложенная сетевая модель (см. подраздел 3.3) показывает, каким образом диалоговый метод доступа к памяти вопросов может быть представлен в виде реляционной базы данных.

В настоящем разделе развивается идея даталогического подхода в направлении моделирования информационных посылок диалоговых агентов для случая эротетического диалогового процесса. Идея даталогического моделирования транзакций эротетического диалога основана на убеждении в том, что декларативные знания, ассоциированные с эротетической транзакцией, есть данные.

Даталогическое представление вопроса будем называть сценой. Сцену будем моделировать при помощи цепи объектов, интерпретируемых через видео- и аудиоканалы компьютера. Таким образом, любой логический компонент эротетической транзакции (например, субъект или требование вопроса) моделируется некоторым участком цепи объектов сцены. На рис. 4.1 схематически представлены принятые соглашения о структуре сцены.

На рис. 4.1 сцена представлена в виде цепи, состоящей из пяти объектов, интерпретируемых через видео- и аудиоканалы компьютера. Объекты сцены

являются дatalogическими «строительными блоками», из которых строятся логические компоненты вопросно-ответной пары. Таким образом, как субъект, так и требование вопроса строятся из одного или нескольких, соединенных в цепь, объектов сцены. Объекты сцены проиндексированы в пределах каждой сцены, и их интерпретация осуществляется последовательно, начиная с первого.

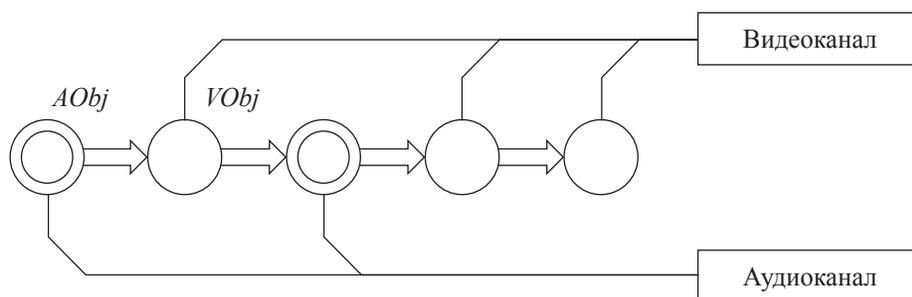


Рис. 4.1. Сцена, представленная цепью, составленной из видео объектов (*VObj*) и аудио объектов (*AObj*)

Поскольку для формирования одного и того же логического компонента вопроса (например, субъекта вопроса) могут использоваться и видео-, и аудио объекты (например, синхронный вывод звука и изображения), то работа видеоканала должна быть согласована с работой аудиоканала. Согласование каналов предполагает наличие набора правил, позволяющих правильно располагать объекты в цепи, составляющей сцену.

В зависимости от особенностей диалогового метода доступа изменяется распределение количества информации, передаваемой через видео- и аудиоканалы.

В том случае, когда реактивный агент обладает пониженным зрением, роль аудиоканала возрастает, а видеоканал играет вспомогательную роль. Роль аудиоканала становится решающей в случае, например, телефонного разговора с искусственным диалоговым агентом [125].

4.1. Атрибутивные и неатрибутивные описатели объекта сцены

Любой объект сцены будем описывать совокупностью атрибутов, необходимых и достаточных для его интерпретации через видео- или аудиоканал. Однако представление объекта сцены исключительно в виде набора атрибутов возможно лишь для простейших объектов. Для случая сложных объектов атрибутивное описание необходимо дополнять данными, которые невозможно или нерационально разлагать на атрибуты. Будем такое описание называть неатрибутивным.

Таким образом, примем, что любой объект сцены может быть представлен двумя типами описаний:

- атрибутивное описание;
- неатрибутивное описание.

В [126] отмечается, что неатрибутивное описание соответствует информации, поступающей непосредственно от прототипа, минуя восприятие и интеллектуальную обработку человека. Прототип — это некий, существующий вне системы, объект, на базе которого строится неатрибутивное описание объекта сцены. Например, фотографическое изображение или музыкальная мелодия.

Атрибутивное описание формируется в результате интеллектуальной деятельности человека. Это может быть либо анализ реального прототипа и его последующее разложение на атрибуты, либо ментальный синтез несуществующего в природе объекта и его представление в виде набора атрибутов.

Рассмотрим процесс формирования неатрибутивного и атрибутивного описаний видео объектов.

В случае формирования неатрибутивного описания, информация о прототипе видеообъекта передается в виде светового потока или в виде энергетических характеристик каждой из составляющих его точек. Для формирования неатрибутивного описания необходимо техническое устройство, преобразующее световой поток в набор данных, характеризующих цвет и яркость каждой точки. На рис. 4.2 приведена диаграмма, иллюстрирующая процесс формирования неатрибутивного описания и последующей интерпретации видеообъекта.



Рис. 4.2. Формирование и интерпретация неатрибутивного описания видеообъекта сцены

Для того, чтобы неатрибутивное описание имело конечный размер, необходимо, чтобы преобразователь выполнял дискретизацию светового потока по пространству, уровню и цвету. В этом случае преобразователь представляет собой известные устройства, например, сканер, цифровая фото- или видеокамера и т. д. Отметим, что полученное с помощью схемы, приведенной на рис. 4.2, неатрибутивное описание будет оставаться таковым, даже если перейти от

значений яркости и цвета к другим характеристикам, выполнив некоторые преобразования. Например, перейти от значений яркости и цвета к спектральной интенсивности [127]. В любом случае интерпретатор неатрибутивного описания должен выполнить обратное преобразование.

Ряд эффективных и практически пригодных методов формирования неатрибутивного описания объекта сцены предложены в [128]. Не рассматривая подробно методы, предложенные в [128] и в работах других авторов, будем считать, что в описании объекта сцены неатрибутивное описание представлено в виде цифрового файла соответствующего формата.

Атрибутивное описание формируется человеком в результате восприятия прототипа, его категоризации и последующей интеллектуальной обработки полученного ментального образа. Смысл интеллектуальной обработки заключается в упрощении ментального образа путем представления его в виде конечного набора атрибутов. На рис. 4.3 приведена диаграмма, иллюстрирующая процесс формирования атрибутивного описания и интерпретации видеообъекта сцены.



Рис. 4.3. Формирование и интерпретация атрибутивного описания видеообъекта сцены

Ясно, что задача формирования атрибутивного описания решается неоднозначно и определяется человеком, формирующим атрибутивное описание. Можно получить сколь угодно много различных атрибутивных описаний в зависимости от того, например, с какой целью формируется описание.

4.2. Классы атрибутов видеообъекта сцены

Рассмотрим пример реализации изложенных выше идей дatalogического моделирования сцены. Ниже описан один из способов атрибутивного описания видеообъекта, который использовался при практических разработках диалоговых приложений [107, 129].

Систему атрибутивного описания видеообъекта сцены можно рассматривать как некую систему классификации, согласованную с организацией интерпретирующей системы. Поэтому, прежде чем вводить какую-либо систему классификации для видеообъекта, необходимо принять общие соглашения об организации интерпретирующей системы. Будем считать, что интерпретирующая система строится, исходя из следующих соглашений.

1. Видеообъект представляет собой прямоугольник, который может иметь различные размеры и отображаться в различных участках экрана монитора. Видеообъекты могут перекрываться. Предполагается, что из различных прямоугольных видео объектов можно сконструировать достаточно сложную сцену.

2. После того, как процесс отображения видеообъекта на экран монитора завершён, он может подвергаться топологическим преобразованиям в виде поступательного перемещения, вращения и деформации.

3. Неатрибутивное описание выводится внутрь видео объекта.

4. Видеообъект является «окном», через которое просматривается только часть неатрибутивного описания.

5. Видеообъект предназначен для визуальной презентации субъекта и требования вопроса, а также для связи процессов-демонов со сценой в том случае, когда для интерпретации недостаточно стандартных средств унифицированного процесса *DiMC*.

6. Внешним процессам-демонам доступны память истории ответов и проблемная база данных. Проблемная база данных является информационной моделью специфики предметной области диалогового приложения. Взаимодействие *DiMC* с проблемной базой осуществляется опосредованно при помощи процессов-демонов.

На рис. 4.4 приведена диаграмма, иллюстрирующая перечисленные соглашения об организации интерпретирующей системы.

Компоненты, приведенные на рис. 4.4, соответствуют основным «хранилищам» информации и библиотеке внешних процессов интерпретирующей системы, а линии, которые их соединяют, — отношениям между ними. Отметим, что цепь видеообъектов в текущей сцене задаёт последовательность их интерпретации, а не размещение на экране монитора. Привязка к экрану задаётся соответствующими атрибутивными описателями. Из всех объектов, составляющих сцену, только один, интерпретируемый в данный момент, является активным, а остальные рассматриваются как фон. На рис. 4.4 активный видеообъект назван текущим. Имеется только один буфер, обслуживающий активный объект во время его интерпретации. Та часть буфера, которая может «просматриваться» через каркас объекта, задаётся соответствующими атрибутивными описателями.

Все ответы реактивного агента запоминаются в памяти истории ответов. В зависимости от того, как в сценарии описана транзакция, текущий ответ реактивного агента может либо непосредственно отображаться в имя следующей транзакции, либо опосредованно, при помощи внешнего процесса *AnsDemon*.

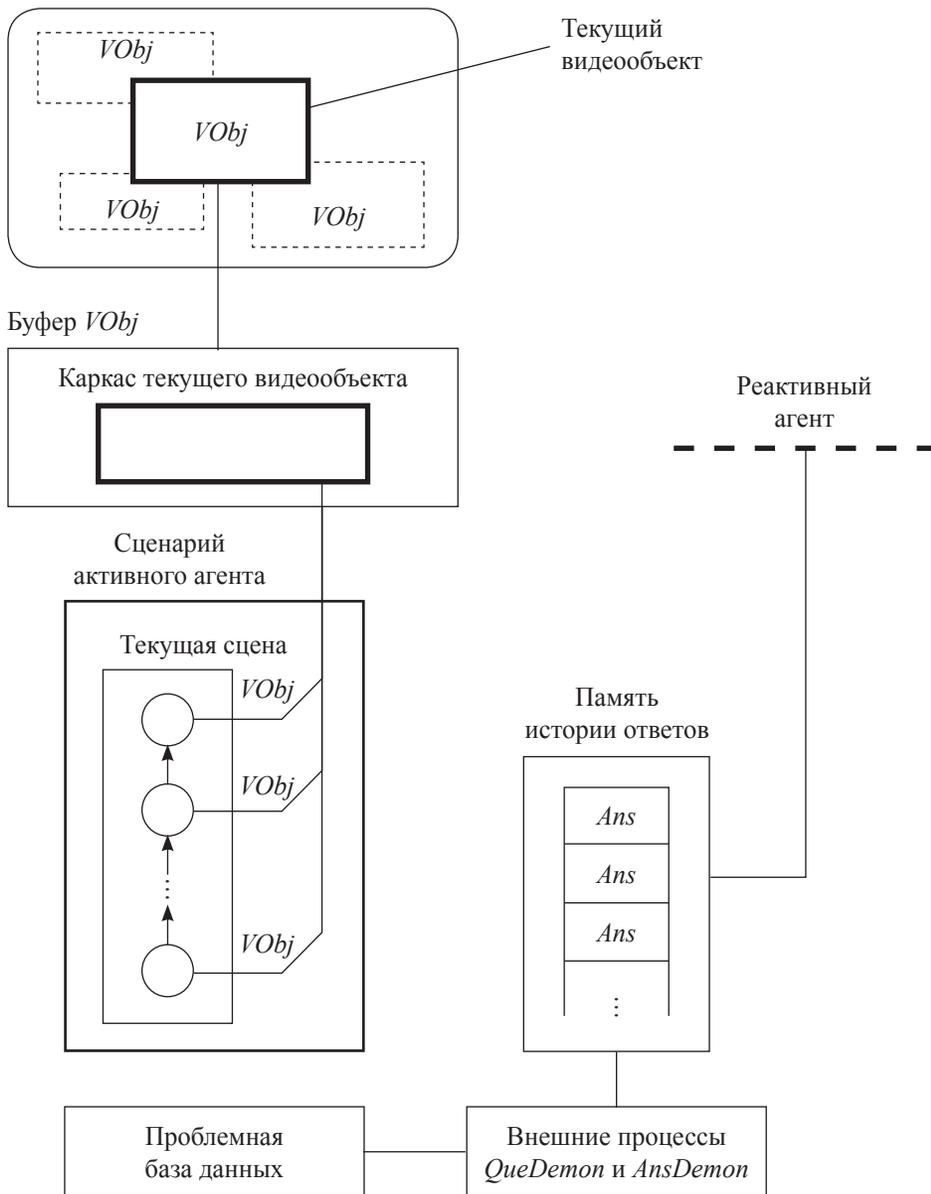


Рис. 4.4. Организация системы интерпретации сцены, состоящей из видеообъектов. Линии, соединяющие блоки, соответствуют не информационным потокам, а отношениям между компонентами системы

Предложенное понимание атрибутивного описания объекта сцены и принятые соглашения об организации интерпретирующей системы позволяют сформировать множество классов атрибутивного описания видеообъектов. В

рассматриваемом примере атрибутивного описания видеообъекта сцены принято, что любой видеообъект описывается при помощи семи классов атрибутов: (1) атрибуты каркаса; (2) атрибуты видеоэффектов; (3) атрибуты редактируемого текста; (4) атрибуты поступательного перемещения; (5) атрибуты деформации каркаса; (6) атрибуты источника информации; (7) атрибуты «времени жизни».

Ниже описан один из возможных вариантов реализации перечисленных выше семи классов атрибутов. Эти атрибуты применялись при практических разработках инструментального пакета программ «Процессор Диалога» [129] и ряда диалоговых приложений [130, 131]. При специфицировании базы данных все атрибуты получают соответствующий тип. Для того, чтобы не загромождать текст излишними деталями, в таблицы, описывающие атрибуты, включены только имена основных атрибутов соответствующего класса и опущены имена их типов.

Атрибуты каркаса. Каркасом видео объекта будем называть прямоугольную рамку с указанием ширины, высоты и координаты базовой точки на экране монитора. В качестве базовой точки выбран левый верхний угол прямоугольника. Рис. 4.5 иллюстрирует идею каркаса, а в табл. 4.1 описаны основные атрибуты каркаса видеообъекта.

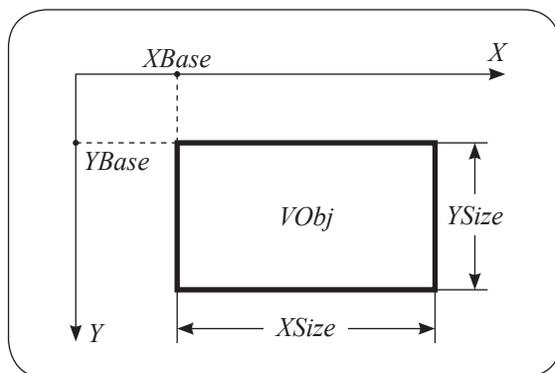


Рис. 4.5. Каркас видеообъекта

Таблица 4.1

Атрибуты каркаса видеообъекта

№ п/п	Наименование	Интерпретация
1	<i>XBase</i>	Горизонтальная координата базовой точки
2	<i>YBase</i>	Вертикальная координата базовой точки
3	<i>XSize</i>	Размер по горизонтали
4	<i>YSize</i>	Размер по вертикали

Кроме атрибутов, приведенных в таблице 4.1, атрибуты каркаса могут включать еще несколько атрибутов, описывающих линию рамки, например, ее толщину и цвет.

Атрибуты видеоэффектов. Атрибуты видеоэффектов описывают различные способы презентации информации внутри каркаса. Путем вариации этих атрибутов можно получить различные динамические эффекты внутри каркаса, которые, как показывает опыт, при их правильном использовании могут ускорить восприятие и понимание как самой сцены, так и её отдельных компонентов. Рис. 4.6 иллюстрирует действие атрибутов видеоэффектов. Каркас позволяет «просматривать» буфер, содержащий неатрибутивное описание (смотри рис. 4.4), которое может быть не только изображением, но и текстом.

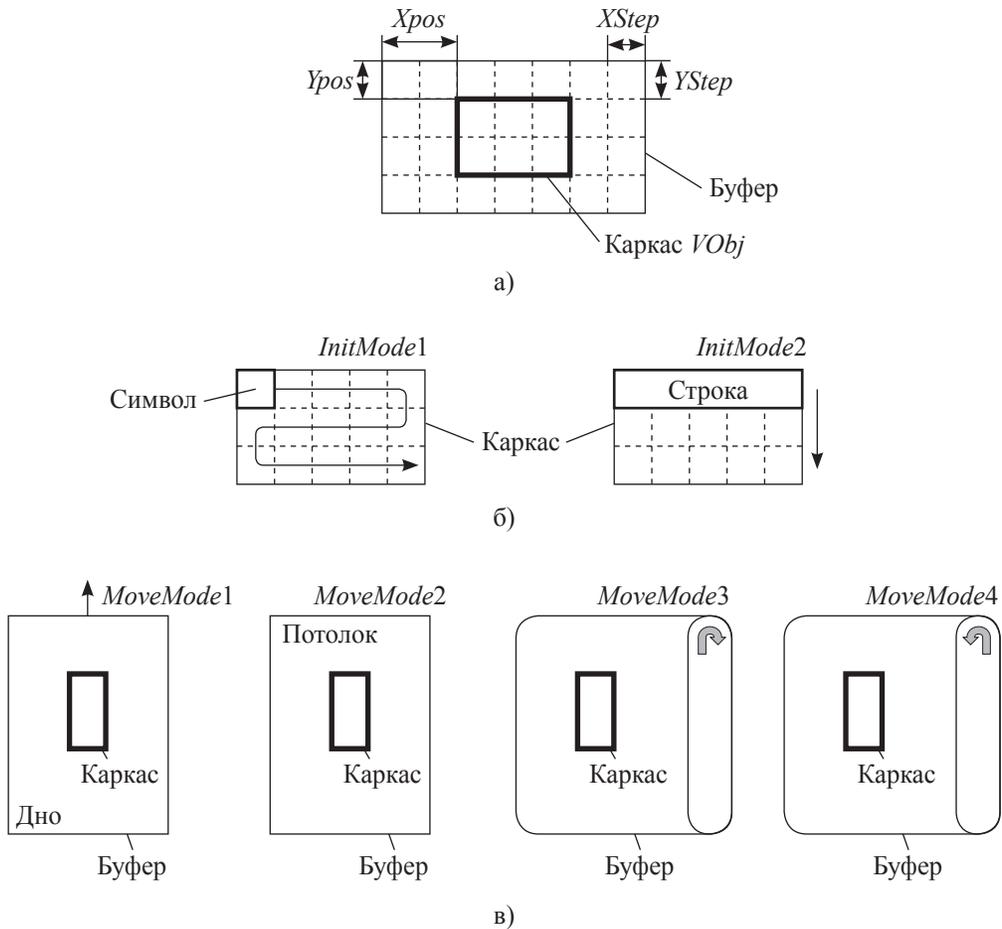


Рис. 4.6. Действие атрибутов видеоэффектов:

- а) начальное позиционирование каркаса;
- б) начальное заполнение каркаса;
- в) перемещение содержимого каркаса после его начального заполнения

В таблице 4.2 приведены описания основных атрибутов видеоэффектов. Предполагается, что через каркас можно просматривать только часть буфера. Вначале каркас позиционируется на заданный участок буфера. После начального позиционирования каркас заполняется адресуемыми элементами буфера. Предлагаются посимвольная и построчная моды заполнения каркаса. Затем каркас может перемещаться по буферу. Предусмотрены четыре моды такого перемещения. Атрибуты видеоэффектов нужны не для украшения сцены. С их помощью можно создавать видеообъекты, адекватные системе восприятия человека. Например, текст в окне может появляться не одновременно, а посимвольно или построчно со скоростью, согласованной со скоростью восприятия.

Таблица 4.2

Атрибуты видеоэффектов

№ п/п	Подкласс	Имя	Интерпретация
5	Начальное позиционирование каркаса	<i>XPos</i>	Начальное позиционирование каркаса на буфер по горизонтали
6		<i>YPos</i>	Начальное позиционирование каркаса на буфер по вертикали
7		<i>XStep</i>	Шаг позиционирования по горизонтали. Каркас перемещается по буферу в горизонтальном направлении пошагово на расстояние, кратное <i>XStep</i>
8		<i>YStep</i>	Шаг позиционирования по вертикали (см. объяснения для <i>XStep</i>)
9	Моды начального заполнения каркаса	<i>InitMod1</i>	«Посимвольно» через заданный промежуток времени. «Символ» имеет размеры <i>Xstep</i> , <i>Ystep</i>
10		<i>InitMod2</i>	«Построчно» через заданный промежуток времени. «Строка» состоит из символов. Длина строки равна длине каркаса
11		<i>InitSpeed</i>	Промежуток времени, используемый в <i>InitMod1</i> или <i>InitMod2</i>
12		<i>InitPause</i>	Выдержка времени после первоначального заполнения каркаса. [В некоторых случаях нужна для заполнения входного буфера декомпрессии. См. пояснения далее в тексте]

Окончание табл. 4.2

№ п/п	Подкласс	Имя	Интерпретация
13	Моды перемещения содержимого каркаса после его начального заполнения	<i>MoveMod1</i>	Вверх через каркас до «дна» буфера с шагом в 1 пиксель.
14		<i>MoveMod2</i>	Вниз через каркас до «потолка» буфера с шагом в 1 пиксель.
15		<i>MoveMod3</i>	Непрерывное перемещение — «барабан» вверх с шагом в 1 пиксель.
16		<i>MoveMod4</i>	Непрерывное перемещение — «барабан» вниз с шагом в 1 пиксель.
17		<i>MoveMod5</i>	Не перемещать содержимое.
18		<i>MoveCycle</i>	Количество циклов для <i>MoveMod3</i> и <i>MoveMod4</i>

Атрибуты редактируемого текста. Буфер может заполняться текстом, состоящим из графических символов, вписанных в прямоугольник. Для того, чтобы текст мог редактироваться, снабдим видеообъект атрибутом (с порядковым номером 19) с именем *FontName*, соответствующим имени фонта. Отдельный символ размещается в прямоугольнике размером *XStep*, *YStep* (см. рис. 4.6). Таким образом, предлагается использовать способ работы с текстом, базирующийся на непропорциональных шрифтах (каждый символ размещается в прямоугольнике фиксированного размера). Отсюда следует естественное ограничение предложенного способа, заключающееся в том, что в одном видеообъекте могут использоваться шрифты одного размера. Отметим, что приведенный пример атрибутивного описания текста является простым, но, как правило, достаточным, поскольку в контексте всей сцены видеообъект с редактируемым текстом является лишь одним из ее элементов.

Атрибуты поступательного перемещения. После того, как завершилась интерпретация содержимого каркаса, видеообъект может совершать поступательное перемещение по экрану. Перемещение видеообъекта может быть использовано для создания динамических сцен. Динамической сценой называется сцена, содержащая хотя бы один перемещающийся видеообъект. Так же, как и атрибуты видеоэффектов, атрибуты поступательного перемещения предназначены для облегчения восприятия сцен. Например, при восприятии сложной сцены перемещение очередного объекта создает паузу, необходимую для восприятия предшествующих объектов. Рис. 4.7 иллюстрирует действие атрибутов поступательного перемещения.

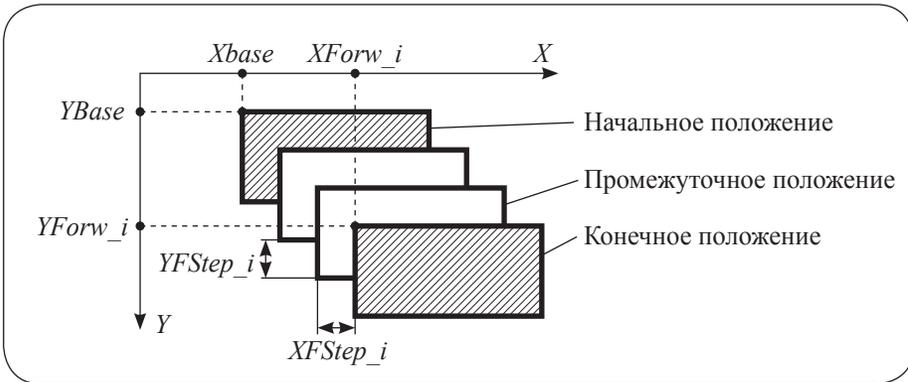


Рис. 4.7. Действие атрибутов поступательного перемещения

Траектория поступательного перемещения видеообъекта может состоять или из одного отрезка, соединяющего начальное и конечное положения базовой точки или несколько последовательно соединённых отрезков. В последнем случае каждый отрезок траектории описывается отдельно. В табл. 4.3 приведено описание основных атрибутов поступательного перемещения.

Таблица 4.3

Атрибуты поступательного перемещения

№ п/п	Наименование	Интерпретация
20	$XForw_i$	Горизонтальная координата конечного положения $VObj$ для i -го отрезка траектории.
21	$YForw_i$	Вертикальная координата конечного положения $VObj$ для i -го отрезка траектории.
22	$XStep_i$	Шаг перемещения по горизонтали для i -го отрезка траектории.
23	$YStep_i$	Шаг перемещения по вертикали для i -го отрезка траектории.
24	$ForwVis_i$	Признак «визуализации траектории перемещения». Возможны два значения: — сохранять начальное и все промежуточные положения; — не сохранять начальное и все промежуточные положения.
25	$ForwPause_i$	Выдержка времени после перемещения в конечное положение. [Эта выдержка может использоваться, исходя из «логики презентации» сцены, а также специально для заполнения входного буфера декомпрессии].

Атрибуты деформации каркаса. Атрибуты деформации каркаса так же, как и атрибуты поступательного перемещения, позволяют создавать динамические сцены, упрощающие восприятие сложной сцены. Под деформацией каркаса понимается его пропорциональное сжатие или растяжение по отношению к любой точке, находящейся как на экране, так и в «заэкранном пространстве». Однако, в предлагаемом наборе атрибутов предполагается более простой случай и считается, что деформация каркаса осуществляется либо относительно геометрического центра каркаса, либо относительно одного из его углов.

В табл. 4.4 приведено описание основных атрибутов деформации каркаса.

Таблица 4.4

Атрибуты деформации каркаса

№ п/п	Наименование	Интерпретация
26	<i>DefoCentr</i>	Центр деформации. Возможно размещение центра деформации в следующих точках: — геометрический центр каркаса; — левый верхний угол; — правый верхний угол; — левый нижний угол; — правый нижний угол
27	<i>XDefo</i>	Горизонтальная координата конечного положения каркаса
28	<i>YDefo</i>	Вертикальная координата конечного положения каркаса
29	<i>DefoType</i>	Направление деформации. Возможны следующие направления: — сжатие; — растяжение
30	<i>DefoStep</i>	Шаг деформации
31	<i>DefoVis</i>	Признак “визуализации траектории деформации”. Возможны два значения: — сохранять начальное и все промежуточные положения; — не сохранять начальное и все промежуточные положения
32	<i>DefoPause</i>	Выдержка времени после завершения деформации. Используется для той же цели, что и атрибут <i>ForwPause_i</i> (см. табл. 4.3)

Атрибуты источника информации. В табл. 4.5 приведены атрибуты источника информации, выводимой в каркас видеообъекта. Рассматриваются три таких источника: (1) сценарий; (2) внешний процесс-демон; (3) реактивный агент.

Таблица 4.5

Атрибуты источника информации

№ п/п	Наименование	Интерпретация
33	<i>Source</i>	Принимает следующие значения: — Элемент вопроса. Вывод в каркас осуществляется из сценария — Результат работы внешнего процесса-демона. Вывод в каркас осуществляет одним из процессов демонов — Ответ. Ввод в каркас осуществляется реактивным агентом

Атрибуты «времени жизни». После того, как завершилась интерпретация видеообъекта, учитывая его возможное перемещение, он может либо сохраняться в пределах сцены неопределённо долго, либо исчезнуть через заданный промежуток времени. Атрибуты «времени жизни» видеообъекта предназначены для управления длительностью пребывания видеообъекта на сцене. Описание атрибутов «времени жизни» приведено в табл. 4.6.

Таблица 4.6

Атрибуты «времени жизни»

№ п/п	Наименование	Интерпретация
34	<i>VObjPause</i>	Выдержка времени после полной интерпретации видеообъекта.
35	<i>DelAfterPause</i>	Удаление видеообъекта со сцены после интерпретации и выдержки времени <i>VObjPause</i> . Принимает два значения: — удалять после паузы; — не удалять после паузы
36	<i>DelAfterScene</i>	Удаление со сцены после завершения интерпретации всех её видеообъектов. Принимает два значения: — удалять после интерпретации сцены; — не удалять после интерпретации сцены

Практическое использование описанных классов атрибутов видеообъектов сцены и их возможности при конструировании сцен иллюстрируется примерами конкретных сцен, приведенных в приложении.

4.3. Проблема фрагментации атрибутивного описания объектов сцены

Приведенный в 4.2 пример системы атрибутивного описания видеообъекта предполагает использование 36 атрибутов. Опыт практического проектирования сцен диалоговых приложений показывает, что не все 36 атрибутов обязательно используются для описания каждого объекта. Например, не обязательно, чтобы каркас каждого объекта подвергался деформации, а сам объект участвовал в поступательном перемещении. Динамические видеообъекты составляют примерно 10% от общего числа объектов сцены.

Атрибуты, отсутствующие в описании объекта, должны быть заполнены специальным символом «отсутствия» (например, символом пробел). Совокупность символов отсутствия представляет собой неиспользуемый фрагмент как в описании отдельной сцены, так и в описании всего сценария и составляет примерно 50% от всего объема сценария. Рис. 4.8 иллюстрирует явление фрагментации на примере описания сцены, состоящей из пяти видеообъектов.

	Каркас	Эффекты	Фонт	Перемещ.	Деформ.	Источник	Время
$VObj_1$							
$VObj_2$							
$VObj_3$							
$VObj_4$							
$VObj_5$							

Рис. 4.8. Иллюстрация фрагментации в атрибутивной части описания сцены. Заштрихованы участки, заполненные символами отсутствия

На рис. 4.8 размер атрибутов в байтах пропорционален площади отдельной ячейки, а размер атрибутивного описания объекта — площади строки. Неиспользуемые участки атрибутивного описания сцены заштрихованы.

Можно предложить два способа уменьшения фрагментации в атрибутивной части описания сцены и сценария:

- 1) использование методов компрессии-декомпрессии цифровых сигналов для устранения избыточности;
- 2) использование типовых видеообъектов с минимальным количеством символов отсутствия.

Применение универсальных и специализированных методов компрессии-декомпрессии [132–134] является радикальным решением проблемы устранения фрагментации и избыточности в сценарии, поскольку действие этого подхода распространяется не только на атрибутивную, но и на неатрибутивную часть сценария. Применение этих методов предполагает совмещение операции чтения видеообъекта из памяти с операцией декомпрессии и операции записи видеообъекта в память с операцией компрессии.

Ясно, что многократно выполняемые операции компрессии и декомпрессии в процессе интерпретации сценария ухудшают инерционные свойства системы и требуют дополнительных мер, направленных на компенсацию временных затрат на компрессию и декомпрессию. Отметим также, что в процессе компрессии удаляется избыточность из цифрового файла методами, не связанными с их внутренней логической организацией. Поэтому целесообразно применять эти методы совместно с другими методами устранения фрагментации.

Использование типовых объектов означает, что любая, сколь угодно сложная сцена конструируется из ограниченного количества объектов, в описании которых отсутствует или минимизирована фрагментация. Напомним, что сценой мы называем даталогическое представление вопроса при помощи цепи объектов, в которой оба логических компонента вопроса (субъект и требование) моделируются последовательно интерпретируемой цепью объектов цепи. Введём типовые объекты, исходя из потребностей конструирования сцены в рамках отмеченной модели. Начнём с того, что разделим все классы атрибутов видеообъекта на два подкласса:

- обязательные атрибуты;
- необязательные атрибуты.

К обязательным атрибутам отнесём: атрибуты каркаса (табл. 4.1), атрибут источника информации (табл. 4.5) и атрибуты времени жизни (табл. 4.6). Все остальные атрибуты будем считать необязательными.

Введём четыре типовых видеообъекта, восприятие которых осуществляется реактивным агентом пассивно, в том смысле, что восприятие этих объектов не предполагает каких-либо действий с клавиатурой либо манипулятором типа «мышь». Каждый типовой видеообъект содержит все обязательные атрибуты и некоторое количество необязательных атрибутов. Характерные признаки и свойства типовых объектов приведены в табл. 4.7.

При конструировании сцены типовые видеообъекты могут агрегироваться в логические компоненты вопроса — субъект и требование. Как в модели, предложенной Белманом и Стиллом (см. подраздел 2.3.3), так и в тернарной модели (см. подраздел 2.3.4) основу субъекта вопроса составляет список, из которого в соответствии с требованием вопроса конструируется ответ.

В том случае, когда сцена моделирует ли-вопрос активного агента, а список субъекта конечен и может быть размещён на сцене, являющейся экраном монитора, то его можно формировать из типовых видео объектов по принципу: «один типовой видеообъект соответствует одному элементу списка субъекта». В этом случае ответ — это один или несколько типовых видеообъектов, выбираемых реактивным агентом.

В том случае, когда сцена моделирует какой-вопрос активного агента, а список субъекта либо бесконечен по природе, либо настолько велик, что не может быть физически размещён на сцене-экране, то из типовых видеообъектов формируется только часть ожидаемого ответа. Остальная часть ответа доопределяется реактивным агентом путем ввода данных в каркас одного из видеообъектов сцены.

Таблица 4.7

Характеристики типовых видеообъектов

Имя типового объекта и его характерные признаки	Необязательные атрибуты
<i>VObjType1</i> Неподвижный объект	атрибуты видеоэффектов; атрибуты редактируемого текста
<i>VObjType2</i> Объект, участвующий в поступательном перемещении	атрибуты видеоэффектов; атрибуты редактируемого текста; атрибуты поступательного перемещения
<i>VObjType3</i> Объект с деформируемым каркасом	атрибуты видеоэффектов; атрибуты редактируемого текста; атрибуты деформации каркаса
<i>VObjType4</i> Универсальный объект	атрибуты видеоэффектов; атрибуты редактируемого текста; атрибуты деформации каркаса атрибуты поступательного перемещения

Агрегат типовых объектов обладает дополнительными атрибутами, присущими этому агрегату, но не присущими каждому из его объектов в отдельности. Введём два типа агрегатов типовых объектов, отличающихся способом формирования ответа реактивным агентом:

- *Aggreg1* — список типовых видеообъектов, дополненных атрибутом, позволяющим реализовать на этом списке операцию выбора;
- *Aggreg2* — список типовых видеообъектов, дополненных набором атрибутов, позволяющих реализовать на этом списке операции выбора и доопределения с помощью операции ввода.

Рассмотрим подробнее атрибуты агрегатов, необходимые для реализации операций выбора и доопределения. Во-первых, отметим, что операция выбора должна обеспечивать выбор нескольких объектов, входящих в состав агрегата. Необходимость выбора нескольких объектов следует из того факта, что в ответ могут переходить несколько элементов списка субъекта вопроса. Для реализации операции выбора нескольких объектов агрегата сделаем следующее. Во-первых, дополним каждый типовой объект его номером в цепи объектов сцены, а во-вторых, дополним весь агрегат отображением (представленным в табличной форме), которое ставит в соответствие каждому ожидаемому набору номеров объектов (по сути, каждому ожидаемому ответу) имя следующего вопроса.

Таким образом, агрегат *Aggreg1* дополняется следующими атрибутами:

$$\{VObjOrder_{\alpha}\}, \alpha = 1, \dots, k, \quad (4.1)$$

где α — индекс типового объекта в агрегате объектов;

k — общее количество типовых объектов, входящих в состав агрегата.

$$r: \{\wedge_{\alpha=1}^n VObjOrder_{\alpha}\} \rightarrow \{t_i\}, i = 1, \dots, m, \quad (4.2)$$

где r — отображение всех подмножеств индексов в агрегате в имя последующей транзакции (каждое из подмножеств объектов агрегата соответствует одному ожидаемому ответу);

n — общее количество подмножеств индексов в агрегате;

$\{t_i\}$ — множество транзакций сценария.

На рис. 4.9 приведен рисунок примера, иллюстрирующий структуру агрегата *Aggreg1* и отображения r .

Сцена, изображенная на рис. 4.9, представляет собой цепь, состоящую из семи типовых объектов, объединённых в один агрегат типа *Aggreg1*. Отображение r описывает индексы четырёх ожидаемых ответов и расшифровывается следующим образом.

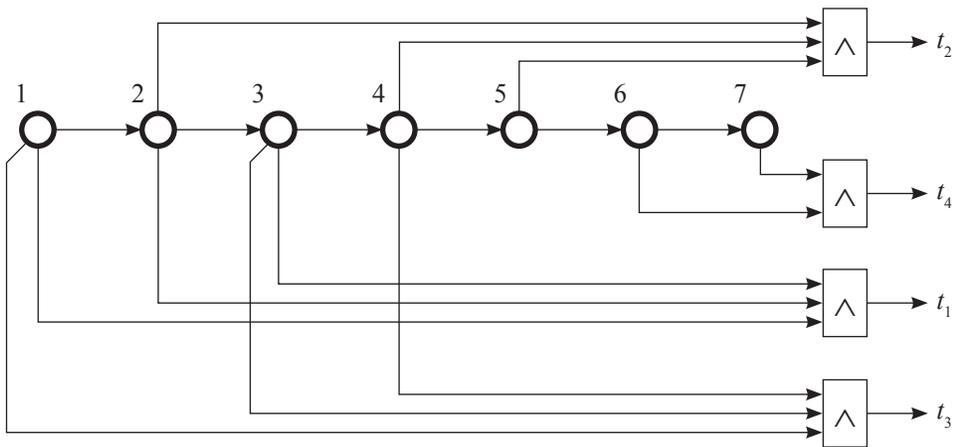
Если выбраны видеообъекты с индексами 1, 2, 3, то перейти к транзакции 1.

Если выбраны видеообъекты с индексами 2, 4, 5, то перейти к транзакции 2.

Если выбраны видеообъекты с индексами 1, 3, 4, то перейти к транзакции 3.

Если выбраны видеообъекты с индексами 6, 7, то перейти к транзакции 4.

Атрибутивное дополнение агрегата *Aggreg2* отличается от атрибутивного дополнения агрегата *Aggreg1* только атрибутом, указывающим, что после выбора видеообъекта осуществляется ввод данных в его каркас.



$VObjOrder = 1$	$VObjOrder = 2$	$VObjOrder = 3$	t_1
$VObjOrder = 2$	$VObjOrder = 4$	$VObjOrder = 5$	t_2
$VObjOrder = 1$	$VObjOrder = 3$	$VObjOrder = 4$	t_3
$VObjOrder = 6$		$VObjOrder = 7$	t_4

Рис. 4.9. Структура агрегата *Aggreg1* и табличное представление отображения r для примера, описанного в тексте

При конструировании сцены необходимо придерживаться следующего правила. Сцена может содержать не более одного агрегата (типа *Aggreg1* или *Aggreg2*), который должен располагаться последним в цепи объектов, составляющих сцену. Приведенное правило очевидно, поскольку реализация операции выбора объекта и последующее доопределение его каркаса означает завершение работы с данной сценой и переход к следующей сцене.

4.4. Модель памяти вопросов

При построении логических моделей эротетического диалогового процесса (см. раздел 3) было принято, что сценарий эротетического диалога состоит из памяти вопросов и диалогового метода доступа к памяти вопросов (*DiAM*). Память вопросов предназначена для хранения декларативные знаний активного агента в виде информационных посылок-вопросов, а метод доступа к памяти вопросов является хранилищем метода достижения цели диалога, или диалоговым методом решения задачи. Диалоговый метод доступа трансформирует имя текущего ответа реактивного агента в имя последующей транзакции. Память

вопросов рассматривается нами как память прямого доступа, которая, получив на входе имя транзакции, формирует на выходе последующий вопрос. На рис. 4.9 изображена структура сценария эротетического диалога, состоящая из памяти вопросов (*QueMem*) и диалогового метода доступа к памяти вопросов.

Для формального описания диалогового метода доступа были предложены три модели: автоматная модель (см. подраздел 3.1); Петри-модель (см. подраздел 3.2) и сетевая модель (см. подраздел 3.3). Сетевая модель основывалась на предположении о том, что *DiAM* представляет собой сеть, состоящую из связанных между собой узловых элементов — транзакций. Исходя из диалогового подхода к проектированию информационных систем, сеть *DiAM* была представлена в виде реляционной базы данных (см. подраздел 3.3.2).

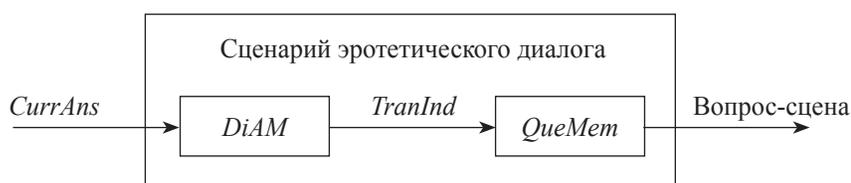


Рис. 4.10. Структура сценария эротетического диалога.

CurrAns — текущий ответ реактивного агента;

TranInd — имя последующей транзакции сценария.

Результаты, полученные в процессе диалогового моделирования вопроса, позволяют расширить реляционную базу диалогового метода доступа и дополнить её реляционной базой, соответствующей памяти вопросов.

В рассматриваемой ниже модели принято, что сцена-вопрос интерпретируется исключительно через видеоканал компьютера. Иными словами, принято, что *вопрос представляет собой цепь видеообъектов, последовательно интерпретируемых на экран монитора.*

Будем моделировать память вопросов базой данных, включающей следующие пять отношений:

$$QueMem = \{VObj, ObligAttr, Type1Attr, Type2Attr, Type3Attr, Type4Attr, Navig\}, \quad (4.3)$$

где *VObj* — отношение, моделирующее множество видеообъектов, достаточных для формирования любой сцены;

ObligAttr — отношение, моделирующее обязательные атрибуты видеообъектов;

Type1Attr — отношение, моделирующее необязательные атрибуты видеообъекта *VObjType1*;

Type2Attr — отношение, моделирующее необязательные атрибуты видеообъекта *VObjType2*;

Type3Attr — отношение, моделирующее необязательные атрибуты видеообъекта *VObjType3*;

Type4Attr — отношение, моделирующее необязательные атрибуты видеообъекта *VObjType4*;

Navig — отношение, моделирующее агрегаты видеообъектов.

Представим каждое из отношений базы (4.3) в виде набора атрибутов. Для определенности будем считать, что одна сцена формируется не более, чем из десяти видеообъектов. В приведенных ниже формулах атрибуты-ключи выделены подчеркиванием, а выражение $dom(Attr) = \{...\}$ специфицирует домен соответствующего атрибута.

$$VObj = \{\underline{QueInd}, \underline{VObjOrder}, \underline{VObjInd}, \underline{VObjType}, \underline{BitFileInd}\} \quad (4.4)$$

Атрибут с именем *QueInd* моделирует имя вопроса-сцены, который формируется из видеообъектов.

Атрибут с именем *VObjOrder* моделирует порядковый номер видеообъекта в цепи объектов сцены, $dom(VObjOrder) = \{1, \dots, VObjQ\}$, где *VObjQ* — наибольшее количество видеообъектов сцены (для определенности принято, что *VObjQ* = 10).

Атрибут с именем *VObjInd* моделирует имя видеообъекта. Каждый видеообъект сценария идентифицируется уникальным именем. В пределах одной сцены может быть несколько видеообъектов с одним именем, но различными порядковыми номерами.

Атрибут с именем *VObjType* моделирует тип видеообъекта, $dom(VObjType) = \{VObjType1, VObjType2, VObjType3, VObjType4\}$.

Атрибут с именем *BitFileInd* моделирует имя неатрибутивного описания, ассоциированного с видеообъектом. Неатрибутивное описание рассматривается как отдельный файл.

$$ObligAttr = \{\underline{VObjInd}, \underline{XBase}, \underline{YBase}, \underline{XSize}, \underline{YSize}, \underline{Source}, \underline{VObjPause}, \underline{DelAfterPause}, \underline{DelAfterScene}\} \quad (4.5)$$

Атрибут с именем *VObjInd* моделирует имя видеообъекта.

Атрибуты с именами *XBase* и *YBase* моделируют координаты левого верхнего угла каркаса видеообъекта.

Атрибуты с именами *XSize* и *YSize* моделируют размеры каркаса видеообъекта по горизонтали и вертикали соответственно.

Атрибут с именем *Source* моделирует источник данных для заполнения каркаса. Каркас может заполняться: (1) реактивным агентом, (2) внешним процессом демоном, (3) из внешнего файла.

Атрибут с именем *VObjPause* моделирует выдержку времени после полного завершения интерпретации видеообъекта.

Атрибут с именем *DelAfterPause* моделирует признак удаления видеообъекта после выдержки времени, которая задается атрибутом *VObjPause*. Атрибут принимает два значения: «удалять», «не удалять».

Атрибут с именем *DelAfterScene* моделирует признак удаления видеообъекта после завершения интерпретации всей сцены. Атрибут принимает два значения: «удалять», «не удалять».

$$\begin{aligned} Type1Attr = \{ & \underline{VObjInd}, \underline{VObjType}, \underline{FontName}, \underline{XPose}, \underline{YPose}, \underline{XStep}, \underline{YStep}, \\ & \underline{InitMode1}, \underline{InitMode2}, \underline{InitSpeed}, \underline{InitPause}, \underline{MoveMode1}, \\ & \underline{MoveMode2}, \underline{MoveMode3}, \underline{MoveMode4}, \underline{MoveMode5}, \underline{MoveCycle} \} \end{aligned} \quad (4.6)$$

Атрибут с именем *VObjInd* моделирует имя видеообъекта.

Атрибут с именем *VObjType* моделирует тип видеообъекта $dom(VObjType) = \{VObjType1, VObjType2, VObjType3, VObjType4\}$.

Атрибут с именем *FontName* моделирует имя фонта, который используется в случае заполнения каркаса видеообъекта текстом. Отдельный символ размещается в прямоугольнике размером *XStep*, *YStep*.

Атрибуты с именами *XPose* и *YPose* моделируют начальное расположение (позиционирование) каркаса видеообъекта в буфере.

Атрибуты с именами *XStep* и *YStep* моделируют шаг перемещения каркаса видеообъекта по горизонтали и вертикали. Каркас видеообъекта перемещается по буферу в горизонтальном и вертикальном направлениях пошагово на расстояние, кратное *XStep* и *YStep* соответственно.

Атрибуты с именами *MoveMode1–MoveMode5* моделируют моды перемещения каркаса видеообъекта после его начального заполнения (см. табл. 4.2). Атрибуты принимают значения «да» или «нет». Значение «да» может принимать только один атрибут.

Атрибут с именем *MoveCycle* моделирует количество циклов вращения «барабана» (см. табл. 4.2) в том случае, если один из атрибутов *MoveMode3* или *MoveMode4* принял значение «да».

$$\begin{aligned} Type2Attr = \{ & \underline{VObjInd}, \underline{VObjType}, \underline{Segment}, \underline{FontName}, \underline{XPose}, \underline{YPose}, \\ & \underline{XStep}, \underline{YStep}, \underline{InitMode1}, \underline{InitMode2}, \underline{InitSpeed}, \underline{InitPause}, \\ & \underline{MoveMode1}, \underline{MoveMode2}, \underline{MoveMode3}, \underline{MoveMode4}, \\ & \underline{MoveMode5}, \underline{MoveCycle}, \underline{XForw}, \underline{YForw}, \underline{XForwStep}, \\ & \underline{YForwStep}, \underline{ForwVis}, \underline{ForwPause} \}. \end{aligned} \quad (4.7)$$

Один кортеж отношения (4.7) описывает один отрезок траектории поступательного перемещения видеообъекта типа *VObjType2*. Номер отрезка траектории моделируется атрибутом *Segment*. Атрибуты отношения *Type2Attr* состоят из атрибутов отношения *Type1Attr*, к которым добавлены атрибуты, специфицирующие поступательное перемещение на заданном отрезке траектории.

Атрибуты с именами *XForw* и *YForw* моделируют координаты конечного положения видеообъекта по горизонтали и вертикали соответственно.

Атрибуты с именами *XForwStep* и *YForwStep* моделируют шаг перемещения по горизонтали и вертикали соответственно.

Атрибут с именем *ForwVis* моделирует признак визуализации траектории перемещения видео объекта. Атрибут принимает одно из двух значений: «сохранять начальное и все промежуточные положения» и «не сохранять начальное и все промежуточные положения».

Атрибут с именем *ForwPause* моделирует выдержку времени после перемещения видеообъекта в конечное положение.

$$\begin{aligned} \text{Type3Attr} = \{ & \underline{VObjInd}, \underline{VObjType}, \text{FontName}, \text{XPose}, \text{YPose}, \\ & \text{XStep}, \text{YStep}, \text{InitMode1}, \text{InitMode2}, \text{InitSpeed}, \text{InitPause}, \\ & \text{MoveMode1}, \text{MoveMode2}, \text{MoveMode3}, \text{MoveMode4}, \\ & \text{MoveMode5}, \text{MoveCycle}, \text{DefoCentr}, \text{XDefo}, \text{YDefo}, \\ & \text{DefoType}, \text{DefoStep}, \text{DefoVis}, \text{DefoPause} \} \end{aligned} \quad (4.8)$$

Атрибуты отношения *Type3Attr* состоят из атрибутов отношения *Type1Attr*, к которым добавлены атрибуты специфицирующие деформацию каркаса видеообъекта.

Атрибут с именем *DefoCentr* моделирует местоположение точки, относительно которой выполняется деформация каркаса. Атрибут принимает значения, определяющие в качестве центра деформации либо геометрический центр каркаса, либо один из его углов.

Атрибуты с именами *XDefo* и *YDefo* моделируют координаты конечного положения левого верхнего угла каркаса видеообъекта после деформации.

Атрибут с именем *DefoType* моделирует направление деформации и принимает два значения: «сжатие», «растяжение».

Атрибут с именем *DefoStep* моделирует шаг деформации.

Атрибут с именем *DefoVis* моделирует признак визуализации траектории деформации каркаса видеообъекта. Атрибут принимает одно из двух значений: «сохранять начальное и все промежуточные положения» и «не сохранять начальное и все промежуточные положения».

Атрибут с именем *DefoPause* моделирует выдержку времени после завершения деформации каркаса видеообъекта.

$$\begin{aligned} \text{Type4Attr} = \{ & \underline{VObjInd}, \underline{VObjType}, \text{FontName}, \text{XPose}, \text{YPose}, \text{XStep}, \\ & \text{YStep}, \text{InitMode1}, \text{InitMode2}, \text{InitSpeed}, \text{InitPause}, \\ & \text{MoveMode1}, \text{MoveMode2}, \text{MoveMode3}, \text{MoveMode4}, \\ & \text{MoveMode5}, \text{MoveCycle}, \text{DefoCentr}, \text{XDefo}, \text{YDefo}, \\ & \text{DefoType}, \text{DefoStep}, \text{DefoVis}, \text{DefoPause}, \text{XForw}, \\ & \text{YForw}, \text{XForwStep}, \text{YForwStep}, \text{ForwVis}, \text{ForwPause} \} \end{aligned} \quad (4.9)$$

Атрибуты отношения *Type4Attr* состоят из всех необязательных атрибутов, перечисленных в отношениях (4.6)–(4.8).

Отношения (4.6)–(4.9) содержат атрибуты, необходимые для интерпретации типовых объектов сцены (*VObjType1*, *VObjType2*, *VObjType3*, *VObjType4*). Отношение с именем *Navig* содержит атрибуты, необходимые для интерпретации агрегатов *Aggreg1* и *Aggreg2*. Напомним, что сцена может содержать не более одного агрегата объектов, поэтому целесообразно ассоциировать атрибуты агрегатов не с отдельным объектом, а с отдельной транзакцией.

$$VObj = \{QueInd, AggregType, NextTranInd, VObj1, \dots, VObj10, SimbFileInd\} \quad (4.10)$$

Один кортеж отношения (4.10) описывает одну строку отображения *r* (один ожидаемый ответ на рис. 4.9) для агрегата объектов типов *Aggreg1* или *Aggreg2*.

Атрибут с именем *QueInd* моделирует имя вопроса-сцены, который формируется из видеообъектов.

Атрибут с именем *AggregType* моделирует тип агрегата и принимает два значения, соответствующие *Aggreg1* или *Aggreg2*.

Атрибут с именем *NextTranInd* моделирует имя следующей транзакции сценария.

Атрибуты *VObj1–VObj10* моделируют признак включения соответствующего видеообъекта в ожидаемый ответ. Каждый из этих атрибутов принимает два значения: «объект включен в ожидаемый ответ», «объект не включен в ожидаемый ответ». Ранее было принято, что сцена-вопрос состоит не более, чем из десяти видеообъектов.

Атрибут с именем *SimbFileInd* моделирует имя файла, в который помещаются вводимые реактивным агентом символьные данные для случая, когда атрибут *AggregType* специфицирует агрегат типа *Aggreg2*.

ЧАСТЬ 2

**ДИАЛОГОВЫЙ
РЕШАТЕЛЬ ЗАДАЧ**

5

ЭРОТЕТИЧЕСКИЙ ДИАЛОГ КАК МЕТОД МОДЕЛИРОВАНИЯ ПРОЦЕДУРНЫХ ЗНАНИЙ ЭКСПЕРТА

Деление знаний на декларативные (простые и структурированные факты, представленные при помощи символов или образов) и процедурные (умения и навыки) является общепринятой и конструктивной классификацией. Декларативно-процедурное различие в представлении знаний лежит в основе известных унифицированных когнитивных моделей *ACT-R* [54] и *SOAR* [135]. Декларативные, или фактические, знания можно относительно легко представить их описаниями, в то время как процедурные знания представимы действиями или поступками. Хорошим примером декларативных знаний являются фактические знания об алгоритмических языках и методах программирования, а хорошим примером процедурных знаний — умение разрабатывать программы.

Как это следует из предыдущего изложения, активный агент, как правило, не «вычисляет» очередной вопрос, а разыскивает его в памяти вопросов *QueMet*, используя в качестве метода доступа *DiAM* метод достижения цели диалога. Таким образом, метод доступа к памяти вопросов является одновременно и методом достижения цели диалога или диалоговым методом решения задачи.

Применительно к описанным выше моделям эротетического диалогового процесса память вопросов будет рассматриваться нами как хранилище декларативных знаний, а диалоговый метод доступа — как хранилище процедурных знаний активного агента или эксперта, если активный агент является носителем экспертных знаний.

Настоящий раздел посвящён, главным образом, сравнительному анализу того, как моделируются процедурные знания эксперта (как моделируется процесс решения задачи) в эротетической диалоговой системе и в классической экспертной системе, основанной на правилах. В разделе также рассмотрены вопросы совмещения диалогового процесса и процесса дедуктивного вывода [98, 136–141].

5.1. Моделирующие возможности диалогового процесса

Интерактивный и диалоговый процессы являются удобной и естественной формой общения человека с программной системой, которая чаще всего используется для ввода команд оперативного управления программой либо исходных данных, необходимых для работы сугубо вычислительных процедур. Однако известны примеры диалоговых программ, в которых диалоговое общения пользователя с программой приобретает свойства метода решения задачи [142, 143]. Работа таких программ осуществляется пошагово, новые данные вводятся порциями на каждом шаге, а программа реагирует на каждые введенные данные изменением своего поведения. Таким образом, поведение программы на последующем шаге определяется данными, введенными на предыдущем шаге.

Для того, чтобы иметь возможность реагировать на множество внешних вводимых данных, диалоговая программа должна иметь внутреннюю логическую избыточность и, вообще говоря, быть сложнее, чем не диалоговая и сугубо алгоритмическая программа. Следовательно, можно предположить, что алгоритмические вычисления являются частным случаем диалоговых вычислений и могут выводиться из последних.

В [144] проводится сопоставление диалогового и алгоритмического методов вычислений на основе стандартных схем программ [145] и подтверждается предположение о том, что класс сугубо алгоритмических вычислений может рассматриваться как подкласс класса диалоговых вычислений. Автор отталкивается от логического понимания сущности алгоритма. Алгоритм трактуется как дедуктивная цепь, связывающая входное слово F с выходным словом G . Отмечается проблема построения алгоритма «максимальной» мощности, названная проблемой распознавания выводимости, которая формулируется следующим образом. Для любых двух слов F и G необходимо определить, существует ли дедуктивная цепь, ведущая от F к G . Делается вывод о том, что проблема распознавания выводимости алгоритмически неразрешима, иными словами, может случиться так, что для любых F и G не удастся найти дедуктивную цепь, ведущую от F к G .

Диалоговый процесс рассматривается не как способ решения проблемы распознавания выводимости, а как некоторая конструкция, хранящая множество дедуктивных цепей, ведущих от входного слова F к каждому элементу множества выходных слов

$$\{g_i\}, i = 1, \dots, n.$$

Отмеченная диалоговая конструкция позволяет в процессе диалоговых вычислений реализовать одну из множества дедуктивных цепей. Диалоговые вычисления управляются агентом диалогового процесса, а диалоговая конструкция может быть описана следующим образом:

m_0 : *ВЫВОД*(ИмяПроц.1, ..., ИмяПроц.n, ($V \in M = \{m_i\} i = 1, \dots, n$)); *СТОП*;
ЕСЛИ $v = m_1$ *ТО* на m_1 *ИНАЧЕ*
ЕСЛИ $v = m_2$ *ТО* на m_2 *ИНАЧЕ*
 ...
ЕСЛИ $v = m_n$ *ТО* на m_n *ИНАЧЕ* *НА* m_0 ;
 m_1 : *ВЫВОД*(ПараметрыПроц.1); *СТОП*;
 реализация Проц.1;
 на m_0 ;
 ...
 m_{n-1} : *ВЫВОД*(ПараметрыПроц.n-1); *СТОП*;
 реализация Проц.n-1;
 на m_0 ;
 m_n : *ОСТАНОВ*,

где $M = \{m_1, m_2, \dots, m_n\}$ — множество меток операторов схем программ;
 V — переменная.

Конкретная дедуктивная цепь, реализуемая в процессе диалоговых вычислений, полностью задаётся последовательностью значений переменной V , которая названа протоколом порождения алгоритма. Следовательно, диалоговая конструкция может порождать сколь угодно много различных алгоритмов, если учесть, что протокол порождения алгоритма может иметь циклический характер. Логическую основу диалоговой конструкции составляют предложения типа *ЕСЛИ–ТО–ИНАЧЕ*, используемые для связи отдельных процессов в дедуктивные цепи.

Приведенные рассуждения подкрепляют вывод о том, что возможности диалогового процесса значительно шире, чем его традиционное использование в системах электронной обработки данных, и подтверждают предположение о том, что сценарий активного агента может рассматриваться как метод решения задачи.

Сферой применимости диалогового метода решения задач являются задачи, относящиеся к классу плохо формализуемых [146]. Примерами таких задач могут служить задачи проектирования и обучения.

Под методом решения плохо формализуемой задачи будем понимать процедурные знания человека-эксперта, умеющего решать задачу. Таким образом, диалоговый метод решения плохо формализуемой задачи должен имитировать поведение эксперта в процессе решения плохо формализуемой задачи. В [147] приводятся соображения, объясняющие поведение эксперта, решающего задачу:

«...Этот тип (знаний) представляет собой врождённые или приобретённые правила поведения, которые позволяют в данной конкретной ситуации принять решение о необходимых действиях. Он (эксперт) использует информацию в порядке, обратном тому, в котором она была

получена. В качестве примера можно привести рассуждения типа: «Я знаю, что это действие приводит к такому-то результату, поэтому, если я хочу получить именно этот результат, я могу рассмотреть это действие». Человек постоянно пользуется этим типом знаний при восприятии, формировании концепций, решении задач и формальных рассуждениях. Появление экспертных систем связано с необходимостью принятия в расчёт именно этого фундаментального типа человеческих знаний».

В табл. 5.1 приведены основные характеристики «решающего задачу» поведения человека-эксперта при решении плохо формализуемой задачи и эротетического диалогового процесса со стороны активного агента, иллюстрирующие аналогию между этими двумя процессами.

Таблица 5.1

**Характеристики «решающего задачу» поведения человека эксперта
и активного агента эротетического диалога**

Характеристики поведения эксперта при решении плохо формализуемой задачи	Характеристики поведения активного агента эротетического диалога
Многошаговый процесс. На каждом шаге выполняется простое действие, в результате чего образуется некоторый промежуточный результат	Многошаговый процесс. На каждом шаге-транзакции активный агент передаёт реактивному агенту вопрос и получает релевантный ответ
Эксперт знает, какой должен быть следующий шаг. Процедурные знания эксперта — это знания о связях между предыдущими результатами и последующим действием	Активный агент формирует последующий вопрос после анализа полученного ответа с учётом принципов: «глубины диалога», «истории ответов» и «истории вопросов»
Часто действие, выполняемое в шаге, и полученный результат находятся в отношении «часть — целое»	Ответ, в логическом смысле, является частью субъекта вопроса

Процедурные знания эксперта представлены в *DiAM* в виде сетевой структуры, состоящей из цепочек транзакций, а процесс решения задачи представляет собой управляемый поиск целевого узла или целевой траектории. Ясно, что такой подход а priori предполагает, что сценарий активного партнёра содержит искомый результат или множество искомых результатов решения задачи.

Рассмотрим пример, иллюстрирующий способность *DiAM* хранить процедурные знания эксперта. Эксперты часто пользуются методом аналогий, используя одни и те же навыки при решении различных задач. Это означает, что в совокупности процедурных знаний эксперта можно выделить некоторые схемы — «клише», применимые для целого ряда аналогичных задач. На рис. 5.1 изображен фрагмент *DiAM*, представляющий собой одно из таких клише. С целью упрощения рисунка, для представления сети *DiAM* использована Петри-модель.

Диалоговый метод, иллюстрируемый рис. 5.1, может использоваться для решения ряда аналогичных плохо формализованных задач.

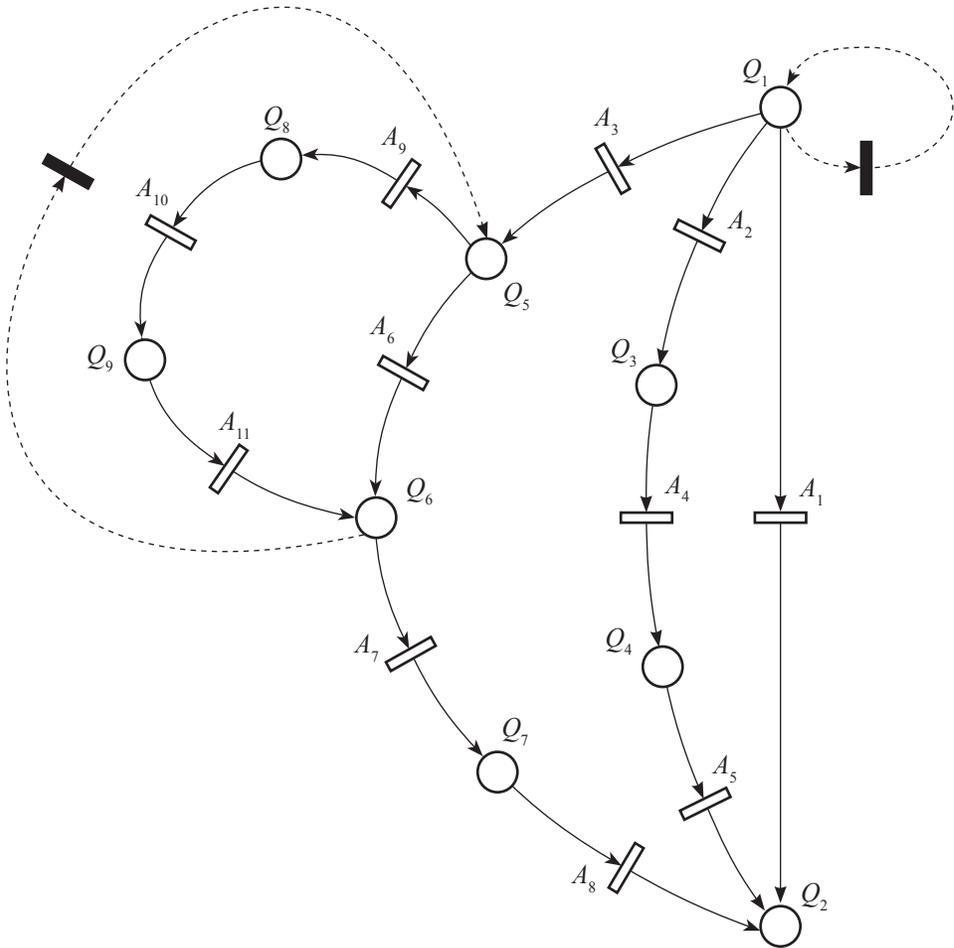


Рис. 5.1. Пример представления процедурных знаний эксперта в *DiAM*. Пунктиром показано «использование» нераспознаваемых ответов реактивного агента. $Q_1...Q_9$ — вопросы активного агента; $A_1...A_{11}$ — ответы реактивного агента

Задача 1. Передача декларативных знаний реактивному агенту, с учётом индивидуальной способности реактивного агента к восприятию материала.

Задача заключается в том, что реактивному агенту должен быть передан некоторый объём декларативных знаний. Весь объём знаний разбивается на множество элементарных обучающих эпизодов, имеющих интеррогативный характер, и передаётся реактивному агенту за различное количество транзакций в зависимости от его индивидуальной способности к восприятию материала.

Реактивному агенту передаётся обучающий эпизод Q_1 . Если на обучающий эпизод Q_1 получен ответ A_1 , то весь объём знаний передаётся реактивному агенту за одну транзакцию, цепь: Q_1-Q_2 . Ответ A_1 , таким образом, означает высокую индивидуальную способность реактивного агента к восприятию материала.

Если на обучающий эпизод Q_1 получен ответ A_2 , то весь объём знаний передаётся реактивному агенту за три транзакции, цепь: $Q_1-Q_3-Q_4-Q_2$. Ответ A_2 , таким образом, означает среднюю индивидуальную способность реактивного агента к восприятию материала.

Если на эпизод Q_1 получен ответ A_3 , а на эпизод Q_5 — ответ A_6 , то весь объём знаний передаётся реактивному агенту за четыре шага, цепь: $Q_1-Q_5-Q_6-Q_7-Q_2$. Получение ответа A_6 на эпизод Q_5 при условии, что на эпизод Q_1 получен ответ A_3 , означает низкую индивидуальную способность реактивного агента к восприятию материала.

Если на эпизод Q_1 получен ответ A_3 , а на эпизод Q_5 — ответ A_9 , то весь объём знаний передаётся реактивному агенту за шесть шагов, цепь: $Q_1-Q_5-Q_8-Q_9-Q_6-Q_7-Q_2$. Получение ответа A_9 на эпизод Q_5 при условии, что на эпизод Q_1 получен ответ A_3 , означает очень низкую индивидуальную способность реактивного агента к восприятию материала.

Нераспознанные ответы реактивного агента (на рис. 5.1 отмечены зачерченными переходами и пунктирными линиями) предполагают либо повторение предыдущего эпизода (случай Q_1), либо возвращение на ближайшую точку ветвления (случай Q_5).

Задача 2. Сбор исходных данных, детерминирующих параметры вычислительной процедуры.

От реактивного агента необходимо получить набор числовых данных (одно данное за одну транзакцию), который определяет значения параметров процедуры-демона, ассоциированного с позицией Q_2 . Процедура-демон, ассоциированная с позицией Q_2 , может реализовывать некоторый сугубо вычислительный алгоритм, или формировать запрос к проблемной базе данных, или обеспечивать связь с системой автоматизированного проектирования общего назначения и т. п. Количество данных, необходимых для работы процедуры-демона, различно и определяется реактивным агентом. Сеть *DiAM*, приведенная на рис. 5.1, обеспечивает формирование различных наборов числовых данных в зависимости от ответов реактивного агента.

5.2. Диалоговый и дедуктивный процессы в экспертных системах

Экспертные системы (ЭС) являются классом прикладных программ, ориентированных на решение плохо формализуемых задач методами, моделирующими процедурные знания эксперта и опирающимися на логические умозаключения. Таким образом, сфера применимости ЭС пересекается со сферой применимости диалоговых систем. Два процесса, «жизненно» важные для функционирования диалоговых систем, являются также неотъемлемой частью функционирования ЭС. Это диалоговый и дедуктивный процессы.

В настоящем подразделе рассмотрены типы и свойства диалогового процесса, сопровождающего работу ЭС, а также методы, применяемые в ЭС при построении дедуктивных умозаключений. Рассматривается класс ЭС, в которых процедурные знания эксперта моделируются набором продукционных правил, поскольку такие ЭС являются доминирующими, интенсивно используют диалоговый процесс и хорошо имитируют способность человека выполнять дедуктивные умозаключения.

Диалог с пользователем является неотъемлемой частью функционирования ЭС. В ходе диалогового взаимодействия с пользователем, ЭС может потребовать от него декларативные знания-факты, необходимые для продолжения процесса вывода. Это происходит в тех случаях, когда требуемые знания отсутствуют в базе знаний ЭС.

Распределение ролей между ЭС и пользователем зависит от того, используется ли при функционировании ЭС её объяснительная компонента. В том случае, когда объяснительная компонента не используется, то ЭС выступает в роли активного агента, а её пользователь — в роли реактивного агента, а сам диалог относится к категории диалогов с фиксированным распределением ролей. Субъект и требование вопроса формируются таким образом, чтобы ответ пользователя содержал запрашиваемые факты-посылки, отсутствующие в базе знаний.

В том случае, когда используется объяснительная компонента, диалог относится к категории диалогов со свободным распределением ролей, а во время объяснений ЭС выступает в роли реактивного агента.

Для того, чтобы убедиться в сказанном, достаточно изучить протокол взаимодействия пользователя и ЭС на некотором, желательно продолжительном, отрезке её функционирования. Ниже приведен фрагмент протокола интерактивного взаимодействия пользователя и ЭС *MYCIN* [147].

-----PATIENT-1-----

1) Patient's name?

David King

2) Age?

34

3) Sex?

Why?

[i.e. WHY is it important to determine the sex of PATIENT-1?]

[1.0] The sex of a patient is one of the 4 standard parameters that identify a patient. This may be useful later in the consultation. For example, it is relevant to determining whether an organism is normally found at this site and is relevant in determining PATIENT-1's most recent creatinine clearance (in ml/min).

[back to question 3]

Male

4) Race?

Caucasian

5) Are there any cultures for David King which may be related to the present illness, and from which organisms have been grown successfully in the microbiological laboratory?

Why?

[i.e. WHY is it important to determine organisms isolated from PATIENT-1?]

[2.0] This aid in determining the therapeutic regimen of PATIENT-1. if

[2.1] information has been gathered about organisms isolated from this patient, organisms noted on smears of this patient, negative cultures of this patient, suspected infections without microbiological evidence, current drugs of this patient and prior drugs of this patient, and

[2.2] an attempt has been made to deduce the organisms which require therapy, and

[2.3] you have given consideration to organisms (other than those seen in cultural smears) which might be present then determine the best therapy recommendation from among the drugs likely to be effective against the significant organisms, or indicate that no therapy is required at this time.

[RULE092]

[First, of course, it is necessary to ask about a positive culture]

[back to question 5]

Why?

[i.e. WHY is it important to determine the therapeutic regimen of PATIENT-1?]

[3.0] Deducing the therapeutic regimen of a patient is the goal of an infection disease consultation

No

6) Are there any pending cultures for David King?

Yes

-----CULTURE-1-----

7) From what site was the specimen for CULTURE-1 taken?

CSF

8) Please, give the date and time when the pending csf culture (CULTURE-1) was obtained (mo/da/yr time)

JUNE/24/84 7:00

9) Was a smear examined at the time that the specimen for the pending csf culture (CULTURE-1) was obtained?

Yes

- 10) Were any organisms seen on the smear of the specimen submitted as pending csf culture (CULTURE-1)?
No
- 11) Are there any other pending culture?
No
- 12) Have there been negative (as opposite to pending) cultures obtained from David King which may be relevant to the present illness?
No
- 13) Do you suspect David King may have an infection at a site from which you have not obtained culture specimens?
No
- 14) During the present illness, has David King been treated with an antimicrobial agent, which is no longer being administrated?
No
- 15) Has David King recently had symptoms of persistent headache or other abnormal neurological symptoms (dizziness, lethargy, etc.)?
Yes
- 16) Has David King recently had objective evidence of abnormal neurological sings (nuchal rigidity, coma, seizures, etc.) documented by physician observation or examination?
Yes

The CSF cultures will be considered to be associated with MENINGITIS.

- 17) Please, give the data on which clinical evidence (symptoms, signs, or laboratory tests) of the meningitis first appeared?
23/June/84 7:00

В приведенном фрагменте протокола диалога, информационные посылки пользователя записаны курсивом, а вопросы ЭС пронумерованы. Анализ приведенного фрагмента диалога позволяет сделать вывод, что если в нём удалить все запросы на вызов объяснительной компоненты в виде *WHY?* и следующие за ними объяснения в квадратных скобках, то оставшаяся часть является примером диалога с фиксированным распределением ролей, в котором ЭС *Mycin* играет роль активного агента.

Диалог в ЭС *Mycin*, как и в других ЭС, поддерживается специальным блоком, форма хранения данных в котором отлична от формы хранения фактов и правил в базе знаний [147]. Таким образом, для ЭС диалог является только формой общения (интерфейса) с пользователем и его не следует рассматривать как метод решения задачи. В случае эротетического диалога, сценарий, состоящий из диалогового метода доступа и памяти вопросов, напротив, является одновременно и формой хранения знаний, и формой общения с противоположным агентом.

Часто ЭС определяют как класс программ, имитирующих способность человеческого интеллекта делать вывод путём построения дедуктивных цепей, опираясь на знания, полученные в ходе диалогового процесса с пользователем

и хранящиеся в базе знаний [148]. В ЭС, основанных на правилах (продукционных ЭС), знания, необходимые для организации дедуктивного процесса, хранятся в виде фактов и связывающих их правил продукции. Факт, с точки зрения дедуктивного процесса, — это либо посылка, либо заключение, либо заключение-посылка, если он располагается внутри дедуктивной цепи. Компонент ЭС, осуществляющий логический вывод и называемый «машина вывода», используя внутреннюю логическую структуру правил продукции, производит формальный вывод. При проведении формального вывода машине вывода нет необходимости «понимать» смысл фактов и правил продукции, достаточно оперировать лишь именами фактов и знать, в каком они находятся отношении. Смысловая интерпретация полученного результата выполняется после завершения вывода.

Типы задач, решаемых при помощи ЭС, определяются двумя способами построения дедуктивной цепи, называемыми прямой вывод и обратный вывод [147]. Цель прямого вывода можно определить как переход от исходного множества фактов-предпосылок к результатному факту-заключению. При прямом выводе каждый из фактов исходного множества предпосылок связывается дедуктивными цепями с фактом-заключением.

Целью обратного вывода является переход от исходного факта-заключения к результатному множеству фактов-предпосылок. При обратном выводе факт-заключение связывается дедуктивными цепями с каждым из фактов множества предпосылок.

Таким образом, можно говорить о двух типах связей между фактами при проведении дедуктивных заключений:

1. Связи между фактами, задаваемые правилами продукции. Каждое правило связывает небольшое (обычно от 2-х до 4-х) количество фактов. Закона, позволяющего автоматически формировать правила продукции, не существует. Совокупность правил продукции отображает знания эксперта (или группы экспертов) о предметной области.
2. Связи между фактами, определяемые в процессе вывода при построении дедуктивных цепей. Дедуктивные цепи связывают между собой большое (десятки и сотни) количество фактов. Закон, управляющий формированием дедуктивных цепей, называется дедуктивным умозаключением [149].

Связи типа 1 существуют в базе знаний ЭС в явном виде и выражаются в описании продукционных правил, а связи 2-го типа — в неявном виде. Они формируются машиной вывода в процессе дедуктивного вывода. Однако, в литературе, посвященной ЭС, основанным на правилах, можно встретить *несколько способов явного представления связей 2-го типа*. Рассмотрим три из них:

- представление при помощи сети вывода (Inference Net) [150];
- представление при помощи дерева типа И/ИЛИ (AND/OR Tree) [151];
- представление при помощи дерева решения (Decision Tree) [152].

В основе того или иного способа явного представления дедуктивных связей 2-го типа лежит способ представления правила продукции. Существует множество нотаций и определений продукционного правила. Общепринятое представление продукционного правила в виде условного предложения приведено в [150].

$$\begin{array}{l}
 IF \quad condition1 \\
 \quad \quad condition2 \\
 \quad \quad \dots \\
 THEN \\
 \quad \quad action
 \end{array}
 \tag{5.1}$$

В условном предложении (5.1) *condition1*, *condition2* и т. д. обозначают факты предпосылки, которые могут интерпретироваться как некоторые условия, а *action* — факт-заключение, который может интерпретироваться как некоторое действие, выполняемое в том случае, когда имеют место все факты предпосылки. Таким образом, факты предпосылки в первой части продукционного правила связаны логической операцией конъюнкции. Это обстоятельство в ряде случаев позволяет моделировать продукционное правило при помощи импликации».

$$\begin{array}{ll}
 IF & \text{факты антецедента} \\
 THEN & \text{факт-консеквент}
 \end{array}
 \tag{5.2}$$

В (5.2) под фактами антецедента и фактом-консеквентом понимаются факты-посылки и факт-заключение, соответственно.

Графическое представление дедуктивных связей 2-го типа чаще всего базируется на графическом представлении отдельного продукционного правила, понимаемого в смысле (5.2). На рис. 5.2 приведены графические представления продукционного правила, используемые для изображения Inference Net (а) и AND/OR Tree (b). Используя графические символы правил продукции, приведенные на рис. 5.2, можно графически представить весь процесс дедуктивного вывода.

Проиллюстрируем на примере, каким образом графические представления правила продукции, приведенные на рис. 5.2, могут быть использованы для графического представления дедуктивных связей 2-го типа при помощи сети вывода и дерева типа И/ИЛИ. Используем для этой цели базу знаний, состоящую из шести правил, которые могут быть использованы для классификации млекопитающих. Этот набор правил приведен в таблице на рис. 5.3.

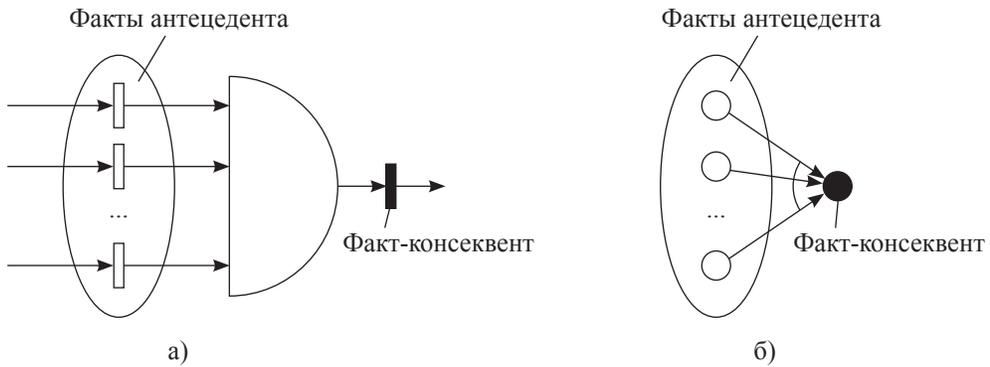


Рис. 5.2. Графическое представление правила продукции, используемое для изображения: а) сети вывода (Inference Net); б) дерева типа И/ИЛИ (AND/OR Tree)

R_1	<i>IF</i>	животное покрыто шерстью
	<i>THEN</i>	это млекопитающее
R_2	<i>IF</i>	животное дает молоко
	<i>THEN</i>	это млекопитающее
R_3	<i>IF</i>	животное млекопитающее имеет остроконечные зубы имеет когти глаза смотрят вперед
	<i>THEN</i>	это плотоядное животное
R_4	<i>IF</i>	животное млекопитающее имеет копыта
	<i>THEN</i>	это копытное животное
R_5	<i>IF</i>	животное плотоядное имеет темно-желтый цвет имеет черные полосы
	<i>THEN</i>	это тигр
R_6	<i>IF</i>	животное копытное имеет белый цвет имеет черные полосы
	<i>THEN</i>	это зебра

Рис. 5.3. Продукционные правила, используемые в примере представления дедуктивных связей при помощи сети вывода и дерева типа И/ИЛИ

На рис. 5.4. приведен пример графического представления дедуктивных связей 2-го типа в виде сети вывода для базы знаний, приведенной на рис. 5.3.

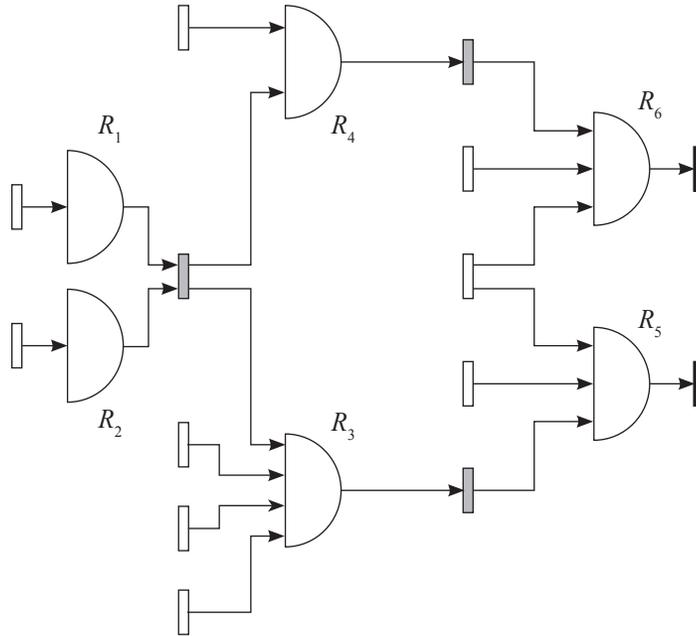


Рис. 5.4. Пример сети вывода для продукционных правил, приведенных на рис. 5.3.

На рис. 5.4 светлые прямоугольники соответствуют исходным фактам (факты антецедента). Темные прямоугольники соответствуют фактам, получаемым в процессе прямого вывода (факты консеквента). Серым цветом выделены факты, которые одновременно являются и фактами антецедента, и фактами консеквента. Это факты, формируемые правилами R_1 , R_2 , R_3 , R_4 .

На рис. 5.5 приведен пример графического представления дедуктивных связей 2-го типа в виде дерева типа И/ИЛИ.

На рис. 5.5 светлые кружки соответствуют исходным фактам (факты антецедента). Темные кружки соответствуют фактам, полученным в процессе вывода (факты консеквента), а серые кружки — промежуточным фактам, являющимся одновременно и фактами антецедента, и фактами консеквента. Пунктирной линией объединены тождественные факты антецедента для правил R_1 , R_2 . Дерево типа И/ИЛИ содержит два типа узлов. Узел типа И отмечен дугой и соответствует продукционному правилу с несколькими фактами антецедента. Узел типа ИЛИ соответствует факту консеквента, являющегося общим для нескольких продукционных правил.

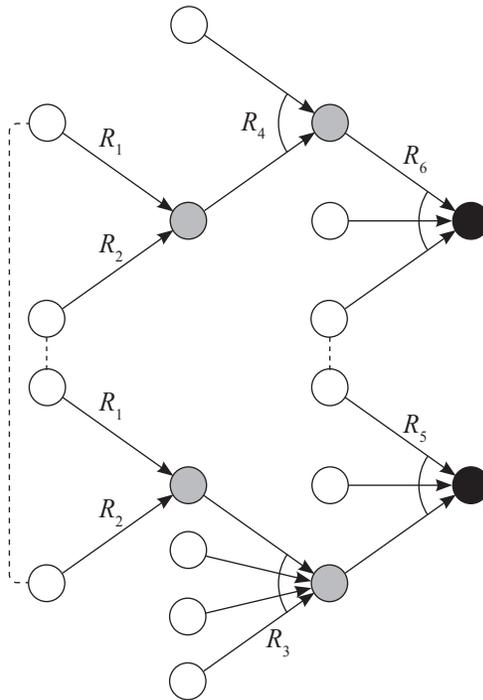


Рис. 5.5. Пример дерева типа И/ИЛИ для продукционных правил, приведенных на рис. 5.3.

В [152] для графического представления связанности фактов в базе знаний используется диаграмма, названная деревом решения (Decision Tree). Этот вид графического представления связанности фактов отличается от только что рассмотренных сети вывода и дерева типа И/ИЛИ тем, что в нём явно не представлены графические символы продукционных правил. Дерево решений представляет собой сеть, в которой имеется два типа узлов, изображаемых прямоугольниками и овалами. Узлам, которые изображаются при помощи овалов, соответствуют вопросы ЭС, а узлам, которые изображаются прямоугольниками — заключения. Ветви, соединяющие узлы, надписаны ожидаемыми ответами пользователя на вопросы ЭС.

На рис 5.6 приведен пример дерева решений. Направление стрелок соответствует прямому выводу.

Пример дерева решений, приведенный на рис. 5.6, иллюстрирует базу знаний экспертной системы, предназначенной для принятия решений о предоставлении или не предоставлении должности соискателю. Система также определяет, какую именно должность можно предложить.

Несложно заметить, что дерево решений похоже на сетевую модель диалогового метода доступа (см. подраздел 3.3) и является его упрощенным вариантом.

Дерево решений, приведенное на рис. 5.6, можно представить в виде сети *DiAM*, которая состоит из узлов прямой навигации типа *Node1* и целевых узлов типа *Node3*. Узлам типа *Node1* соответствуют вопросы и ожидаемые на них ответы, а узлам типа *Node3* соответствуют заключения. Напомним, что с узлом типа *Node3* связано утверждение, а не вопрос.

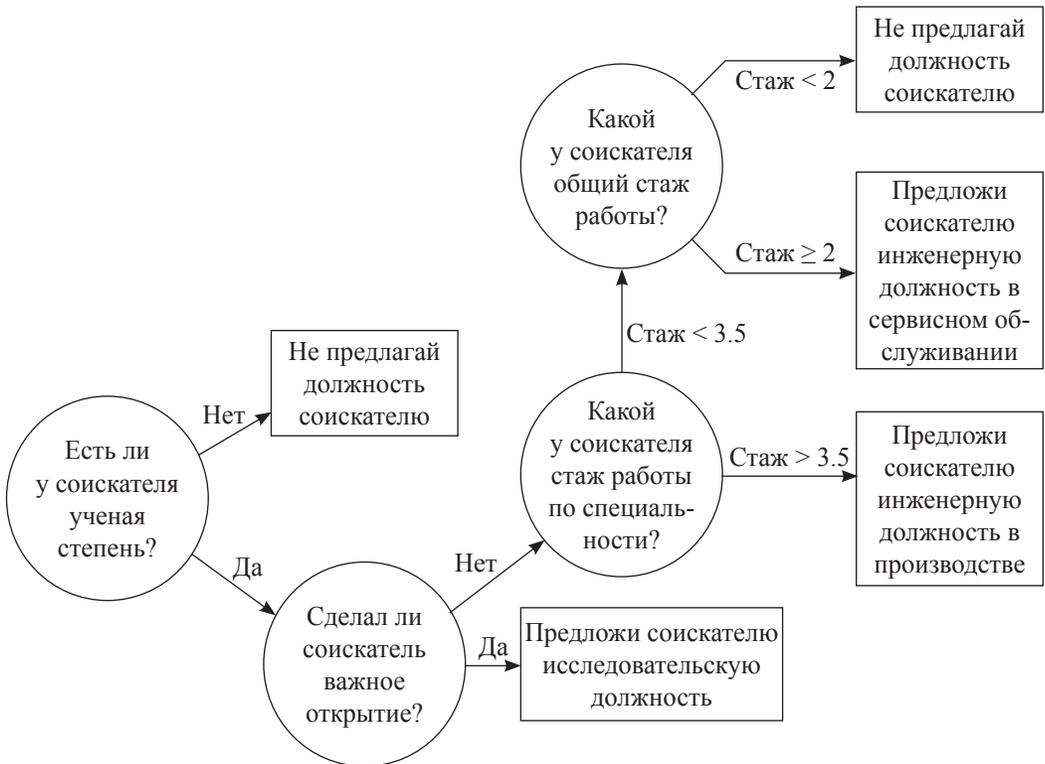


Рис. 5.6. Пример дерева решений для случая прямого вывода

Дерево решений так же, как и дерево типа И/ИЛИ или сеть вывода, является одним из способов графического представления дедуктивных связей 2-го типа в базе знаний. Дерево решений может быть представлено в виде базы знаний, состоящей из набора продукционных правил с неявным представлением дедуктивных связей 2-го типа. Простая процедура позволяет выделить для дерева решений такой набор правил путём прохождения всех возможных траекторий от корня дерева (первый вопрос в левой части рис. 5.6) до одного из заключений. В таблице на рис. 5.7 приведен набор продукционных правил, выделенных из дерева решений, приведенного на рис. 5.6, путём прохождения всех возможных траекторий от корня дерева до всех возможных заключений.

R_1	<i>IF</i>	соискатель не имеет ученой степени
	<i>THEN</i>	не предлагай должность соискателю
R_2	<i>IF</i>	соискатель имеет ученую степень
	<i>THEN</i>	узнай, сделал ли соискатель важное открытие
R_3	<i>IF</i>	соискатель сделал важное открытие
	<i>THEN</i>	предложи соискателю исследовательскую должность
R_4	<i>IF</i>	соискатель не сделал важное открытие
	<i>THEN</i>	узнай, какой у соискателя стаж работы по специальности
R_5	<i>IF</i>	стаж работы по специальности у соискателя более, чем 3,5 года
	<i>THEN</i>	предложи соискателю инженерную должность в производстве
R_6	<i>IF</i>	стаж работы по специальности у соискателя менее, чем 3,5 года
	<i>THEN</i>	узнай, какой у соискателя общий стаж работы
R_7	<i>IF</i>	стаж работы у соискателя меньше, чем 2 года
	<i>THEN</i>	не предлагай должность соискателю
R_8	<i>IF</i>	стаж работы у соискателя больше, чем 2 года или равен 2 годам
	<i>THEN</i>	предложи соискателю инженерную должность в сервисном обслуживании

Рис. 5.7. Продукционные правила, выделенные из дерева решений, изображенного на рис. 5.6

Таким образом, в экспертной системе, основанной на правилах, все факты базы знаний неявно связаны в сеть возможными цепями вывода. Эти неявные связи задаются правилами вывода и явно устанавливаются машиной вывода в процессе работы экспертной системы. Продолжая эту линию рассуждений, можно утверждать, что существует некоторое фиксированное число M , равное наибольшему количеству возможных выводов для данной фиксированной базы знаний. С этой точки зрения, «новые» факты, получаемые в ходе вывода, на самом деле должны находиться в базе знаний, а процесс вывода лишь обеспечивает к ним доступ.

5.3. Дедуктивные связи в модели диалогового метода доступа к памяти вопросов

В ЭС, основанной на правилах, формирование явных связей между фактами в процессе вывода может происходить либо при прямом, либо при обратном выводе. Как при прямом, так и при обратном выводе в рабочей памяти ЭС накапливаются факты. Отличие прямого вывода от обратного вывода заключается в интерпретации фактов, накопленных в рабочей памяти. После завершения процесса вывода набор фактов, накопленных в рабочей памяти, можно разделить на два класса: *промежуточные факты* и *терминальные факты*.

Введём следующие обозначения:

AFact — факт антецедента некоторого продукционного правила;

CFact — факт консеквента некоторого продукционного правила.

Тогда в процессе прямого вывода накапливаемые в рабочей памяти ЭС факты интерпретируются как факты антецедента *AFact*, а конечная цель прямого вывода — это получение терминального факта консеквента *CFact*.

В процессе обратного вывода факты, накапливаемые в рабочей памяти, интерпретируются как факты консеквента *CFact*, а конечная цель обратного вывода — получение терминального списка фактов антецедента *AFact*. Сеть вывода, приведенная на рис. 5.8, иллюстрирует описанное понимание процессов вывода, осуществляемых продукционной ЭС.

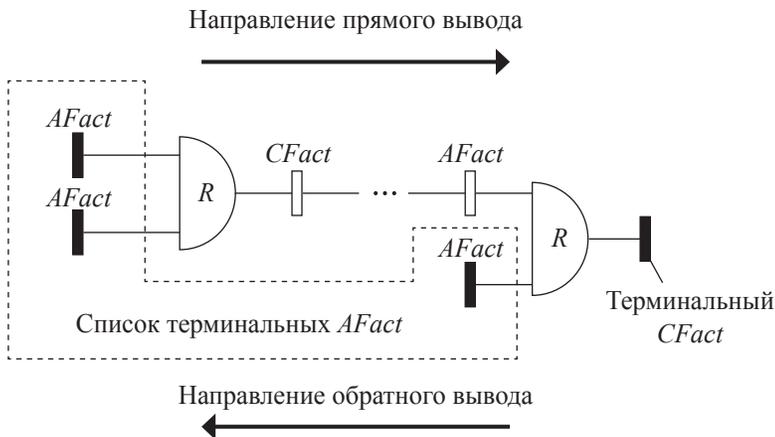


Рис. 5.8. Сеть вывода, иллюстрирующая понятия промежуточного и терминального фактов

Любой промежуточный факт является одновременно и фактом антецедента (*AFact*), и фактом консеквента (*CFact*). Промежуточный факт является фактом

консеквента по отношению к предыдущему правилу и фактом антецедента по отношению к последующему правилу. Промежуточные факты располагаются «внутри» дерева вывода. Их количество может быть достаточно большим, и они необходимы только для того, чтобы, в конце концов, получить терминальные факты. Ясно, что количество промежуточных фактов детерминирует длительность процесса вывода, поэтому желательно, чтобы в каждом конкретном выводе их количество было наименьшим.

Терминальные факты — это совокупность целевых $AFact$ или $CFact$. Они располагаются «по краям» дерева вывода и определяют искомое решение задачи.

После завершения процесса вывода и формирования сети вывода вывод может быть представлен без промежуточных фактов в виде отображения терминального списка фактов антецедента в терминальный факт консеквента (для случая прямого вывода) или в виде отображения терминального факта консеквента в терминальный список фактов антецедента (для случая обратного вывода).

$$\{AFact_i\}, i = 1, \dots, n \rightarrow CFact, \quad \text{прямой вывод,} \quad (5.3)$$

$$CFact \rightarrow \{AFact_i\}, i = 1, \dots, n, \quad \text{обратный вывод,} \quad (5.4)$$

где $\{AFact_i\}$ — терминальный список фактов антецедента,

$CFact$ — терминальный факт консеквента.

Таким образом, после того, как некоторый прямой вывод был однажды проведен, соответствующая цепь вывода в сети вывода может быть «свёрнута» в одно эквивалентное правило продукции вида (5.5), включающего только терминальные факты. Затем правило вида (5.5) может быть помещено в базу знаний. В этом случае, когда возникнет необходимость провести точно такой же вывод, ЭС сможет воспользоваться правилом типа (5.5) вместо повторного построения сети вывода.

$$\begin{array}{l} IF(AFact_1 \& AFact_2 \& \dots AFact_n) \\ THEN CFact \end{array} \quad (5.5)$$

Правило вида (5.5) позволяет завершить вывод за n шагов, если имеется метод, позволяющий на каждом шаге получить точно один $AFact$. Последующая часть настоящего подраздела посвящена тому, чтобы показать, каким образом эротетический диалоговый процесс может быть использован в качестве отмеченного выше метода.

Как было отмечено в 5.2, для фиксированной базы знаний количество возможных выводов ограничено сверху и существует не более M возможных выводов, которые можно моделировать не более, чем M правилами вида

(5.5). Представим конъюнкцию в левой части (5.5) цепью из n последовательно соединённых элементов переключающего типа, каждый из которых соответствует $AFact$ (по аналогии с известным формальным представлением переключающих схем [153]). Для этой цели используем Петри-модель $DiAM$. На рис. 5.9 правило вида (5.5) приведено в графическом изображении в виде фрагмента сети $DiAM$ (направление стрелок совпадает с направлением прямого вывода).

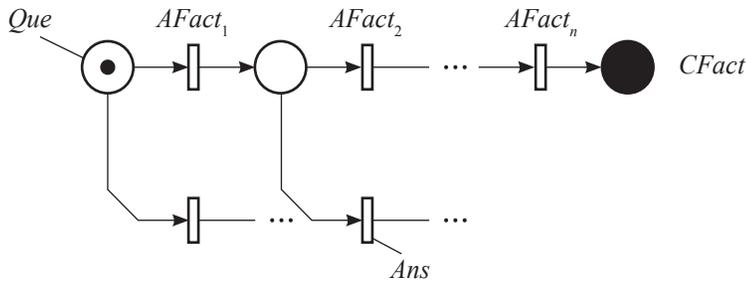


Рис. 5.9. Фрагмент сети $DiAM$, соответствующий правилу вида (5.5)

Будем считать, что факту в ЭС, основанной на правилах, в диалоговой системе соответствует ответ на вопрос (см. пример дерева вывода, приведенный на рис. 5.6). Поэтому на рис. 5.9 факты моделируются переходами. Вопрос, с точки зрения логического вывода, играет второстепенную роль. Он предоставляет реактивному агенту через свой субъект эксплицитное множество родственных и альтернативных фактов. Например, место отдыха — горы, место отдыха — пляж, место отдыха — Петербург. Отвечая на вопрос, реактивный агент выбирает один из альтернативных фактов субъекта вопроса, формируя, таким образом, очередной конъюнкт в antecedенте (5.5).

Очевидно, что если фрагмент сети $DiAM$, приведенный на рис. 5.9, эквивалентен одному дереву вывода, то вся сеть $DiAM$ эквивалентна всем M выводам, возможным для данной базы знаний. Движение фишки по сети моделирует пошаговый процесс вывода, а сам вывод завершается за n шагов-транзакций, где $n + 1$ — количество терминальных фактов в дереве вывода. На каждой транзакции в antecedент (5.5) добавляется ровно один факт. Тип вывода (прямой или обратный) определяется интерпретацией узлов $DiAM$. Если узел, соответствующий $CFact$, завершает цепь вывода, то имеет место прямой вывод. Если с узла, соответствующего $CFact$, начинается цепь вывода, то имеет место обратный вывод.

5.3.1. Когнитивные основы дедуктивных связей в модели диалогового метода доступа

Одно из ключевых положений, сформулированных в 5.3, заключается в том, что после завершения процесса вывода и формирования дерева вывода последнее может быть «свёрнуто» и заменено одним правилом продукции вида (5.5), включающим только терминальные факты. Это положение имеет корни в теории автоматических и осознанных психологических процессов [65, 66].

Интроспекция и экспериментальные данные показывают, что многократное повторение и практика позволяют резко улучшить временные и качественные показатели, определяющие поведенческую активность человека. С точки зрения когнитивной психологии, объяснение этого феномена заключается в том, что многократное решение одной и той же задачи переводит процесс её решения из разряда процессов, контролируемых сознанием, в автоматически выполняемые процессы. К общепринятым характеристикам автоматических процессов относятся следующие [65]:

- автоматические процессы выполняются быстрее, чем процессы, контролируемые сознанием;
- автоматические процессы не уменьшают ментальные ресурсы, необходимые для решения других задач (поскольку не требуют привлечения механизма внимания);
- автоматические процессы не подвластны сознанию;
- автоматические процессы неотвратимы (они безусловно выполняются, если на организм воздействует соответствующий стимул, даже если на этом стимуле не сфокусировано внимание).

Шнейдер и Шифрин [154, 155] сформулировали основные различия между автоматическими процессами и процессами, контролируемыми сознанием:

- процессы, контролируемые сознанием, используют фиксированный и ограниченный ресурс сознания, базируются на механизме внимания и легко адаптируются к новым условиям;
- автоматические процессы не зависят от имеющихся ресурсов сознания, не требуют привлечения механизма внимания, но их трудно модифицировать после того, как они были сформированы и запомнены.

Практика и многократное повторение трансформируют процесс, контролируемый сознанием в автоматический процесс, и такая трансформация сопровождается реструктуризацией схемы, соответствующей данному процессу [156].

На практике, однако, не все процессы могут быть чётко разделены на два класса в соответствии с описанными выше признаками. Существует множество примеров процессов, являющихся автоматическими только частично. Норман и Шалис [171] отмечают, что дихотомическая классификация ментальных процессов упрощает реальность, поскольку не учитывает процессы, занимающие промежуточное

положение. Вместо простой дихотомии процессов на автоматические и управляемые сознанием они предложили три уровня автоматизма ментальных процессов:

- полностью автоматический процесс, управляемый схемой;
- частично автоматический процесс, вовлекающий процедуру разрешения конфликтов схем, которая осуществляется без преднамеренного вмешательства со стороны сознания;
- процесс, находящийся под полным контролем механизма внимания.

Согласно Норман и Шалис, полностью автоматический процесс детерминируется одной схемой, хранящейся в долговременной памяти, и реализуется действительно автоматически с очень малым его осознанием индивидуумом.

В ряде случаев существуют несколько схем, которые могут использоваться для управления процессом. Тогда необходим предварительный этап выбора одной из схем. Норман и Шалис ввели в свою модель специальную контролирующую систему, используемую для осуществления отмеченного выбора. Эта система осуществляет автоматическое разрешение конфликта схем и имеет наименование система конкурирующего планирования (*contention scheduling*). Частично автоматические процессы осознаются индивидом в большей степени, чем полностью автоматические.

Наконец, существует управляющая система, находящаяся на наивысшем уровне управления, и являющаяся, по сути, механизмом внимания (*supervisory attentional system*). Эта система привлекается во всех случаях, когда решение задачи или принятие решения осуществляется впервые. Она отличается наибольшей гибкостью и приспособляемостью к условиям решаемой задачи.

Таким образом, в отличие от модели Шнейдера и Шифрина, модель Норман и Шалис предполагает наличие двух систем управления ментальными процессами: системы конкурирующего планирования и механизма внимания.

Логан [157] предложил теорию, объясняющую, каким образом многократное повторение ментальных процессов переводит их в категорию автоматических. Теория Логана базируется на следующих принципах.

- Начальная версия схемы в виде отдельных «траекторий» формируется в долговременной памяти при первоначальном решении задачи.
- Повторное решение той же задачи ведёт к «разрастанию» схемы и увеличению информации как о самой задаче, так и о методах её решения.
- Схема с увеличенным объёмом хранимой информации приобретает свойство быть быстро найденной в случае необходимости (повторного решения задачи).
- «Автоматизм — это поиск в долговременной памяти: обработка является полностью автоматической, когда она базируется на одношаговом и прямом доступе к прошлому решению в памяти» [157].
- При однократном решении задачи она воспринимается как новая, и её решение формируется с привлечения механизма внимания; после того, как

задача решалась многократно, решение сохраняется в долговременной памяти и может быть получено очень быстро.

Процессы решения задачи интеллектуальными агентами путём поиска в пространстве состояний либо путем логического вывода и принятия решений имеют много общих черт с описанными когнитивными теориями и могут моделироваться аналогичными системами или методами. Интеллектуальные агенты, осуществляющие автоматическое решение задач, могут базироваться на моделях соответствующих ментальных функций. Поэтому такие системы должны «поразному» подходить к первоначальному и повторному решению одной и той же задачи. Внешние отличия должны проявляться в том, что в случае первоначально решаемой задачи агент использует существенное количество ресурсов компьютера (предположительно существенный объём адресного пространства основной памяти), а решение задачи занимает существенное время. Повторно решаемая задача должна решаться быстрее и использовать незначительные ресурсы.

Если, с описанной точки зрения, проанализировать «сворачивание» дерева вывода в правило вида (5.5), то такой подход можно рассматривать как отражение в компьютерной системе феномена дихотомии автоматических и осознанных ментальных процессов.

Задача, сводящаяся к нахождению решения путём логического вывода и решаемая впервые, порождает построение дерева вывода. Это ресурсоёмкий и длительный процесс (аналогичный процессу, контролируемому сознанием), поскольку все используемые факты, как терминальные, так и промежуточные, должны быть использованы и «встроены» в дерево вывода. Ясно, что нерационально повторять этот процесс каждый раз, когда необходимо повторно найти решение той же задачи. Если после каждого решения задачи система в состоянии создать правило вида (5.5) и запомнить его в *DiAM* во внешней памяти, то повторное решение задачи может быть получено за минимальное время — n шагов диалогового процесса. Процесс нахождения решения при помощи правила вида (5.5) является аналогом автоматического процесса. Продолжая эту аналогию, можно рассматривать *DiAM* как модель долговременной памяти, в которой накоплен «прошлый опыт» решения задач.

В унифицированной когнитивной модели *SOAR* предложен похожий механизм «сворачивания» знаний [135]. *SOAR (State Operator and Result)* представляет собой систему, моделирующую процессы решения задач и принятия решений человеком. Центральная архитектура *SOAR* моделирует механизмы центральной переработки и не включает восприятие и моторное поведение. Ниже перечислены основные принципы, характеризующие центральную архитектуру *SOAR*.

1. Решение задачи разыскивается не в одном, а в нескольких проблемных пространствах.
2. Знания, хранимые в долговременной памяти, моделируются набором продукционных правил.

- 3. Поиск решения путем логического вывода основан на процедуре, использующей систему предпочтений (приоритетов).
- 4. Переход от текущего проблемного пространства к последующему связан с появлением тупиковой ситуации и изменением цели в процессе решения задачи.
- 5. При разрешении тупиковой ситуации автоматически создается новое продукционное правило.

Центральная архитектура *SOAR* генерирует цели и подцели динамически, по мере решения задачи. Генерирование очередной подцели не запланировано явно, а происходит только в том случае, когда *SOAR* попадает в тупиковую ситуацию. Тупиковой ситуацией называется такая ситуация, когда процедура принятия решений *SOAR* не в состоянии сделать выбор и *SOAR* не может продолжать решение задачи. Возникновение тупиковой ситуации не означает остановку процесса решения задачи. *SOAR* находит выход из тупика путем динамической генерации подцели. Более того, тупик — это единственная причина, приводящая к генерации подцели. Рис. 5.10 иллюстрирует цикл принятия решений *SOAR*.

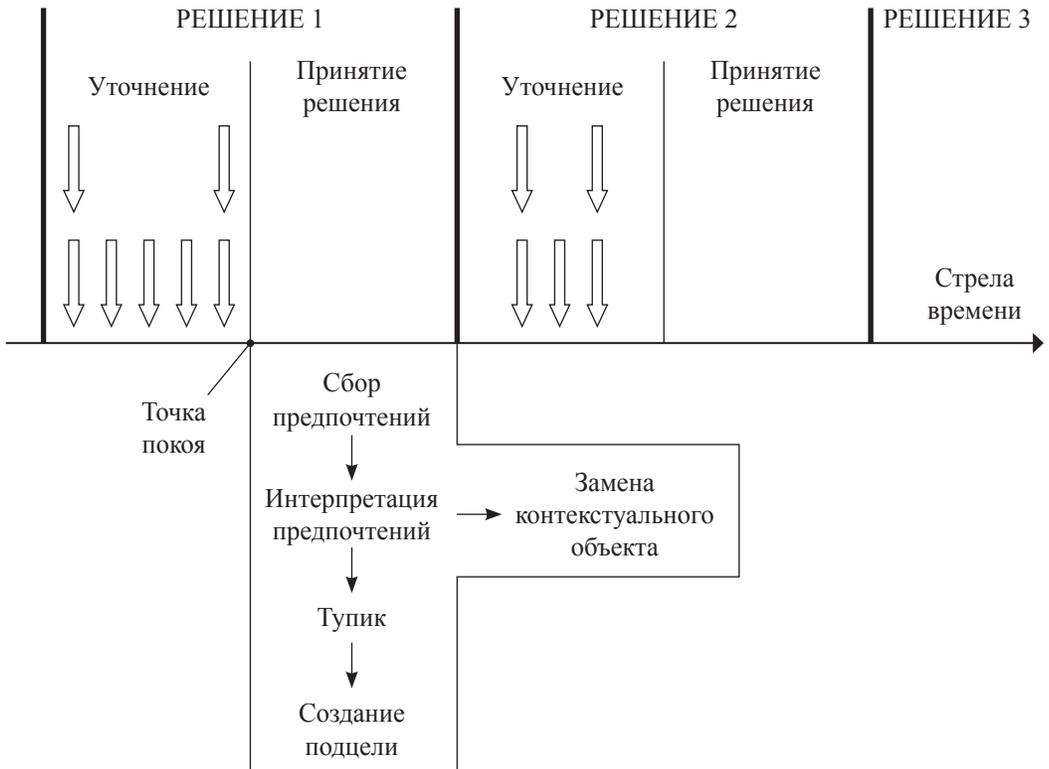


Рис. 5.10. Цикл принятия решений *SOAR*

Автоматическая генерация продукционных правил является способом конвертирования процесса решения проблемы в единственное правило продукций. При генерации правила продукции вновь полученные результаты используются для формирования консеквента, а элементы рабочей памяти — для формирования антецедента. Рис. 5.11 иллюстрирует процесс автоматической генерации продукционных правил в *SOAR*.

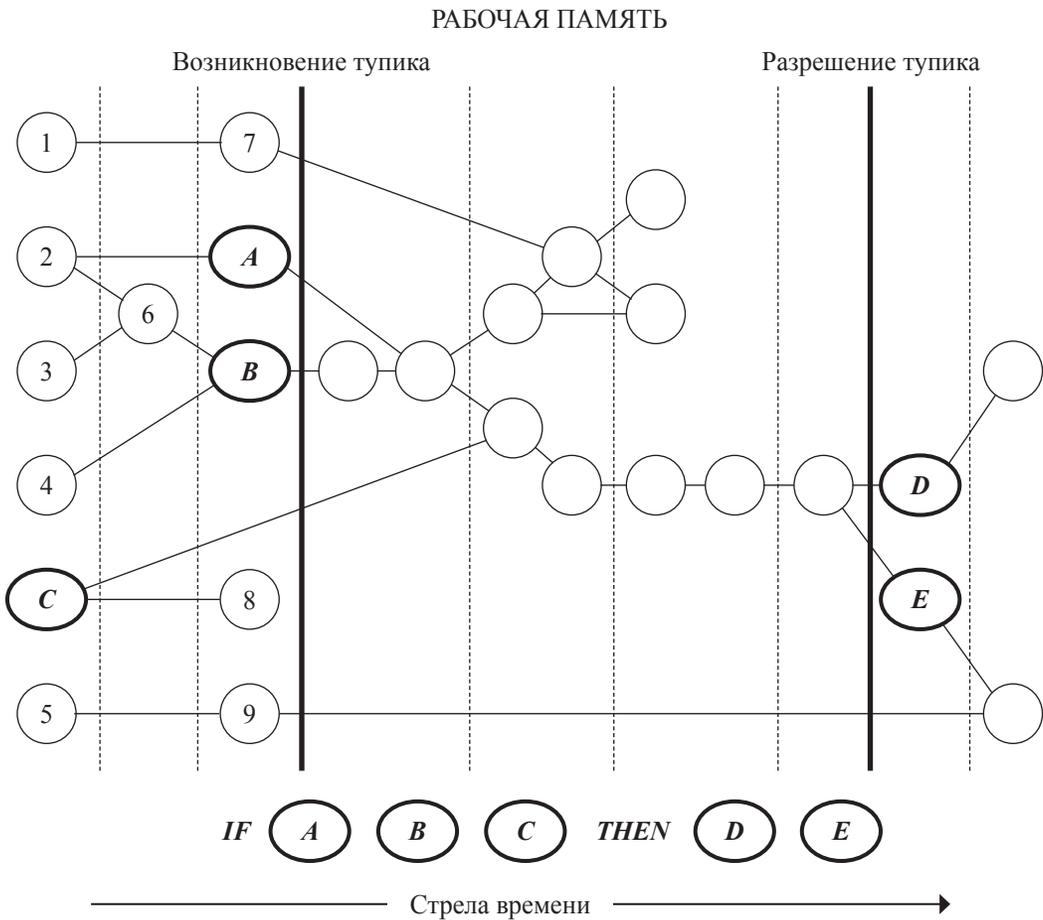


Рис. 5.11. Диаграмма, иллюстрирующая процесс автоматического создания продукционного правила в *SOAR*

Окружности и овалы на рис. 5.11 соответствуют фактам рабочей памяти, образующимся при активизации продукций. Каждый факт рабочей памяти создается правилом продукции, антецедент которого был удовлетворён текущим содержанием рабочей памяти. Эта каузальная связь на рис. 5.11 представлена

линиями, соединяющими факты. Например, факт **B** является результатом активизации некоторой продукции, у которой в антецеденте фигурируют факты 4 и 6. Циклы принятия решений на рис. 5.11 отделены друг от друга вертикальными пунктирными линиями. Момент возникновения тупиковой ситуации отмечен левой сплошной вертикальной линией. При возникновении тупика *SOAR* порождает новую подцель и пополняет рабочую память новым контекстом. Последующие циклы принятия решений направлены на достижение подцели или разрешения возникшей тупиковой ситуации. Предположим, что, в конце концов, в рабочую память попадают два факта **D** и **E**, которые позволяют достичь подцели и разрешить тупиковую ситуацию. Последующий цикл принятия решений возвращается к контексту предыдущего уровня, в котором ранее возникла тупиковая ситуация. На рис. 5.11 этот момент отмечен правой сплошной вертикальной линией.

В момент разрешения тупиковой ситуации осуществляется автоматическое создание нового продукционного правила. Консеквент нового правила формируется из полученного результата — элементов **D** и **E**, а антецедент — из трех элементов **A**, **B** и **C**, которые привели к полученному результату и существовали до момента возникновения тупика. Вновь созданная продукция пополняет содержимое долговременной памяти. Ясно, что вновь созданная продукция исключает возникновение тупиковой ситуации, описанной выше, поскольку теперь *SOAR* «знает», что наличие в рабочей памяти, кроме всего прочего, трех элементов **A**, **B** и **C** должно сопровождаться действиями **D** и **E**. Таким образом, автоматическое создание продукционных правил может рассматриваться как форма обучения *SOAR* на основании приобретенного опыта. Результат решения проблемы запоминается в виде, позволяющем, в случае повторного возникновения такой же проблемы, не решать её заново, а использовать полученный ранее результат.

5.4. Формальное описание дедуктивных связей в модели диалогового метода доступа

5.4.1. Логический вывод в сети диалогового метода доступа

С целью получения формализованного описания дедуктивных связей в диалоговой модели, рассмотрим пример трансформации базы знаний ЭС, основанной на правилах, в сеть *DiAM*. Приведенный ниже пример описан в [148]. Ещё несколько примеров такой трансформации приведено в [158]. Пусть база знаний продукционной ЭС состоит из некоторого небольшого количества

фактов и правил. Представим набор фактов этой базы знаний в виде следующих пропозиций

$$\begin{array}{l}
 a \text{ намерение — отдых;} \\
 b \text{ дорога ухабистая;} \\
 c \text{ использовать джип;} \\
 d \text{ место отдыха — горы;} \\
 e \text{ место отдыха — пляж;} \\
 f \text{ нужна скорость.}
 \end{array}
 \left. \vphantom{\begin{array}{l} a \\ b \\ c \\ d \\ e \\ f \end{array}} \right\} \quad (5.6)$$

Представим правила, связывающие факты (5.6), в виде

$$\begin{array}{l}
 1 \text{ IF } a \ \& \ b \ \text{ THEN } c; \\
 2 \text{ IF } d \ \text{ THEN } b; \\
 3 \text{ IF } a \ \text{ THEN } f; \\
 4 \text{ IF } e \ \text{ THEN } b.
 \end{array}
 \left. \vphantom{\begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array}} \right\} \quad (5.7)$$

Правила (5.7) можно рассматривать как некоторые элементарные схемы вывода. Например, правило 1 позволяет вывести заключение c при наличии посылок a и b . Правила (5.7) можно использовать для построения деревьев вывода, допустимых для рассматриваемой базы знаний. На рис. 5.12 изображено несколько деревьев вывода, соответствующих правилам (5.7), с использованием нотации дерева типа И/ИЛИ.

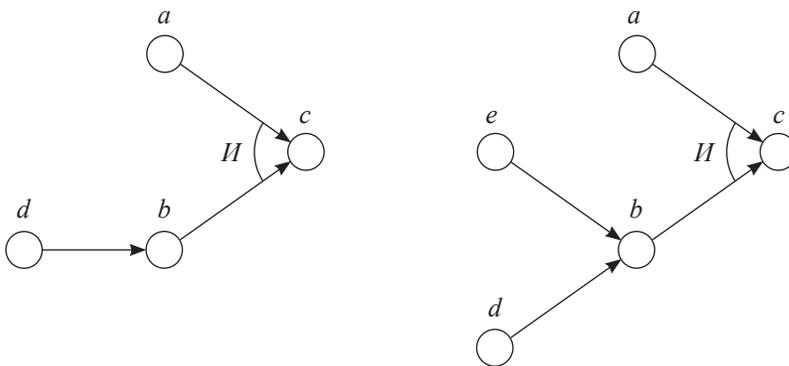


Рис. 5.12. Примеры деревьев вывода для базы знаний (5.6) и (5.7)

Правило 3 введено в базу знаний для иллюстрации потенциальной конфликтности набора правил, подобранных случайным образом. Ясно, что при наличии в рабочей памяти одновременно фактов a и b возможны как вывод c (правило 1), так и вывод f (правило 3). Для разрешения конфликтных ситуаций, возникающих в

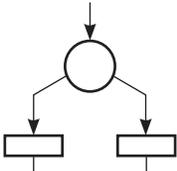
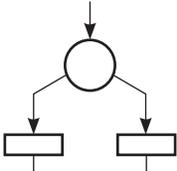
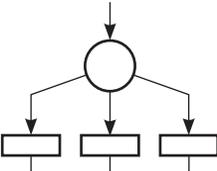
процессе вывода в ЭС, основанной на правилах, используются специальные методы, в основе которых лежат знания внешнего (по отношению к базе знаний) интеллекта, способного оценить «стоимость перехода», «коэффициент доверия» и т. п. Оценки вводятся извне и определяются либо путём расчётов, либо внешними экспертами. В [159] приведен краткий обзор наиболее известных методов разрешения конфликтных ситуаций. Отметим, что процесс снабжения правил продукции коэффициентами доверия можно рассматривать как неявный (скрытый) диалог с экспертом или как опрос эксперта, проведенный заранее при формировании базы знаний.

Для заполнения рабочей памяти ЭС можно использовать эротетический диалоговый процесс, в котором фактам соответствуют элементы субъектов вопросов активного агента. На каждом текущем шаге вывода активный агент передает реактивному агенту вопрос, субъект которого содержит потенциальные факты-ответы, ожидаемые на текущем шаге.

В табл. 5.2 в первой колонке приведен список фактов (5.6). Во второй колонке приведен соответствующий список ли-вопросов. Субъект каждого вопроса содержит соответствующий факт. В третьей колонке приведено графическое изображение соответствующей транзакции сценария с использованием нотации Петри-модели *DiAM*.

Таблица 5.2

Отображение фактов базы знаний ЭС, основанной на правилах, в соответствующие узлы *DiAM* для примера, описанного в тексте

Факты базы знаний, приведенные в (5.6)	Вопросы, ответы на которые порождают соответствующие факты	Графическое изображение узла <i>DiAM</i>
<i>a</i> намерение — отдых	Что вы намерены делать: отдыхать или работать?	
<i>b</i> дорога ухабистая	Какая ожидается дорога: ухабистая или ровная?	
<i>d</i> место отдыха — горы <i>e</i> место отдыха — пляж	Где вы собираетесь отдыхать: в горах, на пляже или где-нибудь ещё?	

В табл. 5.2 вопросы сформулированы таким образом, что в них факт, необходимый для очередного шага вывода, присутствует в качестве одного из элементов списка его субъекта. Требования всех вопросов предполагают альтернативность элементов списка субъекта.

При трансформации фактов в соответствующие узлы *DiAM* факты *d* и *e* включены в субъект одного вопроса. Несмотря на то, что в базе знаний эти факты представлены как отдельные пропозиции, с точки зрения логики эротетических транзакций, они представляют собой элементы списка субъекта одного и того же вопроса.

Представление фактов в виде узлов сети *DiAM*, приведенных в табл. 5.2, позволяет построить сеть *DiAM*, эквивалентную всем выводам, возможным для базы знаний, описанных набором фактов (5.6) и набором правил (5.7). Фрагмент такой сети изображен на рис. 5.13.

На рис. 5.13 целевая позиция диалогового процесса обозначена зачерненной вершиной. Для целевой позиции нарушается смысловая интерпретация позиций и переходов (напомним, что согласно Петри-модели, переход соответствует факту-ответу, а позиция — вопросу). Целевая позиция в случае прямого вывода содержит факт-заключение (*CFact*) и, возможно, очередной вопрос. При изображении сети на рис. 5.13 предполагалось, что после получения заключения и положительного ответа на вопрос: «Будете продолжать работу?» сценарий системы возвращается в начальное состояние.

Сопоставляя способы представления знаний при помощи сети *DiAM* и при помощи продукционной базы знаний, принятой для ЭС, основанной на правилах, можно сделать следующие выводы.

Сеть *DiAM* (даже в случае прямого вывода) имеет одну точку вход, соответствующий началу диалогового процесса, а дерево вывода (представленное, например, в виде дерева типа И/ИЛИ либо в виде сети вывода) имеет несколько точек входа. Наличие множества точек входа дерева вывода затрудняет или даже исключает совмещение процесса логического вывода с диалоговым процессом в рамках классической ЭС, основанной на правилах.

В сети *DiAM* процесс получения заключения (процесс вывода) сводится к процессу навигации на сети *DiAM*, управляемому реактивным агентом. Таким образом, один из ключевых компонентов архитектуры ЭС — машина вывода — трансформируется в машину навигации, функции которой реализуются стандартным процессом *DiMC* совместно с процессами-демонами (см. подраздел 3.3).

Привлекательным свойством хранения знаний в виде сети *DiAM*, а не в виде набора фактов и правил, является принципиальное отсутствие конфликтных ситуаций в процессе вывода-навигации. Отмеченное свойство является следствием естественного ограничения, существующего в эротетическом диалоговом процессе и отраженного в сети *DiAM*. Ограничение заключается в том, что при

ведении эротетического диалога на любой текущий ответ предполагается формирование только одного последующего вопроса.

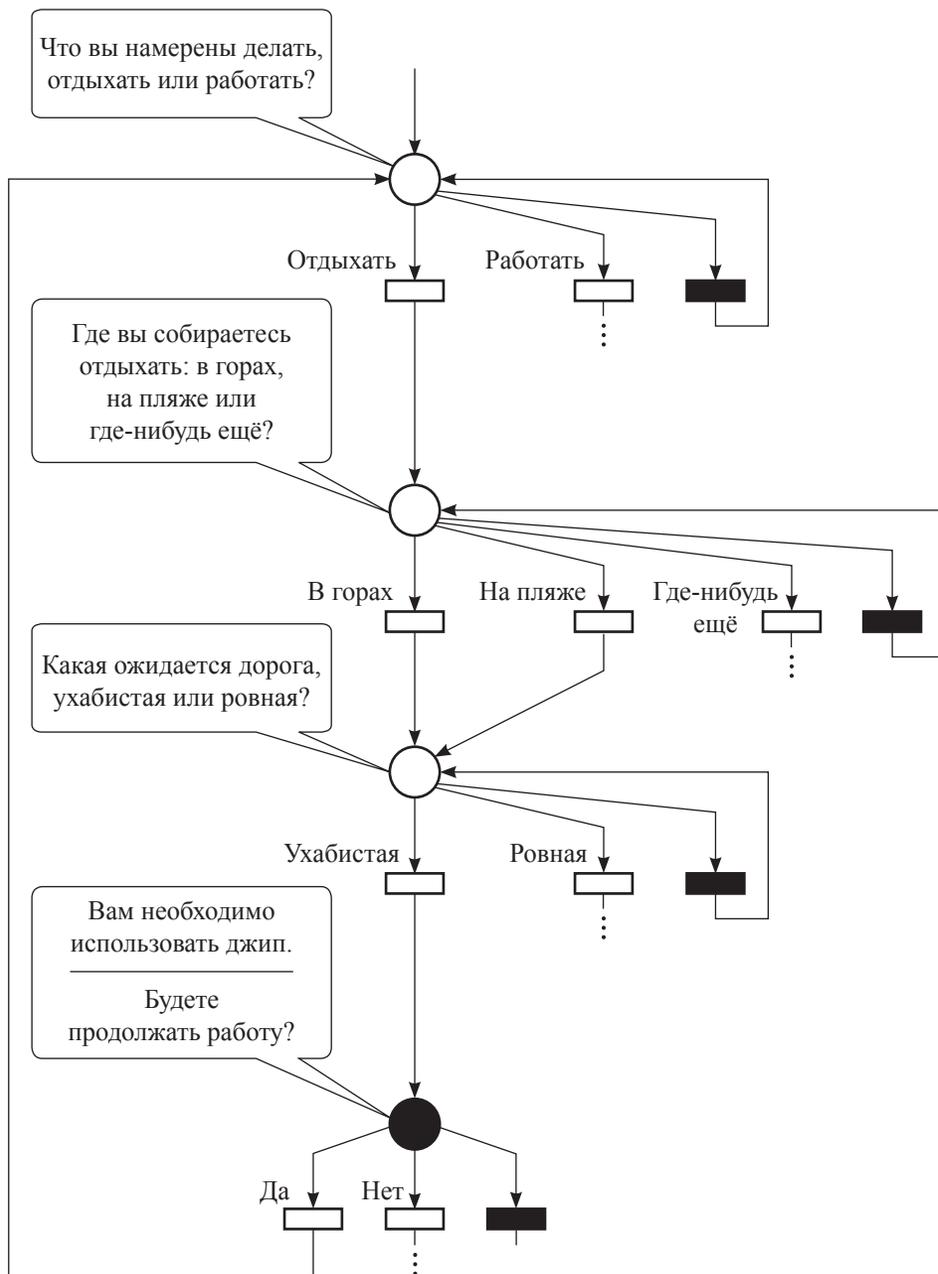


Рис. 5.13. Сеть *DiAM*, эквивалентная деревьям вывода, возможным для базы знаний, описанной фактами (5.6) и правилами (5.7)

Ещё одним заметным позитивным свойством сети *DiAM* является возможность объединения нескольких фактов в одном узле в том случае, если они являются (или могут быть) элементами субъекта одного и того же вопроса. В рассмотренном примере два исходно различных факта: «место отдыха — горы» и «место отдыха — пляж» включены в субъект одного вопроса: «Где вы собираетесь отдыхать: в горах, на пляже или где-нибудь ещё?»

В процессе работы машины вывода ЭС, основанной на правилах, количество данных, хранящихся в рабочей памяти, возрастает по мере развития процесса вывода и, в связи с ограниченностью компьютерных ресурсов, должно быть регламентировано. В диалоговой базе знаний, построенной на основе *DiAM*, необходимо на каждом шаге диалога хранить данные, соответствующие только одной транзакции, вне зависимости от длины дедуктивной цепи.

Есть ещё одно соображение общего характера, которое уместно привести. Если согласиться с тем, что ЭС, основанная на правилах, в процессе вывода не создает факт-заключение, а отыскивает его в имеющейся базе знаний, то тогда машину вывода можно мыслить как некий метод доступа к данным, хранящимся в базе данных. В этом смысле *DiAM* обеспечивает более эффективный доступ. Эффективность проявляется в высокой скорости (нет необходимости строить дерево вывода и разрешать конфликтные ситуации) и низком потреблении ресурса памяти (вне зависимости от длины дедуктивной цепи в рабочей памяти хранится информация только о текущей транзакции).

5.4.2. Вопросно-ответные отношения в контексте логического вывода

Логическая связь между вопросом (*Que*) и ответом (*Ans*) рассматривалась ранее в общем случае в подразделе, посвященном логической структуре диалоговой транзакции (см. подраздел 2.3.3). Модифицируем полученные формализмы, рассматривая эротетический диалоговый процесс с позиций логического вывода [160, 161].

Как было отмечено в 2.3.3, согласно логике вопросов и ответов [96], для получения очередного факта, необходимого для процесса вывода, активный агент предъявляет реактивному агенту информационную посылку, имеющую логическую организацию вопроса, представленного в виде интеррогатива

$$Que \stackrel{\text{def}}{=} ? \text{ Subj, Req,}$$

где *Subj* — субъект вопроса;

Req — требование вопроса;

? — признак интеррогатива.

Как это следует из примеров, рассмотренных в 2.3.4, знания, ассоциированные с субъектом вопроса, чаще всего представимы одной из формул (2.20) или (2.21) (см. подраздел 2.3.4)

$$K_{subj} = \langle \text{объект-вещь} \rangle \{ \text{список объектов-свойств} \},$$

$$K_{subj} = \langle \text{объект-свойство} \rangle \{ \text{список объектов-вещей} \}.$$

Иными словами, в большинстве случаев субъект вопроса представляет собой определенную вещь, связанную с расширенным списком свойств, либо — определенное свойство, связанное с расширенным списком вещей. Запишем приведенные формулы в более формальном виде.

$$Subj = x, \{P_\alpha(x)\}, \alpha = 1, \dots, m, \quad (5.8)$$

$$Subj = P(x), \{x_\alpha\}, \alpha = 1, \dots, m, \quad (5.9)$$

где x — объект-вещь,

$P(x)$ — одноместный предикат, интерпретируемый как x **ОБЛАДАЕТ СВОЙСТВОМ P** ,

$\{P_\alpha(x)\}$ — расширенный список свойств P ,

$\{x_\alpha\}$ — расширенный список вещей x ,

m — количество элементов в расширенном списке.

Например, набор фактов:

место отдыха — горы,

место отдыха — пляж,

место отдыха — Петербург

можно представить в виде (5.9) следующим образом:

$$\left. \begin{aligned} Subj = P(x), \{x_\alpha\}, \alpha = 1, \dots, 3 = \{Fact_1, Fact_2, Fact_3\} \\ Fact_1 = P(x) \mid x = x_1 \\ Fact_2 = P(x) \mid x = x_2 \\ Fact_3 = P(x) \mid x = x_3 \end{aligned} \right\} \quad (5.10)$$

где $P(x)$ — одноместный предикат предикат x **ОБЛАДАЕТ СВОЙСТВОМ БЫТЬ МЕСТОМ ОТДЫХА**,

x_1 — горы,

x_2 — пляж,

x_3 — Петербург.

Поскольку для формирования антецедента в (5.5) на каждом шаге вывода необходимо формировать точно один конъюнкт, то это означает, что из субъекта *Subj* на каждой транзакции необходимо получить точно один факт-ответ, представляющий собой либо один объект типа вещь, либо один объект типа свойство.

$$Ans/Fact = x, \{P(x)\} \quad (5.11)$$

$$Ans/Fact = P(x), \{x\} \quad (5.12)$$

Формулы (5.11) и (5.12) могут быть описаны вербально следующим образом:

- «ответ/факт — это объект-вещь x , обладающий свойством $P(x)$ » либо
- «ответ/факт — это объект-свойство $P(x)$, присущий вещи x ».

Таким образом, в случае прямого вывода, ответ — это один из фактов субъекта, который следует интерпретировать либо как *AFact* (либо промежуточный факт), если процесс вывода не завершён, либо как *CFact*, если процесс вывода завершён.

Требование вопроса *Req* необходимо для формирования ответа, с точки зрения мощности множества ответа. Однако, в том случае, когда эротетический диалог рассматривается в контексте логического вывода, то на каждой транзакции он должен продуцировать точно один ответ-факт. Поэтому при логическом выводе, совмещенном с эротетическим диалогом, смысл требования вопроса — сообщить реактивному агенту, что субъект вопроса содержит альтернативный список объектов.

Резюмируя сказанное, отметим, что транзакцию эротетического диалога можно использовать для получения очередного факта при формировании логического заключения по правилу (5.5). При этом вопрос и релевантный ему факт-ответ можно представить одной из двух пар альтернативных формул:

$$Que = Req, x, \{P_\alpha(x)\}, \alpha = 1, \dots, m, \quad (5.13)$$

$$Ans/Fact = x, \{P(x)\},$$

либо

$$Que = Req, P(x), \{x_\alpha\}, \alpha = 1, \dots, m, \quad (5.14)$$

$$Ans/Fact = P(x), \{x\},$$

где *Req* — требование вопроса, детерминирующее альтернативность элементов расширенных списков $\{P_\alpha(x)\}$ и $\{x_\alpha\}$.

Одной диалоговой транзакции соответствует один узел сети *DiAM*. Цепь, состоящая из последовательно соединённых узлов, моделирует вывод (в смысле правила (5.5)), а вся сеть *DiAM* может рассматриваться как модель базы знаний

ЭС, основанной на правилах, которая хранит процедурные знания, перешедшие в разряд автоматических, в виде сети «всех возможных выводов». Предложим формальное описание *DiAM* как составной части диалоговой базы знаний, используя Петри-модель эротетического диалогового процесса (см. подраздел 3.2).

Пусть *FACT* — комплект фактов-ответов или их идентификаторов, покрывающих некоторую предметную область, которые необходимы и достаточны для формирования диалоговой базы знаний; *QUE* — комплект вопросов или их идентификаторов в виде (5.13) или (5.14).

Пусть совокупность переходов, инцидентных *i*-й позиции сети выводов *DiAM* и соединённых с позицией исходящими ветвями, соответствует множеству фактов

$$F^i \cup NF^i, \quad (5.15)$$

где F^i — множество ожидаемых и распознаваемых альтернативных фактов-ответов из субъекта вопроса для *i*-й транзакции;

NF^i — множество нераспознаваемых фактов-ответов для *i*-й транзакции.

Тогда *i*-й шаг вывода для сети вывода в виде *DiAM* моделируется парой

$$Que_p, (F^i \cup NF^i) \quad (5.16)$$

При графической интерпретации сети выводов *DiAM* вопрос Que_p , как и прежде, соответствует позиции, а множество $F^i \cup NF^i$ — совокупности инцидентных ей переходов. Условием запуска перехода в сети выводов *DiAM* является совпадение ответа, полученного от реактивного агента с одним из фактов-ответов, входящих в (5.15).

Петри-модель *DiAM*, которая рассматривается как база знаний, хранящая процедурные знания, перешедшие в разряд автоматических, в виде сети «всех возможных выводов», зададим четвёркой

$$DiAM = (QUE, FACT, NextQue, NextFact), \quad (5.17)$$

где

$$NextFact : QUE \rightarrow FACT. \quad (5.18)$$

Функция последующих и ожидаемых фактов-ответов, задающая отображение комплекта вопросов (или их идентификаторов) в комплект ожидаемых фактов-ответов.

$$NextQue : FACT \rightarrow QUE. \quad (5.19)$$

Функция последующих вопросов (или их идентификаторов), задающая отображение комплекта ожидаемых фактов в комплект вопросов. Специфика эротематического диалога, заключающаяся в том, что активный агент в течение одной транзакции формирует и передает реактивному агенту только один

вопрос, накладывает на функцию *NextQue* следующее ограничение. Для каждого факта-ответа из комплекта ожидаемых фактов-ответов функция последующих вопросов детерминирует единственный вопрос из комплекта вопросов.

Таким образом, *i*-я позиции сети выводов *DiAM* (*i*-я транзакция диалога) может быть описана следующей парой формул:

$$\text{NextFact}(\text{Que}_i) = (F^i \cup NF^i), \quad (5.20)$$

$$\text{NextQue}(\text{Fact}_j) = \text{Que}_j. \quad (5.21)$$

На рис. 5.14 приведено графическое представление одного шага вывода Петри-модели сети выводов *DiAM*, иллюстрирующее введенные понятия.

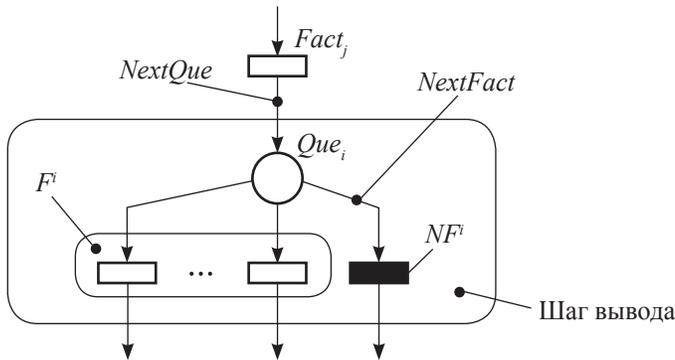


Рис. 5.14. Графическая интерпретация основных понятий Петри-модели сети выводов *DiAM*

Проведенное исследование позволяет предложить концепцию ЭС с распределённой архитектурой.

Организация ЭС, основанной на правилах, подчиняется некоторому стандарту *de facto*, в соответствии с которым архитектура такой системы включает следующие базовые компоненты [162, 163]:

- глобальную базу фактов;
- множество правил продукции, или базу правил;
- систему управления, или интерпретатор правил.

Кроме отмеченных базовых компонентов, в ЭС входит также развитый интерфейс с пользователем. Архитектура продукционных ЭС эволюционирует в направлении на совершенствование каждого из перечисленных компонентов в отдельности. Иными словами, архитектура современных ЭС, основанных на правилах, базируется на концепции сосредоточения базовых функций системы в отдельных блоках. Такая «сосредоточенная» архитектура является универсальной и позволяет независимо совершенствовать отдельные компоненты. Однако

она порождает ряд проблем. Одна из таких проблем связана с разработкой эффективной стратегии интерпретатора правил, которая учитывает потенциальную конфликтность набора правил.

Под распределённой архитектурой ЭС будем понимать такую ее организацию, когда все функции системы распределены между множеством однотипных и структурно подобных элементов. Каждый из элементов содержит фрагмент глобальной базы фактов, базы правил, системы управления и интерфейса с пользователем.

Процесс функционирования распределённой ЭС заключается в пошаговой интерпретации отмеченных однотипных элементов в соответствии с корректировкой цели, вводимой в процессе диалога с пользователем. Таким образом, в распределённой ЭС диалог с пользователем является существенной частью процесса решения задачи. На основании полученных в настоящем разделе результатов, основу организации распределённой ЭС может составлять диалоговая база знаний, представленная *DiAM* и памятью вопросов.

6 РЕШЕНИЕ ПЛОХО ФОРМАЛИЗУЕМЫХ ЗАДАЧ ДИАЛОГОВЫМИ МЕТОДАМИ

Шестой раздел посвящён исследованию классов плохо формализуемых задач с точки зрения применимости диалоговых методов для их решения.

Впервые систематической разработкой методов решения плохо формализуемых задач занялся Ньюэлл [164]. Ньюэлл считал, что к классу плохо формализуемых задач (*ill-formalized* или *ill-structured*) относятся задачи, обладающие одной или несколькими из следующих характеристик:

- задачи не могут быть заданы в числовой форме;
- цели не могут быть выражены в терминах точно определённой целевой функции;
- не существует алгоритмического решения задачи;
- алгоритмическое решение существует, но его нельзя использовать в связи с ограниченностью компьютерных ресурсов (время, память).

По сути, примеры диалоговых процессов и систем, описанные в первом разделе, можно рассматривать с точки зрения решения плохо формализуемых задач. В области прикладных систем информационных технологий можно выделить достаточно много классов систем, направленных на решение соответствующих плохо формализуемых задач. В настоящем разделе внимание сфокусировано на следующих классах таких систем: интеллектуальные персонализированные системы обучения [131, 165, 166]; электронные тексты со встроенным интеллектуальным компонентом [97, 167–170]; интеллектуальные системы автоматизированного проектирования [130, 171–176]; системы диалогового общения с базой данных [177–183]. Решение задач в рамках каждой из перечисленных систем осуществляется в ходе эротетического диалога пользователя и системы. Причём диалог здесь используется не столько для реализации интерфейсных функций, сколько является метаметодом решения задачи.

6.1. Персонализированные системы обучения

Коллинс и Стивен, анализируя поведение опытного человека-тьютора с целью улучшения целенаправленного поведения предложенной ими тьюторской системы *WHY* [26], обнаружили два типа диалоговых эпизодов в процессе обучения, названные ими диагностикой и коррекцией (см. подраздел 1.2.1). Диагностика определяется как процесс, позволяющий проследить связь между множеством поверхностных ошибок и глубоким непониманием материала, а возможность коррекции базируется на способности классифицировать ошибки и выбрать релевантную корректирующую стратегию.

Персонализированный тьютор будет рассматриваться нами как дальнейшее развитие идеи диагностических и корректирующих эпизодов Коллинса и Стивена в направлении учета когнитивных особенностей (когнитивного профиля) субъекта обучения. В дальнейшем будем для краткости вместо словосочетания «субъект обучения» использовать слово «обучаемый» или «учащийся». Таким образом, персонализация системы электронного обучения понимается нами как автоматическая и непрерывная адаптация учебного материала и стратегии обучения к текущим когнитивным характеристикам обучаемого. Персонализированная система обучения — это диалоговый агент, способный поддерживать целенаправленный эротетический диалоговый процесс, направленный на решение задачи передачи знаний обучаемому. Для управления процессом обучения в стиле Сократовского диалога (см. подраздел 2.2) персонализированный тьютор играет роль активного агента и старается вести гармоничный диалоговый процесс или такой диалоговый процесс, в котором оба агента удовлетворены распределением ролей. Однако, в общем случае, персонализированный тьютор может на некоторое время выступать в роли реактивного агента.

В настоящем подразделе исследуется возможность реализации непрерывной диагностики, совмещённой с процессом обучающего диалога в соответствии с теорий, рассмотренной ранее в подразделах 5.3 и 5.4. Конечной целью исследований является ответ на следующий прагматический вопрос: «Каким образом следует строить сценарий диалога, позволяющий осуществлять непрерывную диагностику в процессе передачи знаний?» В качестве предметной области рассматриваются лингводидактические тьюторские системы или системы, предназначенные для персонализированного обучения иностранным языкам.

6.1.1. Обучающие стимулы

Обучающим стимулом будем называть такую информационную посылку активного агента диалога/тьютора, в которой он передает реактивному агенту/обучаемому новую порцию декларативный знаний, ранее неизвестных реактивному

агенту. Обучающий стимул будем рассматривать как микропроблему, которую необходимо решить, или как вопрос, на который необходимо ответить. Не все информационные посылки, передаваемые реактивному агенту, являются обучающими. Например, инструкцию о том, как нужно повторять звучание нового слова в некоторой лингводидактической системе, мы не относим к обучающим стимулам.

В лингводидактической практике в зависимости от выбранного метода обучения отдаётся предпочтение различным обучающим стимулам. Анализ приведенных в литературе конкретных обучающих процедур позволяет классифицировать обучающие стимулы:

1. По типу восприятия: воздействующие преимущественно на зрительный анализатор; воздействующие преимущественно на слуховой анализатор; комбинированные, воздействующие и на зрительный, и на слуховой анализаторы.
2. По степени вовлечения моторных реакций при формировании ответа: моторные реакции необходимы для формирования ответа; моторные реакции не нужны для формирования ответа.
3. По сложности организации.
4. По длительности воздействия. и т. д.

Проведём анализ несколько примеров обучающих стимулов, применяемых в различных классических методах обучения иностранным языкам, с точки зрения их имитации средствами диалогического представления вопроса (см. раздел 4).

В ситуативном методе обучения часто используются следующие обучающие стимулы [184, 185].

Обучающий стимул 1. Преподаватель держит в руках наручные часы и произносит вслух: «This is a watch». Потом он указывает на настенные часы и говорит «That is a clock». Цель стимула — научить учащихся понимать и различать выражения: «This is a...» и «That is a ...». Стимул воздействует одновременно на зрительный и слуховой анализаторы и может быть реализован сценой путем прямой имитации или рефлексии, когда описанная выше сцена воспроизводится в виде анимации на мониторе компьютера с одновременным воспроизведением звука с помощью аудиоканала компьютера.

Обучающий стимул 2. Учащиеся хором повторяют слова и фразы, ранее произнесенные преподавателем. Цель стимула такого рода — обучить учащихся правильному произношению. Стимул воздействует на слуховой анализатор и не требует вовлечения моторных реакций для формирования ответа. Такой стимул крайне сложно реализовать с помощью сцены методом прямой имитации, но можно предложить другой стимул, эквивалентный рассматриваемому, с точки зрения достижения цели. Например, после соответствующих инструкций один из аудиообъектов сцены произносит фразу, которую затем повторяет учащийся.

Контроль произношения осуществляется с помощью микрофона и соответствующей программы.

Обучающий стимул 3. Преподаватель последовательно демонстрирует и называет учащимся небольшие предметы, извлекая их из коробки. Учащиеся повторяют вслух наименования предметов. Цель такого стимула двоякая: обучить учащихся правильному произношению и установить зрительные ассоциации с называемыми предметами. Такой стимул может быть имитирован с помощью динамической сцены следующим образом. На экране появляется коробка с предметами, из которой учащийся последовательным нажатием клавиши «извлекает» предметы. Аудиообъект сцены произносит наименование предмета и просит повторить вслух это наименование. Для проверки правильности произношения используется микрофон и соответствующая программа.

Для аудиолингвистического метода характерны следующие обучающие стимулы [186].

Обучающий стимул 4. Учащемуся предлагается предложение, из которого удалён ряд слов. Учащийся должен восстановить предложение с минимумом искажений исходного предложения. Целью такого стимула является обучение учащегося применению конкретной грамматической конструкции. Стимул воздействует преимущественно на зрительный анализатор и при формировании реакции требует применения моторных навыков в виде письма. Такой стимул может быть реализован при помощи отмеченной выше прямой имитации. Моторная реакция реализуется при помощи клавиатуры.

Обучающий стимул 5. Преподаватель предлагает паре учащихся прочитать вслух диалог. Цель стимула — научить учащихся воспринимать речь и правильно произносить фразы. Стимул воздействует на слуховой анализатор. Ясно, что такой стимул невозможно реализовать в компьютере способом прямой имитации, но можно предложить другой стимул, с помощью которого достигается та же цель обучения. Например, один из аудиообъектов сцены произносит фразу тьютора, затем на экран выводится текст последующей фразы, которую должен произнести учащийся. Контроль произношения учащегося осуществляется с помощью микрофона.

Для метода общего физического отклика характерны следующие обучающие стимулы [187].

Обучающий стимул 6. Преподаватель предлагает учащимся устные команды, которые они должны понять и выполнить. Цель такого стимула — тренировка понимания устной речи. Для его реализации можно использовать следующий подход. Учащийся работает в пределах некоторой сцены, изображенной на экране компьютера, которая, например, представляет собой интерьер помещения. По аудиокомандам тьютора учащийся выполняет действия, используя манипулятор типа «мышь», указывая на предметы, перемещая их и т. д.

Обучающий стимул 7. Преподаватель задает простые вопросы, на которые учащийся должен отвечать жестом, например, указывая на предмет рукой. Цель этого стимула такая же, как и в предыдущем случае — получение навыков понимания устной речи. Реализация такого стимула может быть осуществлена способом, описанным выше. При этом единственным действием, которое выполняет учащийся, является выбор предмета с помощью манипулятора типа «мышь».

В [188] приведен ряд упражнений, из которых выделим следующие обучающие стимулы.

Обучающий стимул 8. Преподаватель предлагает учащимся небольшой фрагмент текста, в котором в неявной форме описано некоторое понятие. Задача учащегося — определить, о каком понятии идёт речь в тексте.

Обучающий стимул 9. Преподаватель предлагает учащемуся текст, из которого удалено несколько слов. Учащийся должен прочесть текст и попробовать заполнить пустые места словами, которые соответствуют тексту по смыслу.

Цель обучающих стимулов 8 и 9 — изучение, понимание и запоминание новых слов с использованием контекста. Этот вид стимулов помогает учащемуся понять, что благодаря контексту можно легче выяснить значение трудных или незнакомых слов. Стимул воздействует преимущественно на зрительный анализатор и требует вовлечения моторных реакций. Поскольку обучающие стимулы 8 и 9 имеют одинаковую цель, то они могут быть трансформированы в однотипные сцены, например, следующим образом. Учащемуся предъявляется текст в течение фиксированного промежутка времени, затем список возможных заголовков. Задача учащегося — выбрать подходящий заголовок.

В [189] рассмотрены примеры обучающих сценариев, основанных на коммуникационном методе (*communicative methodology*) обучения, для которого характерны следующие обучающие стимулы.

Обучающий стимул 10. Преподаватель предлагает учащемуся текст. Учащийся, прочитав текст, должен выразить его содержание различными средствами (с помощью другого текста, рисунка и т. д.). Цель стимула — развитие навыков чтения и понимания текстового материала, а также умения выразить полученные знания. Предложим несколько вариантов обучающих стимулов, эквивалентных стимулу 10 в смысле тождественности целей. Вариант 1. Сцена предъявляет учащемуся текст и список вопросов. Учащийся должен выбрать в тексте предложения, отвечающие на эти вопросы. Вариант 2. Сцена предъявляет учащемуся фрагменты текста в произвольном порядке. Задача учащегося — расставить фрагменты в правильном порядке.

Проведенный анализ типовых обучающих стимулов классических методов обучения иностранным языкам позволяет сделать несколько заключений о возможности их имитации при помощи сцены.

Имитация обучающего стимула средствами сцены всегда предполагает трансформацию реального обучающего стимула. В некоторых случаях проблема трансформации не вызывает затруднений. Стимул трансформируется методом, названным выше непосредственной или прямой рефлексией. Прямая рефлексия чаще всего предполагает разработку динамического/анимационного фрагмента, копирующего реальный обучающий стимул. В ряде случаев (например, случай хорового пения) непосредственная рефлексия неприменима. В этом случае предлагается конструировать обучающие стимулы в соответствии с *принципом эквивалентности целей*. Предполагается, что в каждом обучающем стимуле можно выделить некую цель, заложенную в обучающий стимул преподавателем. Предполагается также, что достижение цели обучающего стимула важнее способа достижения, заложенного в исходный обучающий стимул.

Принцип эквивалентности целей означает, что реальный обучающий стимул, который невозможно имитировать непосредственной рефлексией, заменяется другим, эквивалентным реальному в смысле поставленной цели и легко реализуемым средствами сцены.

6.1.2. Диагностические стимулы

Понимание персонализации процесса обучения как непрерывной адаптации к текущим когнитивным характеристикам обучаемого требует сравнительного анализа методов, применяемых в психодиагностике, поскольку именно эта наука накопила наибольший опыт в области диагностики как общепсихологических, так и когнитивных характеристик человека.

Психодиагностика характеризуется широким спектром методических подходов. Многообразие этих подходов обуславливает существование различных систем классификации психодиагностического эксперимента в зависимости от выделяемых различными авторами значимых для классификации атрибутов [190–193]. Для наших целей таким значимым атрибутом может служить формализуемость психодиагностической методики, которая определяет возможность использования её в диалоговом сценарии. Для конкретизации понятия «формализуемость» целесообразно выделить сравнительно самостоятельные элементы психодиагностического эксперимента, поддающиеся систематизации. К ним относятся, в первую очередь, диагностические стимулы и ответы обучаемого, а также способ анализа полученного набора данных. Соответственно, основанием для классификации психодиагностических методик могут служить различные сочетания форм диагностических стимулов со способами реагирования обучаемых, дополненные характеристиками процедур обработки данных.

В табл. 6.1 представлены формы диагностических стимулов психодиагностических методик.

Стандартизированные стимулы являются одинаковыми для всех обучаемых, в то время как индивидуально ориентированные подбираются в отдельности для каждого конкретного обучаемого.

Таблица 6.1

Виды диагностических стимулов и порядок их предъявления

Виды стимулов				Порядок предъявления стимулов		
Вербальные		Невербальные		Фиксированный	Переменный	
Стандартизированные	Индивидуально ориентированные	Статические	Динамические		Случайный	С обратной связью

Невербальные стимулы могут обращаться к сфере не только зрительного восприятия, а и других органов чувств. Параметры статических невербальных стимулов постоянны во времени, а у динамических — могут изменяться форма, цвет, высота тона звука и т. д. Характеристика диагностических стимулов дополняется описанием порядка предъявления стимулов в процессе психодиагностического эксперимента. Этот порядок может быть фиксированным или переменным. Разновидностью переменного порядка является случайное предъявление стимулов. Другая разновидность предполагает использование обратной связи с обучаемым, когда содержание и форма текущего стимульного материала зависят от реакций обучаемого на предыдущие стимулы.

Способы ответов обучаемого на тестовые стимулы разделяются на закрытые, открытые и динамическое реагирование при помощи органов управления и приведены в табл. 6.2.

При закрытом способе все ответы в совокупности образуют группу заранее известных ответов. Эти ответы могут осуществляться в форме выбора из «меню ответов» с помощью оценивания значения какого-либо признака по заданной шкале путем восстановления частей предложений или фигур, а также посредством переструктурирования данных.

Открытые способы предполагают ответы обучаемого в свободной форме. При этом могут быть регламентированы лишь самые общие требования к форме ответа (вербальная или невербальная, ориентировочные объем и время).

Открытые способы включают в себя ответы в виде дополнения заданного стимула и ответы, в которых полностью допускается свободное конструирование. При динамическом реагировании при помощи органов управления используются перцептивные и моторные способности учащегося.

Таблица 6.2

Виды ответов на диагностические стимулы

Закрытые				Открытые		Динамическое реагирование через органы управления
Выбор	Оценивание признака по заданной шкале	Восстановление частей	Переструктурирование	Дополнение	Свободное конструирование	

Рассмотрим возможные сочетания диагностических стимулов и ответов учащегося и используемые процедуры анализа тестовых наборов данных для каждого сочетания.

Диагностический стимул:
вербальный, стандартизированный.

Ответ:
закрытый, типа «выбор».

Сочетание вербальных стандартизированных стимулов с закрытыми ответами типа «выбор» определяет самый обширный круг психодиагностических методик — тесты-опросники. Диагностические стимулы в этих тестах представлены в виде вопросов или утверждений. Чаще всего используется двух- или трёхальтернативный набор ожидаемых ответов, например, «Верно», «Неверно», «Не могу сказать». Тесты-опросники условно разделяют на опросники-анкеты, биографические опросники и личностные опросники [192–199]. Применение тестов-опросников характеризуется малой степенью вовлеченности психодиагноста в процедуру обследования. Алгоритмы обработки результирующих наборов данных для этих тестов, как правило, представляют собой несложную процедуру подсчета количества совпадений ответов обучаемого с так называемым

диагностическим ключом и последующее приведение полученного результата к нормированному виду.

Диагностический стимул:
невербальный, статический.
Ответ:
закрытый, типа «выбор».

Тесты этого типа часто применяются для исследования пространственного воображения, комбинаторных способностей и способностей соотнесения невербальных стимулов к определенному классу из заданного множества классов. В качестве стимулов могут выступать не только статические, но и динамические объекты сцены. Обработка результатов тестирования в данном случае сводится к оценке количества правильно произведенных выборов. В качестве диагностического параметра нередко используется время решения заданий теста.

Диагностический стимул:
вербальный, стандартизированный;
невербальный, статический.
Ответ:
закрытый, типа «восстановление частей».

Психодиагностические тесты рассматриваемого типа являются модификацией тестов с множественным выбором. Отличительной особенностью этих тестов является то, что обучаемому предлагается самому сформулировать ответ, не прибегая к меню возможных ответов. Таким образом, обучаемый не ограничен какими-либо рамками в своих ответах. В то же время при таком подходе предполагается, что существуют правильные ответы на задания теста, которые могут быть выражены в достаточно определенной форме. Примером методик указанного типа может служить тест Векслера [199] нахождения недостающих деталей, с помощью которого изучаются особенности зрительного восприятия, наблюдательность, способность отличить существенные детали.

Диагностический стимул:
вербальный, стандартизированный;
невербальный, статический.
Ответ:
закрытый, типа «переконструирование».

Диагностические стимулы, предполагающие ответы типа «переконструирование данных», заключаются в составлении комбинаций из заданного набора вербальных или невербальных элементов. Как правило, считается известной «правильная» комбинация элементов, но это не является обязательным условием, так как оценке могут подлежать такие параметры, как, например, оригинальность созданной комбинации. В качестве типичного примера можно привести тест последовательности картинок в шкале измерения интеллекта Векслера. С помощью этого теста исследуется способность к организации фрагментов в логическом поле, пониманию ситуации и предвосхищению событий [193]. В приведенных примерах обработка результатов содержит как количественный, так и качественный компоненты. Но в целом ограниченный набор стимулов, подвергающихся переструктурированию, и, соответственно, ограниченный и известный набор возможных комбинаций даёт основание отнести указанные методики к достаточно четко структурированным и допускающим сравнительно высокую степень формализуемости процедуры обработки.

Диагностический стимул:

*вербальный, стандартизированный;
невербальный.*

Ответ:

*закрытый, типа «оценивание значения
признака по заданной шкале».*

Психодиагностические тесты этого типа связаны с оценкой объектов (словесных утверждений, изобразительного материала, конкретных лиц и т. п.) по выраженности в них качества, заданного шкалой (например, «тёплый — холодный», «сильный — слабый» и т. д.). Примером может служить тест семантического дифференциала [191], используемый для измерения различий в интерпретации понятий. Применяя технику семантического дифференциала для оценки множества объектов одним обучаемым или одного объекта группой обучаемых, на выходе получают числовые таблицы вида объект — признак, которые в дальнейшем могут быть подвергнуты анализу многомерных группировок как объектов, так и признаков. Этот анализ позволяет выявлять особенности психосемантики и отношений к объектам и явлениям. Метод семантического дифференциала дает полезную информацию о «понятийной структуре» обучаемого, направленности и особенностях личностного общения, идентификации себя с окружающими. Интерпретация результатов, полученных методом семантического дифференциала, трудноформализуема, и, как считается, основной акцент в такой интерпретации приходится на психодиагноста, который, помимо своих профессиональных

знаний, должен хорошо представлять особенности алгоритмов выявления структуры многомерных данных.

Диагностический стимул:

*вербальный, индивидуально ориентированный;
невербальный.*

Ответ:

закрытый, типа «оценивание значения признака по заданной шкале».

Примером индивидуально-ориентированного подхода к субъективному шкалированию является техника репертуарных решеток. Она предложена Келли и направлена на изучение индивидуально-личностных конструктов, опосредующих восприятие и самовосприятие при анализе личностного смысла понятий [193]. Несмотря на внешнее сходство, техника репертуарных решеток радикально отличается от метода семантического дифференциала. Здесь используются не заданные извне признаки, а выявляются собственные, индивидуальные конструкты, создаваемые непосредственно в ходе обследования. Поэтому техника репертуарных решеток является гораздо более гибким и тонким инструментом. Между тем эти качества обуславливают значительную вовлеченность эксперта в процесс психодиагностики, и хотя для эффективного использования техники репертуарных решеток совершенно необходима компьютерная поддержка методики, в самой процедуре обследования и интерпретации результатов акцент ещё более смещается в сторону использования трудно формализуемых знаний и опыта эксперта — психодиагноста.

Диагностический стимул:

вербальный, стандартизированный.

Ответ:

открытый, типа «дополнение».

Примером психодиагностических тестов данного типа может служить методика завершения предложений. Обучаемому предлагается серия незаконченных предложений, состоящих из одного или нескольких слов, которые он должен завершить по своему усмотрению. Предложения в данной методике формулируются таким образом, чтобы стимулировать обучаемого на ответы, относящиеся к изучаемым свойствам личности. Другим примером является методика завершения историй. Стандартизация рассматриваемых методик достигается путем

отнесения произвольных ответов обучаемого к некоторому опорному множеству психологических категорий.

Диагностический стимул:
невербальный, статический.
Ответ:
открытый, типа «дополнение».

Примером такого теста является рисуночный тест Вартегга [200]. Обучаемому необходимо в имеющемся пространстве выполнить рисунки с учетом уже изображенных знаков. В процессе тестирования фиксируется последовательность выполнения рисунков, а после завершения задания обучаемому предлагается ответить на ряд вопросов (какой рисунок более удался, каково содержание рисунков и пр.) При интерпретации полученных результатов исходят из определенных свойств, приписываемых графическим символам. Так, например, точке приписывают органичность и неопределенность; квадрату — угловатость, темноту, тяжесть и т. д.

Диагностический стимул:
*вербальный, стандартизованный;
невербальный.*
Ответ:
открытый, типа «свободное конструирование».

Для тестов, предполагающих ответы в форме свободного конструирования, стимулы могут быть самыми разнообразными — как вербальными, так и невербальными. То же самое можно сказать и о виде ответов обучаемого — это могут быть рисунки на созвучную тему, рассказы, интерпретации изображений и т. д. В этом случае основная доля приходится на проективные тесты. Диагностика с помощью проективных тестов осуществляется на основе анализа взаимодействия обучаемого с внешне нейтральным материалом, становящимся, в силу недостаточной структурированности, «стимулом», порождающим фантазию и воображение обучаемого [193]. К проективным тестам относятся также и рассмотренные выше методики с ответами в виде дополнений стимулов. Но целесообразно выделить методики со свободным конструированием ответов в отдельную группу, так как они имеют еще менее формализуемую структуру. В качестве примеров приведем два популярных теста — тест Роршаха [201], в котором учащийся должен придать смысл симметричным аморфным черно-белым и цветным изображениям, и тест тематической аперцепции, основанный

на толковании обучаемым сюжетов специально подобранных картинок [202]. В проективных методиках количественные диагностические оценки могут быть получены на основании измерения объема ответа обучаемого, подсчета частоты обращения к отдельным «темам» и т. п. На практике использование проективных методик часто опирается на интуицию и теоретическую подготовку психодиагноста.

Диагностический стимул:

невербальный, динамический.

Ответ:

динамическое реагирование через органы управления.

Рассматриваемое сочетание стимулов и ответов соответствует классу психодиагностических методик, которые обычно называют аппаратными тестами. Эти тесты используются в исследованиях параметров времени реакции, типологических особенностей высшей нервной деятельности и пр. Наличие обратной связи между ответами учащегося и стимулами свойственно большому количеству критериально-ориентированных аппаратных тестов, в которых моделируются условия какой-либо критериальной деятельности. До сравнительно недавнего времени реализация этих методик требовала специального и часто дорогостоящего оборудования. Развитие компьютерной техники открыло новые перспективы развития аппаратных методик. На экране монитора могут моделироваться разнообразные виды деятельности, имитироваться объекты слежения, управления и т. д. Параллельно с помощью специальных датчиков может производиться съем и ввод в компьютер психофизиологической информации. Отдельно можно выделить также подкласс аппаратных тестов, который в настоящее время стал активно развиваться и в котором моделирование опосредуется компьютерными играми.

Диагностический стимул:

вербальный, индивидуально ориентированный;

Ответ:

открытый.

Методики этого класса можно определить как сугубо диалоговые, в которых предполагается непосредственный контакт психодиагноста с обучаемым и учитываются специфические особенности конкретной диагностической задачи.

Последовательность диалоговых транзакций, построенных на основе диагностических стимулов, порождает набор тестовых данных, подлежащих

заключительной обработке. Рассмотрим методы обработки тестового набора данных, применяемые в психодиагностике.

В большинстве случаев для изучения психологических феноменов анализируются взаимосвязи результирующих показателей психодиагностических методик (разнообразных диагностических шкал, психологических факторов и т.п.). Традиционная методология конструирования психодиагностических шкал достаточно детально разработана. На практике широко используется ряд всевозможных коэффициентов надёжности и валидности как отдельных элементов тестов, так и тестовых методик в целом, а психометрические дискуссии ведутся по поводу нюансов выбора, например, той или иной меры связи между переменными, того или иного критерия внутренней согласованности теста и т.п. В то же время традиционный психодиагностический эксперимент застыл на одной элементарной схеме. Формальные операции обработки результатов психодиагностического тестирования, как правило, заключаются в анализе совпадений ответов обучаемого с так называемым «ключом», который представляет собой заранее известный список ответов, «работающих» на проявление диагностируемой психологической характеристики. Очень часто просто подсчитывается число таких совпадений, реже они суммируются с определёнными весами. Подобные процедуры, в которых ответы обучаемых на диагностические стимулы теста играют роль исходных дихотомических признаков, относятся к известному в прикладной статистике классу линейных диагностических правил. Линейные диагностические правила в настоящее время превалируют в психодиагностике.

Не менее серьёзные вопросы возникают при рассмотрении традиционной практики конструирования психодиагностических тестов. Современный психометрист в процессе создания теста обязательно анализирует два вида статистических связей — связь пунктов теста с критерием внешней валидности или с суммарными результатами теста и попарные связи между пунктами теста. При этом часто считается, что максимальная валидность теста достигается за счёт отбора таких пунктов, которые, обладая значимой корреляцией с внешним критерием, минимально коррелируют между собой. Однако можно привести довольно много аргументов, доказывающих неполноту такого анализа. Ограничимся только одним, но достаточно весомым. Он заключается в том, что пункты теста, каждый из которых по отдельности не коррелирует с внешним диагностическим критерием, способны в совокупности обеспечить полное и чёткое разделение диагностируемых групп обучаемых. Используя традиционный психометрический подход, эти пункты будут совершенно напрасно забракованы как неинформативные.

В последнее время всё больше исследователей начинают видеть реальную альтернативу сложившимся психодиагностическим подходам в применении методов теории искусственного интеллекта и, в частности, теории распознавания образов. Разработка процедур принятия решений (диагностики и классификации) является одной из прерогатив этой теории.

В настоящее время известно большое количество математических моделей, пригодных для решения задач распознавания образов. Большинство этих моделей опирается на геометрическое изображение и истолкование характеристик распознаваемых объектов в пространстве признаков [203–209]. В то же время в психодиагностике, которая сильно повлияла на развитие теории распознавания образов, методы этой теории находят чрезвычайно ограниченное применение. На ограниченное применение алгоритмов распознавания образов в психодиагностике оказали влияние многие причины. Но, конечно, главную роль сыграла высокая размерность пространства признаков, которыми должен оперировать экспериментатор при проведении психодиагностических исследований, так как увеличению размерности сопутствует, как правило, экспоненциальный рост сложности указанных алгоритмов. Однако быстрый темп развития ресурсов современных компьютеров даёт основание рассматривать использование методов теории распознавания образов в качестве ближайшей реальной перспективы психодиагностики.

6.1.3. Диалоговый сценарий, совмещающий диагностику и обучение

Диагностические транзакции, необходимые для адаптации тьютора к когнитивным характеристикам учащегося, включают набор диагностических стимулов, «связанных» стратегией диагностики. Стратегия диагностики в общем случае может быть представлена следующими двумя фазами:

- формирование достаточного набора ответов обучаемого;
- анализ ответов и специфицирование релевантной коррекции.

Современная психодиагностическая практика использует относительно простые методы анализа ответов обучаемого, часто сводящиеся к сравнению фактических ответов с заранее подготовленным диагностическим ключом. В основе такого метода лежат умозаключения, заранее проведенные экспертом-психодиагностом. Поэтому актуальными являются результаты исследований, полученные в подразделе 5.4 и посвященные возможности применения методов теории искусственного интеллекта на второй фазе стратегии диагностики.

Напомним, что главной гипотезой, лежащей в основе этих исследований, является предположение о том, что эротетический диалоговый процесс обладает свойствами процесса дедуктивных умозаключений и может рассматриваться как средство непрерывной диагностики учащегося.

Проиллюстрируем полученный в подразделе 5.4 результат примером трансформации экспертной системы, основанной на правилах, в сценарий диалогового процесса. Пусть, например, в рамках некоторой персонализированной лингводидактической системы осуществляется обучение словам и одновременно диагностируется знание пяти слов методом, который описывается при помощи набора правил, приведенного на рис. 6.1

- ЕСЛИ** из пяти слов хотя бы на три получены правильные ответы,
ТО слова изучены с оценкой «удовлетворительно»;
- ЕСЛИ** из пяти слов хотя бы на три получены неправильные ответы,
ТО необходимо продолжить изучение слов;
- ЕСЛИ** из пяти слов хотя бы на четыре получены правильные ответы,
ТО слова изучены с оценкой «хорошо»;
- ЕСЛИ** на все слова получены правильные ответы,
ТО слова изучены с оценкой «отлично»;
- ЕСЛИ** на первые три слова получены правильные ответы,
ТО предварительная оценка «удовлетворительно» и необходимо продолжать тестирование;
- ЕСЛИ** из первых четырех слов хотя бы на три получены правильные ответы,
ТО предварительная оценка «удовлетворительно» и необходимо продолжать тестирование;
- ЕСЛИ** на первые четыре слова получены правильные ответы,
ТО предварительная оценка «хорошо» и необходимо продолжать тестирование.

Рис. 6.1. Продукционные правила, используемые для оценки знаний обучаемого в рассматриваемом примере

Возможные цепи логического вывода на приведенном множестве правил, необходимые для получения заключения об оценке знаний обучаемого, могут быть объединены в соответствующий сценарий эротематического диалога и представлены в виде сети Петри так, как это показано на диаграмме рис. 6.2. Для упрощения диаграммы на ней отмечены только ожидаемые и распознаваемые ответы, а также выделены четыре фрагмента сценария, обозначенные блоками 1, 2, 3 и 4.

На рис. 6.3–6.6 приведены сценарии диалога, соответствующие блокам 1, 2, 3 и 4. На рис. 6.2–6.6 приняты следующие обозначения:

$Stmi$ — обучающий стимул, представляющий i -е слово;

Pi, Ni — правильный и неправильный ответы для i -го слова соответственно;

$CFactI$ — заключение: «слова изучены с оценкой «отлично»;

CFact2– заключение: «слова изучены с оценкой «хорошо»»;
CFact3– заключение: «слова изучены с оценкой «удовлетворительно»»;
CFact4– заключение: «необходимо продолжать изучение слов».

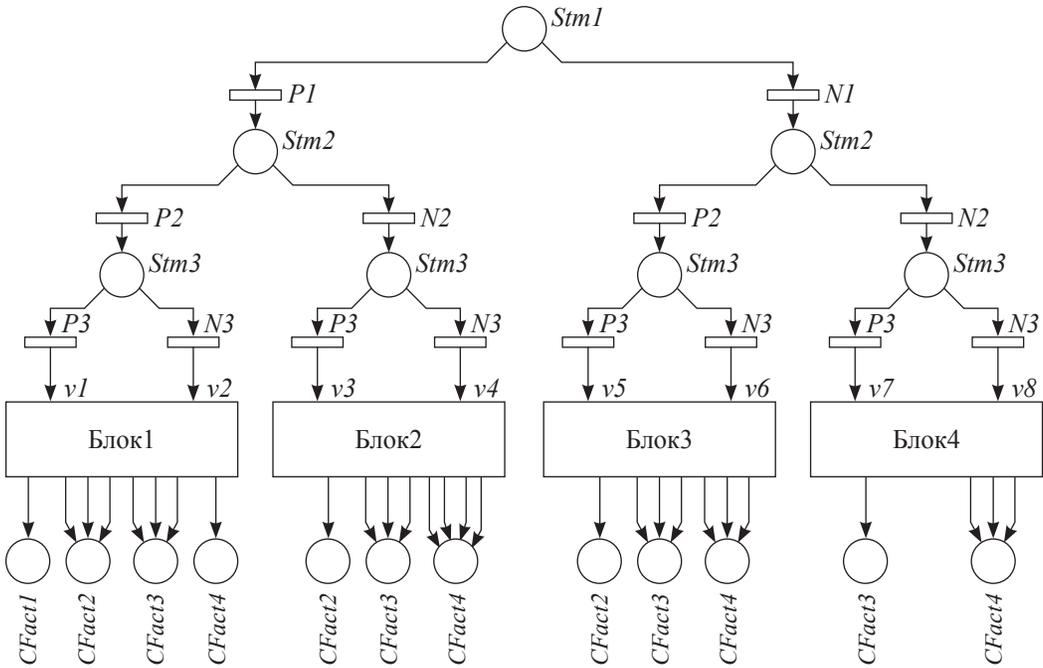


Рис. 6.2. Петри-модель сценария, эквивалентного всем деревьям вывода для примера, описанного в тексте

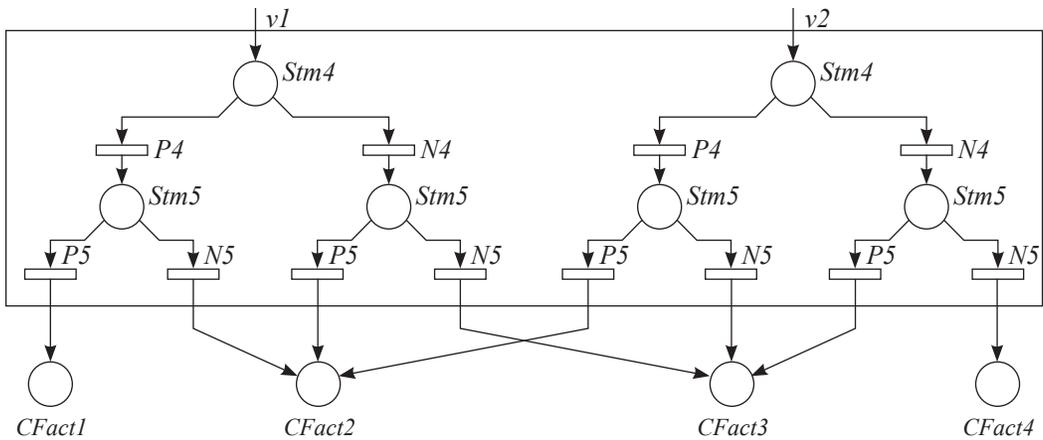


Рис. 6.3. Петри-модель сценария, соответствующего блоку 1 на рис. 6.2

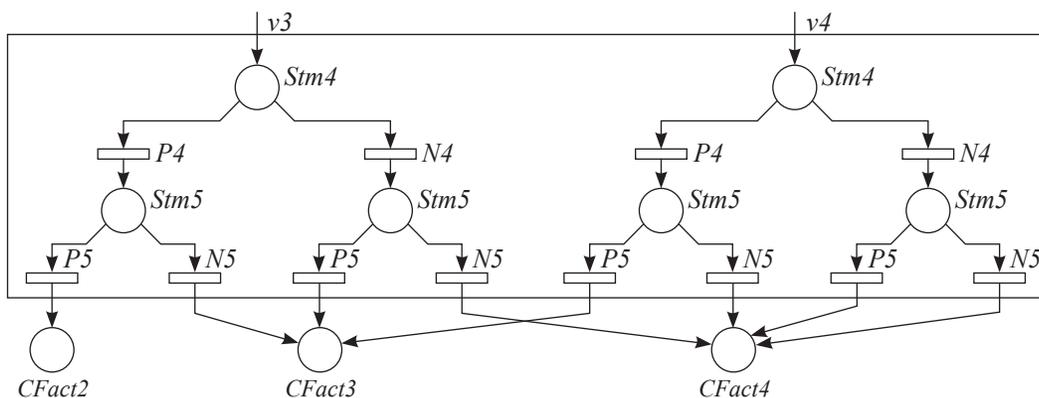


Рис. 6.4. Петри-модель сценария, соответствующего блоку 2 на рис. 6.2

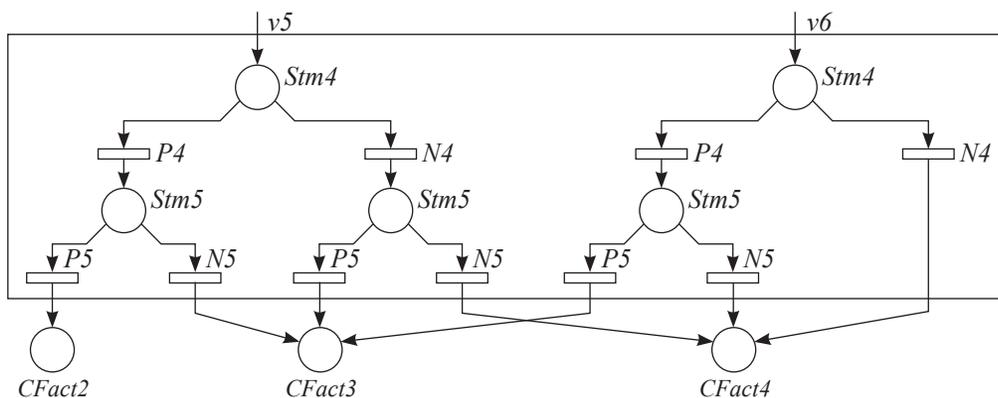


Рис. 6.5. Петри-модель сценария, соответствующего блоку 3 на рис. 6.2

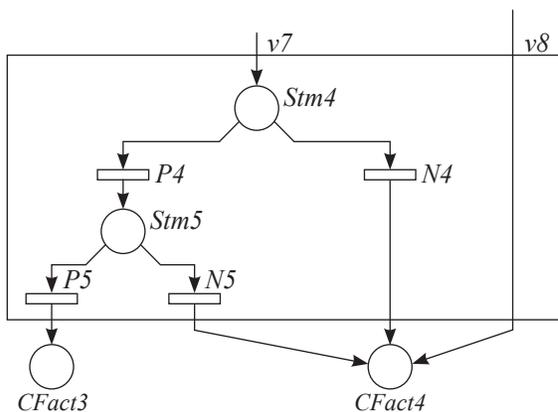


Рис. 6.6. Петри-модель сценария, соответствующего блоку 4 на рис. 6.2

Петри-модель сценария эротетического диалога, изображенная на рис. 6.2, иллюстрирует прямой вывод, который завершается достижением узла, соответствующего *CFact*. Таким образом, для целевой позиции нарушается смысловая интерпретация позиций и переходов (напомним, что, согласно договоренности, переход соответствует факту, а позиция — стимулу). Целевая позиция (в случае прямого вывода) содержит факт-заключение *CFact* и очередной вопрос. Наличие вопроса в целевой позиции необходимо для продолжения диалога. Модель предполагает, что после получения заключения сценарий возвращается в начальное состояние.

6.2. Электронный текст со встроенным интеллектуальным компонентом

Учащиеся сталкиваются с трудностями в процессе формирования семантики изучаемого текстового материала. Эти трудности возникают в тех случаях, когда учащийся работает с текстом, сложность которого находится либо за пределами его читательских навыков, либо за пределами его концептуального базиса. Учащиеся могут пропускать или не понимать ключевые понятия текста или быть незнакомыми с его техническим словарём. Отмеченные трудности усугубляются, если учащийся работает с текстом, написанным не на его родном языке. С особыми трудностями сталкиваются учащиеся-инвалиды, страдающие различной формой дислексии, недостатками слуха или зрения. Главная причина отмеченных трудностей — ограниченная или неадекватная поддержка, необходимая для понимания, манипулирования, изучения и применения знаний, полученных из текстового материала, к задачам реального мира.

В настоящем подразделе описываются основные идеи, лежащие в основе компьютерных систем (главным образом, системы *SmartText* [97]), предназначенных для помощи учащимся в формировании семантики изучаемого текстового материала. Системы типа *SmartText* являются объединением двух направлений исследований в области информационных технологий и искусственного интеллекта: концепции «текста, расширенного ресурсами» и теории эротетического диалога.

6.2.1. Текст, расширенный ресурсами

Текст, расширенный ресурсами (в англоязычной литературе — Supported Text Document, или STD) представляет собой специальным образом адаптированную форму гипертекста, разработанную с целью обеспечения учащихся

помощью в процессе формирования семантики изучаемого текстового или мультимедийного материала. В литературе описано несколько практических систем, представляющих собой реализацию текста, расширенного ресурсами, как для локального использования, так и в качестве Web-приложения. Примером может служить проект *Perseus*, описанный в [210] и предназначенный для изучения истории и культуры древней Греции.

Текст, расширенный ресурсами, состоит из трёх основных компонентов: исходного текста (*source text*), набора ключей (*set of keys*) и набора ресурсов (*set of resources*).

Исходный текст — это тот текстовый документ, с которым работает учащийся и который он пытается понять и ассоциировать с ранее сформированными когнитивными схемами.

Ключи представляют собой те части исходного текста, которые могут вызывать у учащегося трудности в восприятии и понимании в процессе решения задачи формирования семантики. Ключи отмечены в тексте ключевыми маркерами, которые, будучи активированы, обеспечивают доступ учащихся к пакетам ресурсов. Подготовленные учащиеся могут работать с исходным текстом, практически не пользуясь ключевыми маркерами. Для таких учащихся ресурсы «свёрнуты» и невидимы и не нарушают естественный ход восприятия материала. Однако в том случае, когда учащийся, учитель или встроенный интеллектуальный компонент обнаруживают необходимость в ресурсе, ключевой маркер активизируется и обеспечивает доступ к специфическому ресурсу.

Ключи делятся на две категории. Синтаксические ключи представляют собой физические элементы исходного текста: технический или контекстный словарь, сложные или запутанные предложения или параграфы, символы, заголовки, ссылки и т.п. Поскольку эти ключи являются частью исходного текста, они, как правило, играют роль ключевых маркеров.

Семантические ключи не могут быть промаркированы непосредственно при помощи элементов исходного текста, поскольку они не являются физическими элементами текста. Они относятся к когнитивным процессам, связанным с пониманием текста и, возможно, нуждающимся в дополнительных знаниях.

В [211] приведен отчёт исследований, посвящённых изучению понимания учащимися исторических текстов, в котором отмечается, что спецификой понимания таких текстов является умение связывать исторические события с соответствующими причинами или умение конструировать соответствующие причинно-временные отношения между событиями, описанными в тексте. Для выполнения отмеченной работы учащемуся, возможно, понадобится обращаться к некоторому количеству ресурсов: ресурсам, необходимым для создания письменных заметок, вербально отражающих когнитивные схемы; ресурсам, необходимым для коммуникации с коллегами; к понятийным обзорам, посвящённым изучаемому материалу; списку исторических событий и т.п. Учащийся

может получить необходимые ресурсы либо непосредственно от учителя, либо из текста, расширенного ресурсами.

В [97] предлагается следующая категоризация ресурсов:

1. Трансляционные ресурсы (translational resources) обеспечивают альтернативными формами или определениями отдельные слова и фразы или переводом на родной язык учащегося.
2. Иллюстративные ресурсы (illustrative resources), демонстрирующие примеры, графическую или видео- интерпретацию.
3. Резюмирующие ресурсы (summarizing resources) с обзорами структуры, содержания или главных отличий (или идей) текстового материала.
4. Ссылочные ресурсы (referential resources), обеспечивающие доступ к информации, на которую имеются ссылки.
5. Учебные ресурсы (instructional resources), содержащие инструкции по взаимодействию учащегося с текстом.
6. Сопутствующие ресурсы (collateral resources), пополняющие и расширяющие текстовый материал.
7. Заметочные ресурсы (notational resources) для ведения заметок, аннотаций, установки закладок или подчёркивания текстовых элементов.
8. Кооперативные ресурсы (collaborative resources), обеспечивающие возможность кооперации с другими учащимися.
9. Ресурсы общего назначения (general purpose resources), такие, например, как энциклопедии.

Более подробное описание текста, расширенного ресурсами, можно найти в [212].

6.2.2. Задачи встроенного интеллектуального компонента

В [213, 214] описана система *Electrotext*, представляющая собой текст, расширенный ресурсами, созданный при помощи пакета программ *HyperCard* и ориентированный на работу с локальным компьютером. Система *Electrotext* предназначена для учащихся средних школ, страдающих дислексией. В качестве исходного текста используются короткие и простые рассказы. Ресурсы, необходимость обращения к которым учащиеся должны принимать самостоятельно, представляют собой определения и толкования трудных слов, три типа вспомогательных вопросов, а также картинки, иллюстрирующие некоторые события и понятия.

В [212] описана система *LITERACY-HI*, а также результаты исследования поведения учащихся с недостатками слуха при работе с этим текстом, расширенным ресурсами. Используя гипертекстовую версию отдельных глав из своего школьного учебника и портативные персональные компьютеры, учащиеся

получали доступ к следующим ресурсам: трансляция трудных слов и фраз в текстовое толкование, синтезированную речь или видеоклип; иллюстрации ключевых понятий и процессов при помощи анимации; резюме содержания главы с использованием графики; вспомогательные вопросы, помогающие понять содержание текста.

В [97] описана система *Web de Anza*, представляющая собой web-ориентированный текст, расширенный ресурсами, предназначенный для развития критического мышления учащихся при работе с историческими текстами. *Web de Anza* создан на основе документального материала, посвящённого экспедиции Жана Батиста де Анза (Juan Bautista de Anza), который в период с 1774 по 1776 годы исследовал территорию Калифорнии и основал город Сан-Франциско. Рис. 6.7 иллюстрирует компоненты системы *Web de Anza*.

Опыт применения текстов, расширенных ресурсами для различных групп учащихся, позволяет сформулировать три главные причины, обуславливающие необходимость встраивания в его структуру интеллектуального компонента.

Первая причина. Несмотря на то, что учащиеся легко осваивают технику работы с текстом, расширенным ресурсами, они часто в процессе работы забывают, что текст, расширенный ресурсами — это не просто текст в электронной форме, представленный на экране компьютера. В экстремальных случаях учащиеся либо забывают, что текст, расширенный ресурсами — это интерактивная система, и игнорируют ресурсы, либо фокусируют внимание исключительно на ресурсах и игнорируют исходный текст. Подобные проблемы решаются вмешательством учителя, однако они могут решаться и вмешательством интеллектуального компонента, встроенного в текст, расширенный ресурсами.

Вторая причина. Различные учащиеся, с точки зрения их начальных знаний и когнитивных способностей, требуют различных пакетов ресурсов. Пакеты ресурсов могут быть персонализированы путём использования файлов предпочтений, формируемых либо самим учащимся, либо при помощи учителя. Однако, как правило, учащийся не в состоянии ясно сформулировать свои когнитивные предпочтения. Поэтому было бы полезно осуществлять автоматическую диагностику и формирование файла предпочтений при помощи интеллектуального компонента.

Третья причина. В процессе работы с текстом, расширенным ресурсами, учащимся может понадобиться необходимость пополнения некоторых знаний, нехватка которых существенна и мешает пониманию текста. В [97] отмечается, что концептуальные карты (concept maps) являются эффективным средством представления причинно-временных отношений, необходимых для формирования семантики исторических текстов. Учащиеся, незнакомые с отмеченной нотацией, могут получить по запросу короткий обучающий курс по применению концептуальных карт. Целесообразно возложить эти функции на встроенный интеллектуальный компонент.

Anza Web-Based Study Environment

The image displays a complex web-based study environment for the Anza expedition. It consists of several overlapping browser windows:

- Atlas - Microsoft Internet Explorer:** Shows a map of the Anza region with navigation options like Contents, Archive, Atlas, Biographies, Collateral Materials, Image Gallery, Reference Materials, Study Guides, Timelines, and Bibliography.
- Spanish Expedition Routes:** Lists routes such as Anza 1774-75 Expedition Route, Anza 1775-76 Expedition Route, and Portola Expedition Route.
- Image Gallery - Microsoft Internet Explorer:** Features a grid of images with categories like People, Places, and Things. Specific items include Leaving Tubac, Muskets, Expedition Leaves Tubac, San Gabriel Mission, and Tule Raft.
- Moraga Diary 1775-76 - Microsoft Internet Explorer:** Displays a diary entry for Monday, February 12, 1776, describing the start of the mission to San Gabriel.
- Font Short Diary 1775 - Microsoft Internet Explorer:** Shows a diary entry for Monday, February 12, 1776, in Spanish, detailing the departure from Santa Ana.
- Anza Diary 1775-76 p2 - Microsoft Internet Explorer:** Contains a detailed English diary entry for Monday, February 12, 1776, mentioning the loss of a soldier and the use of muleteers.
- Resource Index - Microsoft Internet Explorer:** Provides a structured overview of the expedition's progress, including dates, locations, and key events.
- San Gabriel Mission - Microsoft Internet Explorer:** Offers historical information about the mission, its establishment in 1771, and its significance.
- Resource Guides:** Lists various resources such as Missions, Diaries, and Pictures.

Arrows indicate the flow of information and cross-referencing between these different components of the study environment.

Рис. 6.7. Компоненты текста, расширенного ресурсами системы *Web de Anza*

Таким образом, можно сформулировать две главные задачи, стоящие перед интеллектуальным компонентом, встраиваемым в текст, расширенный ресурсами:

- компенсация недостатка когнитивных навыков учащихся при оперировании и навигации обширными ресурсами текста;
- адаптация компонентов текста, расширенного ресурсами, к персональным когнитивным характеристикам учащихся.

Обе задачи направлены на углубление персонализации текста, расширенного ресурсами, относительно текущих когнитивных способностей отдельного учащегося. Это означает, что интеллектуальный компонент, встроенный в текст, расширенный ресурсами, может помочь решить проблему разработки персонализированной обучающей среды (см. подраздел 1.2.2) с целью формирования семантики из текстов общего назначения для учащегося с заранее неизвестными когнитивными способностями.

Продолжим обсуждение проблемы персонализации текста, расширенного ресурсами, путём его адаптации к текущим когнитивным способностям учащегося, с точки зрения технологической парадигмы, которую исследователи из Массачусетского Технологического Института (MIT Media Lab) назвали «вещи, которые думают» (things that think). В [215] описано несколько интересных примеров «вещей, которые думают»: обувь, кофеварка, микроволновая печь и другие, которые отражают точку зрения авторов на «встроенный интеллект».

Одним из характерных примеров является поведение «думающей» кофеварки. Если вы приближаетесь к такой кофеварке с пустой чашкой, то она может задать вам вопрос типа: «Ещё чашку кофе, приятель?» Простого ответа «да» или «нет» — достаточно. В ответ машина может заметить: «Между прочим, это твоя третья чашка кофе сегодня, а чашку ты не мыл уже несколько дней...»

Описанное поведение может быть смоделировано при помощи диалоговой системы, агентами которой являются «думающая кофейная машина» и её пользователь. С целью симуляции разнообразного поведения, «думающая кофейная машина» не должна повторять одну и ту же фразу при повторяющихся действиях пользователя. Её поведение должно изменяться, или быть адаптивным. Анализ примеров поведения «думающих вещей», описанных в [215], позволяет перечислить отличительные характеристики таких диалоговых систем:

- «думающие вещи» находятся в диалоге со своими пользователями;
- деятельность «думающих вещей» может быть определена как диалоговое поведение;
- диалоговое поведение «думающих вещей» является целенаправленным и адаптивным.

Возвращаясь к проблеме текста, расширенного ресурсами, в качестве первого приближения, определим его как подкласс систем встроенного интеллекта, который отличается от простого текста, расширенного ресурсами, следующими отличительными характеристиками:

- интеллектуальный текст, расширенный ресурсами, находится в постоянном диалоговом общении с учащимся;
- интеллектуальный текст, расширенный ресурсами, действует автономно или полуавтономно, и его деятельность может быть квалифицирована как поведение;
- поведение интеллектуального текста, расширенного ресурсами, целенаправленно и адаптивно.

Интеллектуальный текст, расширенный ресурсами, реализует своё поведение путем генерации последовательности умных страниц (SmartPages), которые он предоставляет учащемуся. Эта последовательность отражает как метод навигации, так и когнитивные навыки учащегося и соответствует его когнитивным способностям. Содержание и атрибуты умной страницы отражают персонализированный набор компонентов текста, расширенного ресурсами, и соответствуют когнитивным способностям учащегося. Интеллектуальный текст, расширенный ресурсами, не может хранить набор заранее подготовленных умных страниц, поскольку они зависят от учащегося и должны быть персонализированы. Идея заключается в том, что умная страница динамически генерируется непосредственно перед выводом. Содержание умной страницы, ключевые маркеры, пакеты ресурсов и т. д. создаются в процессе сборки в соответствии с зависимым от учащегося (персонализированным) описанием умной страницы. Описание умной страницы отражает когнитивные характеристики обучаемого и генерируется встроенным диалоговым компонентом текста, расширенного ресурсами, в соответствии с информацией, накопленной в ходе диалога с учащимся.

6.2.3. Организация встроенного интеллектуального компонента

Одной из важных характеристик текста, расширенного ресурсами, со встроенным интеллектуальным компонентом является его перманентная интерактивность. Хайес [216] различает два типа интерактивных систем: диалоговые (или разговорные) системы и инструментальные системы. В диалоговых или разговорных системах пользователь взаимодействует с системой с целью совместного решения проблемы. Инструментальные системы полностью контролируются пользователем, с их помощью пользователь формирует последовательность команд без вмешательства системы.

Существует ещё одна важная отличительная характеристика интерактивных систем — автономность. Автономные интерактивные системы, проинициализированные пользователем, функционируют полностью автономно или используют пользователя в качестве сенсора первичной информации. Примером автономных интерактивных систем могут служить большинство производственных экспертных систем диагностического типа. Диалоговые системы — не

неавтономны. Процедурные и декларативные знания, необходимые для решения задачи, распределены между системой и её пользователем.

Важной функцией интеллектуального текста, расширенного ресурсами, является уменьшение информационной перегрузки учащегося путём пошагового и селективного представления релевантной информации. Легко представить обширный текст, расширенный ресурсами, включающий тысячи страниц и десятки тысяч ключей, связанных с ресурсами, в качестве примера системы, в которой даже опытный учащийся будет в затруднении осуществлять навигацию самостоятельно. Способность эротетической диалоговой системы отображать персонализированную информацию предотвращает информационную перегрузку учащегося. Из сказанного следует, что эротетические диалоговые системы являются наиболее подходящим типом систем для реализации текста, расширенного ресурсами со встроенным интеллектуальным компонентом.

Диалог как метод решения проблем предполагает существование двух целенаправленных и взаимодействующих агентов, которые формируют общую когнитивную систему. Для демонстрации целенаправленного поведения текст, расширенный ресурсами, со встроенным интеллектуальным компонентом должен обладать такими свойствами, как пошаговая передача/восприятие знаний, аккумуляция фактических знаний об учащихся, способность осуществлять логический вывод относительно когнитивных способностей учащегося и т. д.

Рассмотрим элементы теории текста, расширенного ресурсами, со встроенным интеллектуальным компонентом и организацию интеллектуального компонента на уровне, инвариантном к предметной области диалога.

Мы рассматриваем диалоговое поведение как дискретное или пошаговое. Шаг диалогового поведения соответствует диалоговой транзакции, а все сценарии диалогового поведения текста, расширенного ресурсами, могут быть синтезированы с использованием некоторого конечного набора транзакций сценария. При выполнении транзакции завершается элементарный цикл обмена знаниями между учащимся и текстом, расширенным ресурсами. Учащийся во время выполнения транзакции получает умную страницу, изучает её, принимает решение о последующих действиях и формирует ответ.

Естественное распределение ролей диалоговых агентов, в случае текста, расширенного ресурсами со встроенным интеллектуальным компонентом, — это реализация учащимся роли реактивного агента диалога и реализация текстом, расширенным ресурсами, роли активного агента диалога. Учащийся логически зависит от текста, расширенного ресурсами. Учащийся не может сформировать произвольный ответ, а должен вернуть тексту, расширенному ресурсами, релевантную порцию декларативных знаний. Будем, следуя Андерсону [55], называть эту порцию декларативных знаний — «chunk». Учащийся не может сформировать произвольный ответ, поскольку в этом случае нарушится логика диалогового процесса, и он превратится в два независимых монолога.

Таким образом, в пределах одной транзакции текст, расширенный ресурсами, и учащийся передают друг другу взаимосвязанные порции декларативных знаний. Умная страница, которую текст, расширенный ресурсами, передаёт учащемуся, имеет логическую структуру вопроса, а порция знаний, которая передаётся от учащегося к тексту, расширенному ресурсами — логическую структуру ответа. Напоминаем, что мы используем термин «логическая структура», чтобы учесть случаи, когда передаваемые порции декларативных знаний представлены не вербально.

Будем обозначать отмеченные порции знаний, имеющие логическую структуру вопросов и ответов, *SmartPage* и *AnsChunk* соответственно. Умная страница, или *SmartPage*, является переносчиком информации двух типов: фрагмента декларативных знаний, из которых формируется ответ, называемого субъектом страницы, и спецификации желаемого ответа, называемой требованием страницы.

Субъект умной страницы — это «исходный материал» для формирования ответа. Учащийся в процессе формирования ответа использует не все знания, хранящиеся в его декларативной памяти, а только небольшой фрагмент этих знаний, задаваемых субъектом умной страницы. Требование страницы определяет, какая часть субъекта страницы должна перейти в ответ. Таким образом, общую логическую структуру *SmartPage* и *AnsChunk* представим в виде

$$SmartPage \stackrel{\text{def}}{=} Subj, Req, \quad (6.1)$$

$$AnsChunk \in Subj, \quad (6.2)$$

где *Subj* — субъект умной страницы;

Req — требование умной страницы.

Будем рассматривать субъект умной страницы как простой набор объектов, хранящих семантически родственные элементы, а требование — как закодированную спецификацию ответа

$$SmartPage = Req, \{Obj_i\}, i = 1, \dots, n, \quad (6.3)$$

где $\{Obj_i\}, i = 1, \dots, n$ — набор объектов, составляющих субъект умной страницы.

Тогда *AnsChunk* представляет собой подмножество объектов субъекта страницы

$$AnsChunk = \{Obj_i\}, i = 1, \dots, m; m < n. \quad (6.4)$$

Как известно, субъект вопроса позволяет создать множество возможных ответов. Поэтому каждая умная страница *SmartPage* может породить множество возможных ответов, или *AnsChunk*

$$ANS = \{AnsChunk_\beta\}, \beta = 1, \dots, k. \quad (6.5)$$

Текст, расширенный ресурсами, со встроенным интеллектуальным компонентом в зависимости от цели планирует (ожидает) получить и распознать более узкое, чем ANS , множество ответов, названное ранее распознаваемым множеством ответов и обозначенное как R^i (где i — номер транзакции). Распознаваемое множество ответов объединяет ответы, в которых текст, расширенный ресурсами, нуждается для продолжения диалога. Все остальные ответы будем квалифицировать как нераспознаваемые и обозначать NR^i .

Представление структуры умной страницы в виде субъекта и требования позволяет относительно легко решить задачу о вербальном и невербальном способах представления информации в пределах страницы. Ясно, что, по крайней мере, объекты субъекта могут быть представлены невербально. Что касается требования, имеющего статус объяснительной компоненты, то оно должно иметь вербальное представление в виде текста или речи.

В умной странице объекты представляют собой элементы исходного текста и сопутствующие ресурсы. Требование трансформирует информацию, представляемую умной страницей, в мини-задачу и детерминирует интерактивность между учащимся и текстом, расширенным ресурсами.

Таким образом, структура каждой диалоговой транзакции между текстом, расширенным ресурсами, со встроенным интеллектуальным компонентом и учащимся состоит из следующих компонентов:

- имя транзакции;
- имя умной страницы;
- множество распознаваемых ответов;
- множество имен транзакций, соответствующих распознаваемым ответам;
- имя транзакции для случая нераспознаваемых ответов.

Взаимодействие между текстом, расширенным ресурсами, со встроенным интеллектуальным компонентом и учащимся предполагает проведение логических умозаключений. Будем рассматривать два уровня умозаключений: (1) умозаключения в пределах транзакции и (2) умозаключения, в результате которых формируется протокол диалога в виде последовательности транзакций (моделируется сценарием диалога).

Умозаключения в пределах транзакции разобьём на три фазы:

- умозаключения текста, расширенного ресурсами, при формировании умной страницы и до получения ответа от учащегося;
- умозаключения учащегося при подготовке ответа;
- умозаключения текста, расширенного ресурсами, после получения ответа учащегося.

Логическая зависимость учащегося от текста, расширенного ресурсами, со встроенным интеллектуальным компонентом управляется последним во время формирования умной страницы. Текст, расширенный ресурсами,

может рассматривать умную страницу как «психологический сенсор», помещённый в сознание учащегося с целью получения необходимой порции знаний. Фрагмент знаний, куда помещается отмеченный «психологический сенсор», определяется субъектом страницы, в то время как требование страницы специфицирует ту часть информации, которая выделяется из субъекта и передаётся в ответе.

Упрощённые рассуждения текста, расширенного ресурсами, со встроенным интеллектуальным компонентом при синтезе умной страницы можно представить следующим образом: «Для продолжения текущего диалогового поведения мне необходима очередная порция знаний от учащегося. Я предполагаю, что необходимая порция знаний находится в пределах данного субъекта умной страницы. Учащийся должен сформировать необходимую порцию знаний из данного субъекта в соответствии с ограничениями, задаваемыми требованием страницы».

Логика учащегося в процессе формирования ответа — это «логика селекции или экстракции». Учащийся формирует ответ из субъекта страницы в соответствии с требованием путём селекции некоторого подмножества объектов.

После получения ответа текст, расширенный ресурсами, со встроенным интеллектуальным компонентом должен принять решение о том, какую следующую транзакцию необходимо выбрать для продолжения текущего диалогового поведения. Таким образом, после того, как получен ответ, текст, расширенный ресурсами, со встроенным интеллектуальным компонентом должен выбрать очередную транзакцию шаг в соответствии с логикой диалогового поведения, воплощенной в сценарии диалога.

Ранее (см. рис. 4.10) сценарий эротетического диалога был определен как совокупность памяти вопросов *QueMem* и диалогового метода доступа к памяти вопросов *DiAM*. Введем параллельный термин «диалоговая база знаний» *DiKB*, под которой также будем понимать совокупность *QueMem* и *DiAM*. Термин «диалоговая база знаний» вполне правомерен, поскольку, с точки зрения процедурно-декларативной дихотомии знаний, память вопросов является хранилищем декларативных знаний, а диалоговый метод доступа к памяти вопросов — хранилищем процедурных. Применительно к тематике текста, расширенного ресурсами, со встроенным интеллектуальным компонентом память вопросов является хранилищем умных страниц *SmartPage*.

С точки зрения логического вывода, *DiKB* может рассматриваться как конечный набор интеррогативных гипотез текста, расширенного ресурсами, со встроенным интеллектуальным компонентом относительно знаний и когнитивных способностей учащегося. Каждая гипотеза эквивалента некоторой цепочке транзакций в пределах *DiKB* (см. рис. 5.9) и может быть описана правилом продукции вида (5.5). Поэтому диалоговый процесс между текстом,

расширенным ресурсами, со встроенным интеллектуальным компонентом и учащимся можно рассматривать как перманентный процесс доказательства истинности последовательности гипотез текста, расширенного ресурсами, относительно знаний и когнитивных способностей учащегося. Иными словами, диалоговый процесс между текстом, расширенным ресурсами, со встроенным интеллектуальным компонентом и учащимся является процессом диагностики декларативных знаний и когнитивных способностей учащегося при помощи дедуктивного процесса.

Таким образом, знания, регламентирующие диалоговое поведение текста, расширенного ресурсами, со встроенным интеллектуальным компонентом, хранятся в *DiKB*, а «материализация» отмеченного диалогового поведения осуществляется при помощи унифицированной процедуры *DiMC* и процедур демонов *QueDemon* и *AnsDemon* (см. подраздел 3.2).

Рассмотрим вариант внутренней организации унифицированной процедуры *DiMC* в виде команды программных агентов. Несмотря на то, что предлагаемая архитектура разработана в рамках проекта «текст, расширенный ресурсами, со встроенным интеллектуальным компонентом», она может рассматриваться как способ реализации унифицированной процедуры *DiMC* в общем случае. Мы интерпретируем понятие «агент» так, как оно было впервые предложено Марвином Мински [217]. Агент — это любая часть сознания, структура и поведение которого относительно просты и понятны, в то время как взаимодействие «команды» таких агентов может продуцировать феномен разумного поведения, понять который значительно сложнее.

Произвольный агент на каждом шаге диалога демонстрирует специфическое поведение, которое определяется его внутренней структурой и состоянием внутреннего и внешнего мира процедуры *DiMC*. Здесь под внешним миром процедуры *DiMC* мы понимаем модель знаний учащегося, а под внутренним миром — модель знаний текста, расширенного ресурсами, со встроенным интеллектуальным компонентом, хранящихся, главным образом, в *DiKB*.

Предлагаемая многоагентная организация процедуры *DiMC* разработана с целью обработки произвольной транзакции, а поведение каждого из агентов команды определяется логикой эротетического диалогового процесса, а не логикой предметной области. Все особенности предметной области хранятся в *DiKB*. Агенты действуют асинхронно, последовательно и циклически передают активность друг другу. Совмещение диалогового процесса и процесса дедуктивного вывода заложено в сценарий диалога и отражено в структуре *DiAM*.

Рис. 6.8 отражает структуру многоагентной организации *DiMC*. На рис. 6.8 компоненты, соответствующие внутреннему миру процедуры *DiMC*, расположены внутри цикла (изображен жирной линией). Предполагается, что внешний мир процедуры *DiMC* расположен снаружи цикла.

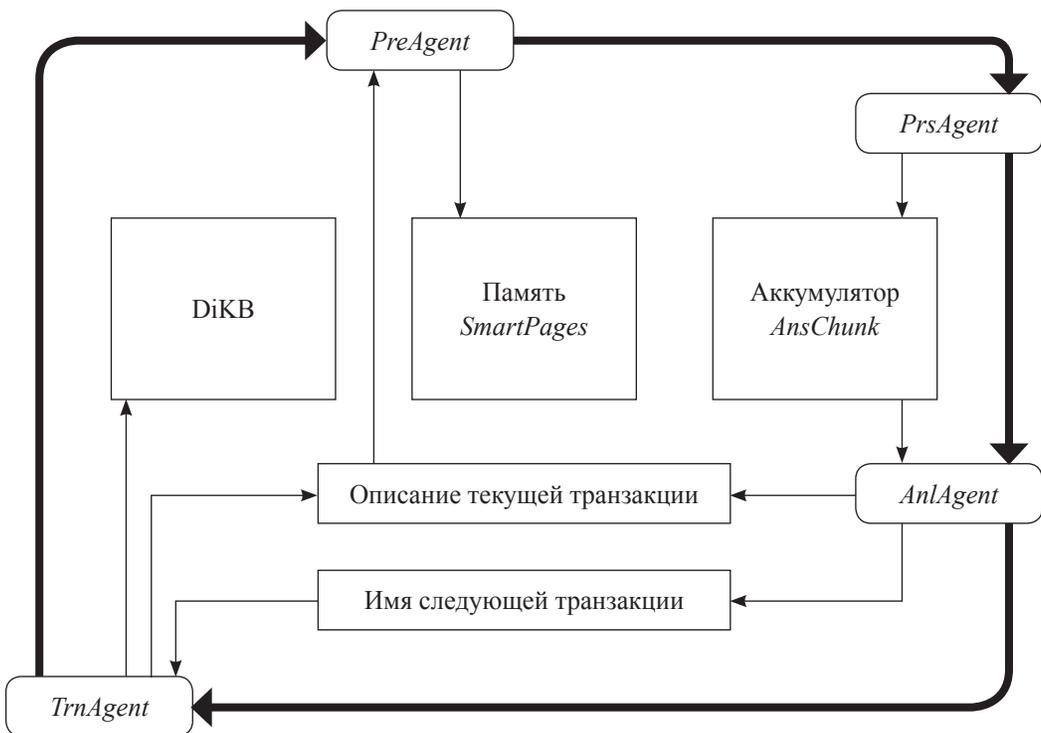


Рис. 6.8. Структура многоагентной организации процедуры *DiMC*

Функционирование *DiMC* детерминировано поведением следующих четырех агентов:

- (1) презентационный агент (*PreAgent*);
- (2) воспринимающий агент (*PrsAgent*);
- (3) анализирующий агент (*AnAgent*);
- (4) агент транзакции (*TrnAgent*).

Поведение презентационного агента. Презентационный агент осуществляет презентацию умной страницы *SmartPage* учащемуся. Например, *SmartPage* может соответствовать вопросу, который спроектирован для получения фактов с целью вывода заключений о степени понимания фрагмента исходного текстового документа учащимся. Взаимодействие с внешним миром однонаправленное, от *PreAgent* к учащемуся. Взаимодействие с внутренним миром осуществляется следующим образом. Из блока памяти, который хранит описания текущей транзакции, *PreAgent* читает имя *SmartPage*, а затем — всю информацию, необходимую для презентации *SmartPage*.

Поведение воспринимающего агента. Агент воспринимает *AnsChunk* от учащегося и аккумулирует их в аккумуляторе *AnsChunk*. Таким образом,

аккумулятор *AnsChunk* хранит историю ответов учащегося. Взаимодействие с внешним миром однонаправленное, от учащегося к *PrsAgent*. Ответы учащегося воспринимаются и кодируются при помощи стандартного периферийного оборудования компьютера. Взаимодействие с внутренним миром осуществляется путем накопления *AnsChunk* в аккумуляторе.

Поведение анализирующего агента. Анализирующий агент распознаёт воспринятый *AnsChunk* путём его последовательного сравнения с элементами множества распознаваемых ответов R^i и определяет имя следующей транзакции. *AnlAgent* не взаимодействует с внешним миром. Его взаимодействие с внутренним миром осуществляется следующим образом. *AnlAgent* читает *AnsChunk* из аккумулятора и R^i из блока памяти, хранящего описание текущей транзакции. Затем определяет имя следующей транзакции и помещает его в память имени следующей транзакции.

Поведение агента транзакции. Агент транзакции обновляет описание текущей транзакции. Агент транзакции не взаимодействует с внешним миром. Его взаимодействие с внутренним миром осуществляется следующим образом. Агент транзакции читает имя следующей транзакции из соответствующего блока памяти, а затем описание этой транзакции из *DiKB*. Полученное описание транзакции помещается в память описаний текущей транзакции.

Многоагентная организация процесса *DiMC* предполагает некоторый стандарт в понимании эротетического диалогового процесса, который опирается на результаты его даталогического представления, полученные ранее в разделах 3 и 4. Например, умная страница собирается из некоторого, типизированного количества объектов сцены (см. раздел 4). Имя следующей транзакции однозначно определяется распознанным *AnsChunk* и хранится в описании транзакции сценария диалогового метода доступа. Унифицированный процесс *DiMC* не может выполнять вычислений.

Для расширения рамок отмеченного стандарта предполагается возможность нарушения цикла активизации агентов, изображенного на рис. 6.8 в двух точках: (1) во время работы презентационного агента *PreAgent* и (2) во время работы анализирующего агента *AnlAgent*. Возможные нарушения цикла активизации агентов планируются в описателях шага. При прерывании цикла управление передаётся специализированным агентам, названным ранее процедурами-демонами *QueDemon* и *AnsDemon* (см. подраздел 3.2)). Агенты-демоны реализуют все необходимые нестандартные функции унифицированного процесса *DiMC*.

6.3. Интеллектуальные системы автоматизированного проектирования

Предыдущие подразделы настоящего раздела посвящены, главным образом, исследованию применимости эротетического диалога к решению плохо формализуемых задач в таких областях, где применение классических методов математического моделирования либо крайне затруднено, либо просто невозможно (персонализированные системы электронного обучения и электронный текст со встроенным диалоговым компонентом). Однако не менее актуальна разработка методов решения плохо формализуемых задач в таких областях, где уже на протяжении длительного времени используются классические методы математического моделирования и компьютерная обработка данных. Такими областями являются автоматизированное проектирование и конструирование. Часто в отмеченных областях основные усилия разработчиков направлены на создание и совершенствование методо-ориентированного ядра системы, поскольку здесь можно опереться как на хорошо разработанный математический аппарат, так и на традиционные технологии программирования [218, 219].

Автоматизации процесса решения плохо формализуемых задач, окружающего методо-ориентированное ядро таких систем, уделяется значительно меньше внимания. Однако такое распределение приоритетов снижает эффект от применения системы автоматизированного проектирования (САПР). В качестве примера можно рассмотреть процесс теплофизического проектирования элемента конструкции энергетической установки. Расчётный блок в такой системе используется для определения полей термических напряжений с помощью либо метода сеток, либо метода конечных элементов. В зависимости от производительности компьютера и количества используемых элементов этот процесс продолжается единицы или десятки минут, в то время как подготовка сетевой модели и данных, необходимых для расчета, занимает недели и месяцы работы эксперта. Не менее трудоёмок процесс интерпретации экспертом полученных массивов чисел, т. е. переход от полей напряжений к конкретным конструкторским решениям. Использование компьютера, как на начальном этапе проектирования, так и на этапе интерпретации расчётных результатов предполагает умение решать плохо формализуемые задачи, имитирующие работу человека-эксперта.

Будем считать, что в общем случае интеллектуальная САПР состоит из трёх блоков:

1. Проблемно-ориентированный блок, предназначенный для снабжения расчётного ядра системы данными, необходимыми и достаточными для выполнения расчётов.
2. Методо-ориентированный блок, предназначенный для выполнения расчётов.

3. Проблемно-ориентированный блок, предназначенный для интерпретации результатов, полученных с помощью блока 2.

Блок 2 строится на базе одного из известных численных методов и реализуется сугубо алгоритмически. Блоки 1 и 3 — это совокупность плохо формализуемых задач, методы решения которых представляют собой эвристические правила, объединённые логикой эксперта.

В табл. 6.3 приведены несколько примеров трёхблочной организации интеллектуальной САПР.

Таблица 6.3

Примеры трёхблочной организации интеллектуальной САПР

Область применения САПР	Назначение блока		
	Блок 1	Блок 2	Блок 3
Теплофизическое проектирование	Генерация конечно-разностного аналога	Расчёт температурного поля методом сеток	Анализ температурного поля. Генерация данных для параметрических расчётов
Прочностное конструирование	Генерация конечно-элементного аналога	Расчёт поля напряжений методом конечных элементов	Анализ поля напряжений. Генерация данных для параметрических расчётов
Проектирование интегральных схем	Генерация логической схемы	Поверочный расчёт методом верификации электронных схем	Анализ результатов верификации. Редактирование логической схемы

6.3.1. Диалоговая имитация работы эксперта при моделировании явлений теплопереноса

Настоящий подраздел посвящён исследованию применимости эротетического диалога для решения плохо формализуемых задач при моделировании явлений теплопереноса (первая строка табл. 6.3). Результаты исследований легли в основу пакета программ под наименованием «Термоанализатор», разработанного в Одесском институте низкотемпературной техники и энергетики по заданию НИИ «Шторм» (г. Одесса) в середине 90-х годов. Пакет обеспечивает замкнутый цикл моделирования (ввод данных — расчёт — отображение результатов) стационарных и нестационарных полей температур, неоднородных по своему составу объектов произвольной геометрической формы при разнообразных

граничных условиях. Описание пакета программ «Термоанализатор» можно найти в [220].

Основная задача, которая ставилась при разработке пакета «Термоанализатор», заключалась в проектировании интеллектуального компонента, хранящего эвристические знания специалистов в области моделирования тепломассопереноса в таком виде, который делает их доступными для использования неспециалистами. Интеллектуальный компонент, называемый также экспертной оболочкой, реализовывал функции первого и третьего блоков трёхблочной организации интеллектуальной САПР (см. табл. 6.3).

В методо-ориентированном (втором) блоке пакета программ «Термоанализатор» реализована технология математического моделирования теплового состояния объектов методами прямой аналогии. В основе метода — сведение дифференциального оператора (уравнение теплопроводности, начальные и граничные условия) к конечно-разностному. С этой целью исследуемый объект заменяется топологически адекватной ему сеточной моделью (дискретным аналогом), для каждого узла которой составляется уравнение теплового баланса, совокупность которых образует систему линейных алгебраических уравнений [221].

Система формирования дискретного аналога, входящая в состав экспертной оболочки, является диалоговой и по форме, и с точки зрения метода решения плохо формализуемых задач.

Особенностью реализации диалогового метода в экспертной части пакета программ «Термоанализатор» является фиксированное закрепление роли активного агента диалога за программой. Таким образом, в ходе формирования дискретного аналога вопросы всегда задаёт программа, а отвечает на эти вопросы всегда оператор-пользователь. Программа «знает», с какого вопроса необходимо начать диалог и каким очередным вопросом необходимо отреагировать после получения ответа на предыдущий вопрос. Вопрос, являющийся переносчиком декларативных знаний, строится в соответствии с описанной выше теорией, рассматривающей логическую структуру вопроса как совокупность субъекта вопроса и требования вопроса. На рис. 6.9 приведен фрагмент модели диалогового метода «оценка мерности модели» в виде сети Петри.

Как видно из фрагмента сценария, приведенного на рис. 6.9, активный агент диалога (программа «Термоанализатор») на каждой транзакции ожидает ответ «Да» или «Нет». Для упрощения сценария, при определении имени следующего шага используется процедура-демон. Аккумулятор ответов оператора представляет собой два одномерных массива: $M1[1..3]$ и $M2[1..3]$, хранящих комбинации нулей и единиц. Для определения имени следующей транзакции агент-демон использует содержимое массива $M1$, представляющее собой целое число в диапазоне от 1 до 8.

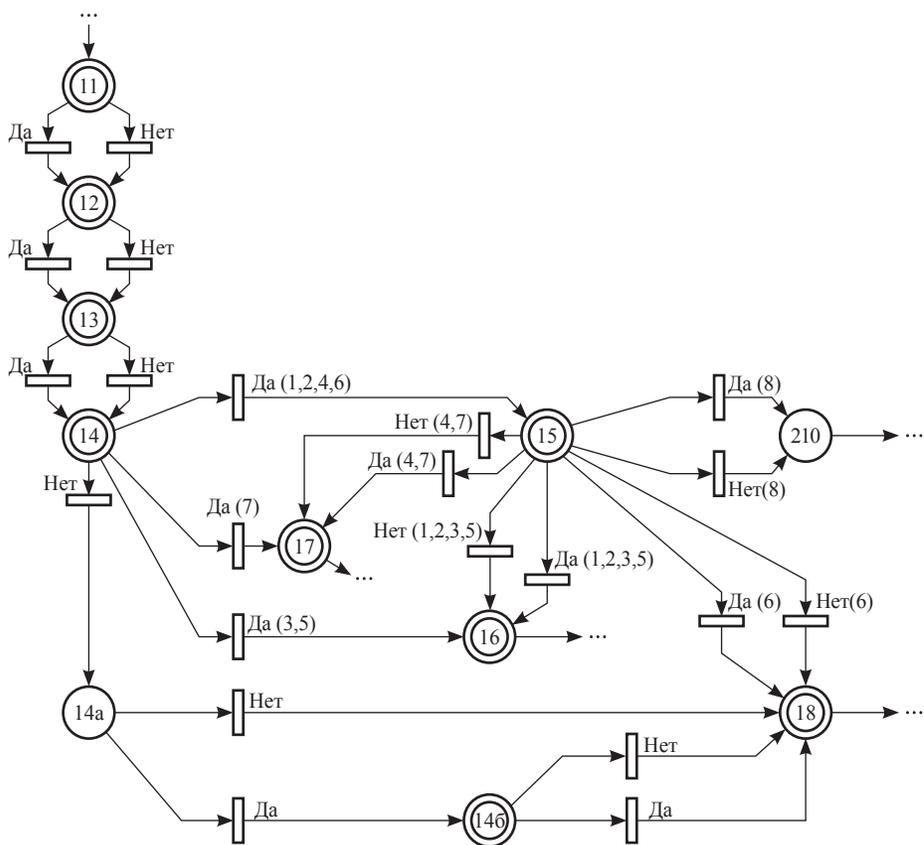


Рис. 6.9. Фрагмент модели диалогового метода «оценка мерности модели»

На рис. 6.9 приняты следующие обозначения.

Позиция, изображённая в виде удвоенной окружности, означает, что при вычислении имени следующей транзакции используется агент-демон, а позиция, изображённая в виде одинарной окружности, означает, что при вычислении имени следующей транзакции агент-демон не используется.

Переходы надписаны ответами «Да» или «Нет». Если после перечисленных ответов изображены круглые скобки, то переход является условным. Он выполняется в том случае, когда, кроме утвердительного или отрицательного ответа оператора, учитывается возвращаемое демоном значение. Список чисел в скобках является альтернативным. Например, если переход имеет надпись: «Да (1,2,4,6)», то это означает, что оператор ответил «Да», а демон вернул одно из чисел в скобках.

В табл. 6.4 приведена расшифровка используемых на рис. 6.9 вопросов и ответов.

Таблица 6.4

Табличное описание сценария «оценка мерности модели»

Имя транз.	Содержание вопроса	Ожидаемый ответ	Аккумулятор ответов		Значение, возвращаемое демоном	Имя следующей транзакц.
			M1	M2		
11	Можно ли предположить однородность материала вдоль оси X?	Да	M1[1]:= 0			12
		Нет	M1[1]:= 1			12
12	Можно ли предположить однородность материала вдоль оси Y?	Да	M1[2]:= 0			13
		Нет	M1[2]:= 1			13
13	Можно ли предположить однородность материала вдоль оси Z?	Да	M1[3]:= 0		1..7	14
		Нет	M1[3]:= 1		8	210
14	Есть ли на поверхностях, ограничивающих объект по оси X, участки, температура которых известна до решения задачи?	Да		M2[1]:= 1	1, 2, 4, 6	15
					3, 5	16
					7	17
		Нет				14a
14a	Поверхности, ограничивающие объект по оси X, теплоизолированы?	Да				14b
		Нет				18
14b	Теплообменом между объектом и окружающей средой по этим поверхностям в направлении оси X можно пренебречь?	Да		M2[1]:= 0		18
		Нет	M1[1]:= 1			18
15	Температуры хотя бы на одной из этих поверхностей по координате U изменяются?	Да	M1[2]:= 1		1, 2, 3, 5	16
					4, 7	17
		Нет	M1[2]:= 0		6	18
					8	210
210	Ваш объект может быть представлен трёхмерной моделью (сеточным аналогом). Правомерность применения модели меньшей мерности можно установить только на основе анализа результатов моделирования	Нажатие любой клавиши				Главное меню системы

Помимо своего основного назначения, подсистема оценки мерности может быть полезна начинающему оператору в приобретении навыков математического моделирования теплового состояния объектов, а также углубления представления оператора о моделируемом объекте и его взаимодействии с окружающей средой.

Интеллектуальная оболочка пакета программ «Термоанализатор» включает следующие подсистемы:

- подсистема экспертной оценки мерности математической модели;
- подсистема формирования сеточной модели;
- подсистема автоматизированного вычисления коэффициентов теплообмена;
- подсистема обеспечения справочной информацией;
- подсистема интерпретации результатов моделирования;
- архив сеточных моделей.

Концепция построения подсистемы автоматизированного вычисления коэффициентов теплообмена предполагает использование двух подходов:

объектно-независимый подход, предполагающий, что определение коэффициента теплообмена выполняется на основе детальных характеристик свойств поверхности, омывающей её среды и т. д.;

объектно-зависимый подход, предполагающий, что определение коэффициентов теплообмена осуществляется для конкретных объектов (диски газовых турбин, котельные агрегаты, элементы радиоэлектронной аппаратуры и т. д.) в конкретных условиях эксплуатации.

Поиск необходимого процесса для вычисления коэффициентов теплообмена осуществляется в диалоговом режиме. Вводя качественные характеристики взаимодействия поверхности объекта с окружающей средой, оператор получает доступ к зависимости или группе зависимостей, позволяющих определить количественную характеристику конвективного теплообмена — коэффициент теплообмена и использовать её при формировании сеточной модели. Аналогичным образом осуществляется поиск и использование зависимостей для расчёта коэффициента теплообмена излучением или сложного теплообмена.

Подсистема интерпретации результатов моделирования представлена двумя опциями:

- отображением температурного поля в виде таблицы;
- отображением профиля температур для заданной оператором последовательности узлов сеточной модели.

6.3.2. Организация интеллектуальной системы автоматизированного проектирования общего назначения

Эффективность эксплуатации приложений, разработанных в среде промышленных систем автоматизированного проектирования общего назначения,

особенно высока в тех случаях, когда проекты повторяются с незначительными вариациями. В тех случаях, когда объём изменений значителен, эксплуатация прикладных систем автоматизированного проектирования становится существенно личностно-зависимой. Проектирование в этом случае целиком зависит от эксперта, создавшего необходимые базы данных и накопившего процедурные знания как о предметной области, так и о системе автоматизированного проектирования общего назначения. Известны случаи, когда эксплуатация прикладной системы автоматизированного проектирования и, следовательно, выполнение соответствующих работ полностью прекращались в связи с увольнением эксперта.

Интеллектуальная система автоматизированного проектирования, доведенная до уровня промышленной эксплуатации, обслуживается оператором, и для обеспечения её функционирования не нужен эксперт, поскольку такая система в состоянии хранить «know-how» знания и воспроизводить логику эксперта в процессе проектирования.

В настоящем подразделе описана интеллектуальная система автоматизированного проектирования, разработанная и испытанная в 90-е годы в Лаборатории диалоговых систем Одесского института низкотемпературной техники и энергетики по заказу Инвестиционно-информационного центра Киевского института Гипрохиммаш.

Одной из особенностей практической реализации системы является использование готовых пакетов программ общего назначения: *AUTOCAD* [222] и Процессор диалога [130]. Рис. 6.10 иллюстрирует распределение функций в системе.

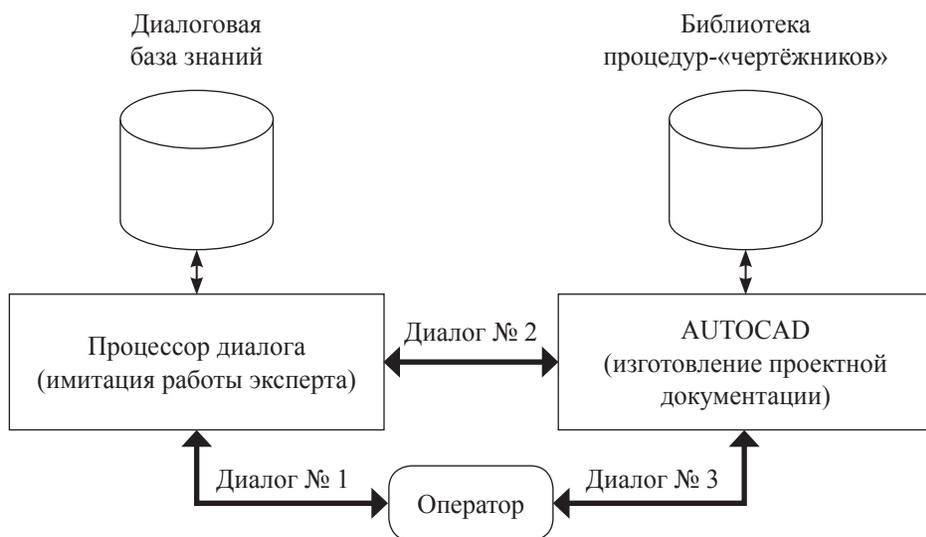


Рис. 6.10. Три типа диалоговых процессов в интеллектуальной САПР

Для реализации функций системы используются три диалоговых процесса.

Диалоговый процесс №1 является основным и осуществляется между оператором и пакетом программ Процессором диалога. В этом диалоговом процессе пакет программ Процессор диалога реализует роль активного агента диалога, а оператор — реактивного. В ходе диалогового процесса №1 Процессор диалога, интерпретируя сценарий диалоговой базы знаний, задаёт оператору серию вопросов с целью конкретизации и уточнения объекта проектирования. Сценарий хранит процедурные знания о проектировании объекта «вообще», например, знания о том, как проектировать принципиальные схемы трубных/электрических соединений. В ходе диалогового процесса №1 оператор снабжает Процессор диалога информацией, необходимой для превращения абстрактного объекта проектирования в конкретный.

Диалоговый процесс №2 осуществляется между пакетами программ Процессором диалога и *AUTOCAD*. В этом диалоговом процессе пакет программ Процессор диалога реализует роль активного агента диалога, а пакет программ *AUTOCAD* — реактивного. Диалоговый процесс №2 необходим для передачи в *AUTOCAD* команд на выполнения конкретных действий и получения от *AUTOCAD* ответов о выполнении команд. Особенность диалога №2 заключается в том, что он не визуализируется и, в отличие от человеко-машинного диалога №1, может иметь простую организацию транзакций.

Диалоговый процесс №3 осуществляется между пакетом программ *AUTOCAD* и оператором. В этом диалоговом процессе пакет программ *AUTOCAD* реализует роль активного агента диалога, а оператор — реактивного. Вообще говоря, все функции диалога №3 может выполнять и диалог №1. Однако в тех случаях, когда необходимо уточнить параметры чертежа, диалог №3 разгружает сценарий проектирования основного диалога от излишних «графических» подробностей.

При проектировании интеллектуальной системы автоматизированного проектирования решалась задача выбора системной архитектуры компьютера. Эта задача имеет принципиальное значение, поскольку от её решения зависят как затраты на разработку системы (временные и материальные), так и её эксплуатационные качества (надёжность и скорость проектирования).

Рассматривались следующие возможные архитектурные варианты:

- одномониторный однопроцессорный компьютер в многозадачном режиме;
- двухмониторный (двухадаптерный) однопроцессорный компьютер;
- двухмашинный сетевой комплекс на базе стандартных портов с серийным или заказным сетевым драйвером.

В результате анализа всех «за» и «против» был выбран двухмашинный сетевой комплекс. Двухмашинная архитектура обладает, по крайней мере, следующими положительными качествами:

- два процессора обеспечивают одновременную и асинхронную работу пакетов программ Процессор диалога и *AUTOCAD*;
- два монитора позволяют одновременно наблюдать диалоговый процесс №1 и создаваемый чертёж в реальном масштабе времени;
- суммарные вычислительные ресурсы разделяются на две независимые части, поэтому каждый из компьютеров сети может обладать примерно половиной необходимых суммарных ресурсов.

На рис. 6.11 приведены основные компоненты архитектуры двухмашинной интеллектуальной САПР.

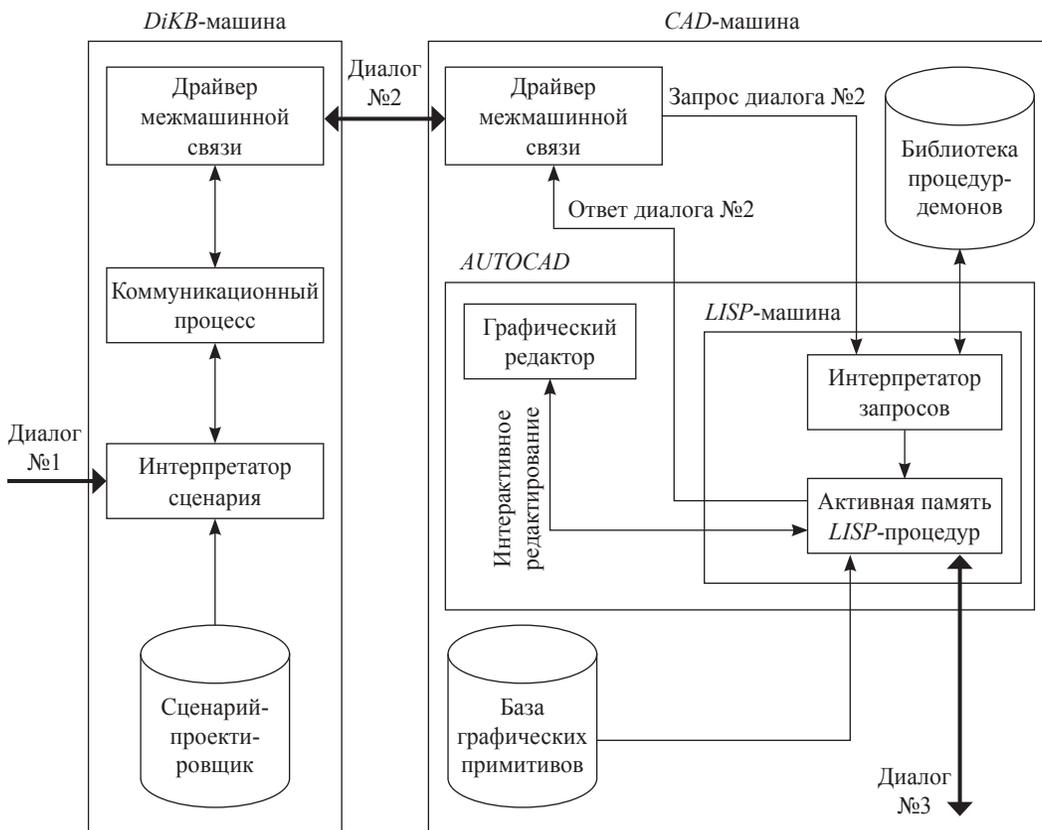


Рис. 6.11. Архитектура двухмашинной интеллектуальной САПР общего назначения

Компьютер, предназначенный для функционирования пакета Процессор диалога, носит наименование «машина диалоговой базы знаний» («*DiKB-машина*»), а компьютер, предназначенный для функционирования *AUTOCAD* — «*CAD-машина*».

Межмашинная связь организуется с помощью двух симметричных драйверов межмашинной связи, основное назначение которых — реализация диалогового процесса №2. Пакет программ Процессор диалога представлен двумя блоками: (1) диалоговая база знаний, хранящая «сценарий-проектировщик», и (2) интерпретатор диалоговой базы знаний (интерпретатор сценария).

Интерпретатор сценария циклически интерпретирует диалоговые транзакции и содержит фиксированный набор интерпретирующих процедур. В том случае, когда необходимо выполнить действия, не предусмотренные циклом интерпретации транзакции, то изготавливается внешний резидентный процесс. На рис. 6.11 такой процесс носит наименование коммуникационный процесс. Коммуникационный процесс решает следующие задачи: (1) формирование запроса для диалогового процесса №2, (2) передача сформированного запроса в драйвер межмашинной связи, а также (3) приём и обработка ответа диалогового процесса №2. Как только запрос диалогового процесса №2 сформирован, он с помощью драйвера межмашинной связи попадает в *CAD*-машину.

Драйвер межмашинной связи работает в ждущем режиме и активизируется только в случае появления данных, необходимых для передачи.

Основным компонентом *CAD*-машины является система *AUTOCAD10*. В её составе выделим графический редактор и *LISP*-машину.

Графический редактор реализует традиционные для системы автоматизированного проектирования общего назначения функции и предназначен для создания чертежа проекта в интерактивном режиме. Однако сам интерактивный режим в интеллектуальной системе автоматизированного проектирования отличается от традиционного. Традиционным интерактивным режимом создания и редактирования чертежа будем считать такой режим, когда инициатива создания чертежа принадлежит пользователю, который в принятии решений основывается только на своих знаниях и опыте. Помочь ему принять то или иное решение система автоматизированного проектирования неспособна.

В интеллектуальной системе автоматизированного проектирования инициатива в создании и редактировании чертежа распределена между пользователем, системой автоматизированного проектирования общего назначения и Процессором диалога. Процессор диалога выполняет функции квалифицированного эксперта в некоторой конкретной области проектирования. Как правило, пользователю достаточно ввести ответ в ходе диалогового процесса №1, и все ресурсы графического редактора будут использованы для генерации чертежа или его части либо без участия пользователя, либо с его минимальным участием. Например, для упрощения сценария диалога №1 можно точку расположения графического объекта определять не в ходе диалогового процесса, а непосредственно на экране *CAD*-машины.

Для поддержки отмеченной технологии создания чертежа используются возможности *LISP*-машины системы *AUTOCAD* (см. рис. 6.11). Ключевым

компонентом *LISP*-машины является интерпретатор запросов, сформированных в процессе диалога №1 и переданных в *CAD*-машину по каналу машинной связи. Основная задача интерпретатора — распознавание запроса, определение процедуры-демона, соответствующего запросу, активизация процедуры-демона и передача ему параметров запроса. После завершения работы процедуры-демона управление возвращается интерпретатору для обработки следующего запроса.

Процедуры-демоны являются основным инструментом для выполнения операций автоматического изображения стандартных графических объектов. При работе процедуры-демона в большинстве случаев от пользователя требуется только минимальная активность: указать точку привязки, размер объекта и т. д. В идеальном случае процедура-демон может представлять собой параметризованный чертёж, настраиваемый пользователем на конкретный проект при помощи параметров. Таким образом, библиотека процедур-демонов является неотъемлемой и весьма существенной частью *CAD*-машины.

Активизация процедур-демонов осуществляется следующим образом. Исходное состояние процедуры-демона — это состояние ожидания уникального условия активизации. Условие активизации может возникнуть в результате интерпретации запроса, полученного от *DiKB*-машины по каналу межмашинной связи. В результате интерпретации запроса генерируется условие для активизации одного из демонов. После активизации демон получает из запроса (если это необходимо) фактические значения параметров и выполняет свою функцию. После завершения работы процедура-демон возвращает управление интерпретатору запроса. Процедура-демон может передать управление пользователю (например, для уточнения точки расположения графического объекта). В этом случае пользователь сам возвращает управление интерпретатору запроса.

Библиотека процедур-демонов содержит как процедуры общего назначения, так и процедуры, разработанные для реализации определённого проекта. Более подробное описание двухмашинной диалоговой системы автоматизированного проектирования можно найти в [130].

6.4. Диалоговый метод формирования запросов к базе данных

Современные средства манипулирования данными в базах данных всё в большей степени ориентированы на пользователя-непрофессионала. Одна из тенденций заключается в возрастании «степени непроцедурности» в языках манипулирования данными (ЯМД). В [223] приведена классификация ряда ЯМД, используемых для формирования запросов к базам данных, описываемых реляционной моделью. Рассмотрен язык структурированных запросов *Sequel*. Здесь же приведены следующие сведения. Среднее время обучения непрофессионала

языку Sequel составляет 10–12 часов, а время формирования запроса средней сложности — от 3 до 5 минут.

Языки структурированных запросов, такие как *Sequel*, *QUEL* и, в особенности, *SQL (Structured Query Language)* являются в настоящее время одним из основных средств формирования запросов к реляционным базам данных. В [224] отмечается, что запросы на этих языках формулируются независимо от способа хранения данных, а пользователь избавлен от необходимости решать многие вычислительные проблемы низкого уровня. Таким образом, перечисленные ЯМД — это настоящие языки высокого уровня, и их создание настолько же важно для использования баз данных, насколько важным было в своё время создание языка *ФОРТРАН* для решения численных задач.

На рис. 6.12 приведен пример запроса на *SQL* в диалекте, используемом системой управления базами данных (СУБД) *DB2* [225].

```
SELECT ФАМИЛИЯ
FROM S
WHERE 'P2' IN
      (SELECT НОМЕР_ДЕТАЛИ
       FROM SP
       WHERE НОМЕР_ПОСТАВЩИКА = S.НОМЕР_ПОСТАВЩИКА);
```

Рис. 6.12. Пример запроса на языке *SQL* в диалекте СУБД *DB2*

Приведенный пример *SQL*-запроса можно записать на естественном языке следующим образом: «Определить фамилии поставщиков, которые поставляют деталь P2». Анализ примера *SQL*-запроса, приведенного на рис. 6.12, позволяет сделать два важных вывода:

- для формирования запроса необходимо знать синтаксис языка *SQL*;
- для формирования запроса необходимо знать логическую организацию базы данных.

Такие «дополнительные» знания часто являются непреодолимым барьером для конечного пользователя и, следовательно, ЯМД типа *SQL* не обладает необходимой для конечного пользователя степенью непроцедурности.

В [226] используется термин «фронтальная программа» для обозначения средства, с помощью которого можно скрыть от конечного пользователя *SQL*, в том смысле, что в процессе взаимодействия с фронтальной программой *SQL*-запросы генерируются автоматически. При этом предполагается, что для работы с фронтальной программой необходимы минимальные или нулевые дополнительные

LEVEL SELECTIONS: PAYROLL LEVEL — EMPLOYEE INFO TIME LEVEL — PERIOD INFO PLEASE SELECT A LEVEL () <ответ пользователя>: 1	} Транзакция 1
PAYROLL LEVEL SEARCH SELECTOR: 1. SEARCH ON PERIOD CONTAINED IN 2. SEARCH ON TYPE (TIME FRAME) 3. SEARCH ON PAY (DOLLARS AND CENTS) PLEASE SELECT A SERCH () <ответ пользователя>: 1	} Транзакция 2
PAY-PERIOD CONTAINED IN SEARCH SELECTOR THESE PAY-PERIODS ARE POSSIBLE: 1. 125 (WEEKLY) 2. 243 (BI-WEEKLY) 3. 453 (SEMY-MONTHLY) 4. 895 (MONTHLY) PLEASE SELECT A PAY-PERIOD () <ответ пользователя>: 4	} Транзакция 3
ANY MOE SEARCH CRITERIA: (Y/N) <ответ пользователя>: Y	} Транзакция 4
AND/OR: () <ответ пользователя>: AND	} Транзакция 5

Рис. 6.14. Запрос в виде фрагмента диалога.
 Фигурными скобками отмечены диалоговые транзакции

Приведенные примеры непроцедурного взаимодействия конечного пользователя и базы данных используем для введения классификации способов получения новых данных из базы. Разделим способы получения данных из базы на два класса:

- прямое интеррогативное взаимодействие с базой данных;
- инверсное интеррогативное взаимодействие с базой данных.

Прямое интеррогативное взаимодействие с базой данных предполагает, что данные из базы данных передаются пользователю в виде ответа на введенный вопрос. Пользователь при прямом интеррогативном взаимодействии с базой данных играет роль активного (спрашивающего) агента.

При инверсном интеррогативном взаимодействии с базой данных запрос к базе данных формируется автоматически фронтальной программой из ответов пользователя на вопросы фронтальной программы. Пользователь при инверсном

интеррогативном взаимодействии с базой данных является реактивным (отвечающим) агентом.

Большая часть приведенных выше примеров иллюстрирует прямой интеррогативный способ получения данных из базы. Сюда относятся языки структурированных запросов: *Sequel*, *QUEL*, *SQL*, ряд систем, использующих фронтальные программы (*QBE* и *NFQL*). Только один из приведенных примеров может быть отнесен к инверсному интеррогативному способу получения данных из базы данных — диалог, приведенный на рис. 6.14.

Не проводя подробно сравнительный анализ прямого и инверсного способов получения данных из базы данных, отметим одно существенное отличие — реактивность конечного пользователя при инверсном интеррогативном взаимодействии. Реактивность пользователя позволяет строить компьютерные системы, требующие от пользователя минимальных либо нулевых сведений о внутренней организации компьютерной системы. Следовательно, применение инверсного интеррогативного способа получения данных из базы данных во фронтальных программах позволяет разрабатывать пакеты программ, обладающие максимальной степенью непроцедурности при формулировке запросов к базе.

Теоретической основой метода инверсного интеррогативного взаимодействия могут быть результаты, полученные при разработке формальных моделей эротетического диалогового процесса (см. раздел 3) и даталогической модели структуры вопроса (см. раздел 4). Эти результаты позволяют сформулировать общие принципы построения фронтальной программы, ориентированной на инверсное интеррогативное взаимодействие конечного пользователя с базой данных на основе эротетического диалогового процесса.

Фрагмент эротетического сценария, автоматически генерирующей *SQL*-запрос, должен состоять из цепочки транзакций. С точки зрения логической структуры транзакции, вопрос-сцена может состоять из следующих компонентов.

- компонент, при помощи которого детерминируется расширенное множество объектов-вещей (или объектов-свойств) субъекта вопроса;
- компонент, при помощи которого детерминируется ассоциированное с расширенным множеством единственный объект-свойство (или объект-вещь) субъекта вопроса;
- компонент, при помощи которой детерминируется мощность ответа в виде требования вопроса.

Инверсное интеррогативное взаимодействие пользователя с базой данных является универсальным методом построения фронтальных программ для общения с базами данных, которые не требуют от пользователя знаний как о внутренней организации самой системы, так и о структуре базы данных.

Идеи инверсного интеррогативного взаимодействия пользователя с базой данных были экспериментально проверены и описаны в [182].

7 ОРГАНИЗАЦИЯ ПРОБЛЕМНО-НЕЗАВИСИМОГО ДИАЛОГОВОГО РЕШАТЕЛЯ ЗАДАЧ

Диалоговым решателем задач будем называть систему, позволяющую создавать, хранить и редактировать сценарии, необходимые для поддержки эротетического диалогового процесса, ориентированного на решение плохо формализуемых задач. Как декларативные, так и процедурные знания, необходимые для реализации отмеченного диалогового процесса, хранятся в диалоговой базе знаний *DiKB*.

Порции декларативных знаний, имеющие логическую структуру вопроса и ответа, будем обозначать соответственно *QueChunk* и *AnsChunk*. В диалоговом решателе задач принят принцип раздельного хранения декларативных и процедурных знаний. Декларативные знания хранятся в памяти вопросов *QueMem*, а процедурные знания — в памяти диалогового метода доступа *DiAM*.

Мы рассматриваем памяти *QueMem* и *DiAM* с точки зрения метода доступа к информации о текущем и последующих транзакциях диалога. С этой точки зрения, память *QueMem* представляет собой память с произвольным доступом к спецификациям вопросов *QueChunk* и поэтому для получения доступа к спецификациям конкретного вопроса требуется имя соответствующей спецификации. *DiAM* — это сетевая структура, которая способна транслировать текущий ответ реактивного агента *AnsChunk* в имя спецификации релевантного вопроса активного агента. Как было показано выше, *DiAM* может рассматриваться как хранилище метода решения задачи.

Результаты, полученные в настоящем и предыдущих разделах, позволяют синтезировать структуру проблемно-независимого решателя задач. Будем считать, что проблемно-независимый решатель задач представляет собой объектно-ориентированную программную систему, для моделирования структуры которой будем использовать средства диаграмматического языка UML (Unified Modeling Language).

На рис. 7.1 приведена UML диаграмма классов, моделирующая упрощённую структуру проблемно-независимого диалогового решателя задач. Как видно на рис. 7.1, с диалоговой базой знаний *DiKB* ассоциированы два класса: класс *DiMC* (Dialogue Method Cycle), осуществляющий пошаговую интерпретацию знаний,

хранящихся в диалоговой базе знаний, и класс *DiGen* (Dialogue Generator), представляющий собой полноэкранный редактор *DiKB*.

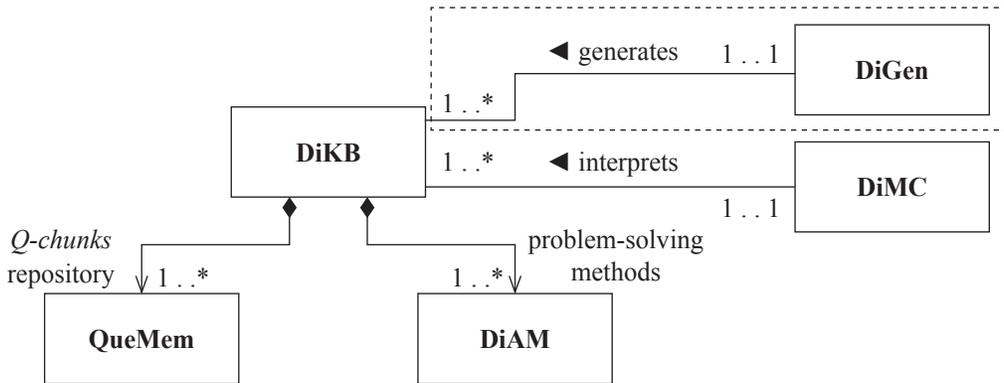


Рис. 7.1. Упрощённая структура проблемно-независимого диалогового решателя задач

Методы класса *DiMC* единообразно интерпретирует любую диалоговую транзакцию, и фазы его работы не зависят ни от специфики шага, ни от специфики предметной области.

После того как сформирована диалоговая база знаний, отпадает необходимость в классе *DiGen*. Этот класс используется на этапе генерации и редактирования диалоговой базы знаний. Отметим, что методы классов *DiMC* и *DiGen* могут быть распределены между классами диалоговой базы знаний, если последняя рассматривается как объектная база.

Используя результаты, полученные в третьем разделе при синтезе логической структуры сети *DiAM* (см. рис. 3.10), синтезируем уточненную структуру диалогового решателя задач.

Уточненная структура диалогового решателя задач в виде UML диаграммы классов приведена на рис. 7.2. Для того, чтобы не загромождать диаграмму классов на рис. 7.2, на ней опущены поля классов диалогового метода доступа *DiAM* (классы *Tran*, *Navig1*, *Navig2*, *QueDemon*, *AnsDemon*). Они тождественны тем, которые приведены на рис. 3.10.

Таким образом, диалоговая база знаний на рис. 7.2 представлена классами *Tran*, *Navig1*, *Navig2*, *QueDemon*, *AnsDemon* и *QueMem*.

Предполагается, что поведение класса *DiMC* реализуется командой программных агентов, предложенных в 6.2.3 (см. рис. 6.8). Каждый из агентов реализуется отдельным методом класса *DiMC*: метод *preAgent* реализует презентационного агента; метод *prsAgent* — воспринимающего агента; метод *anlAgent* — анализирующего агента; метод *trnAgent* — агента транзакции. Поля классов, составляющих диалоговую базу знаний, открыты для методов-агентов класса *DiMC*.

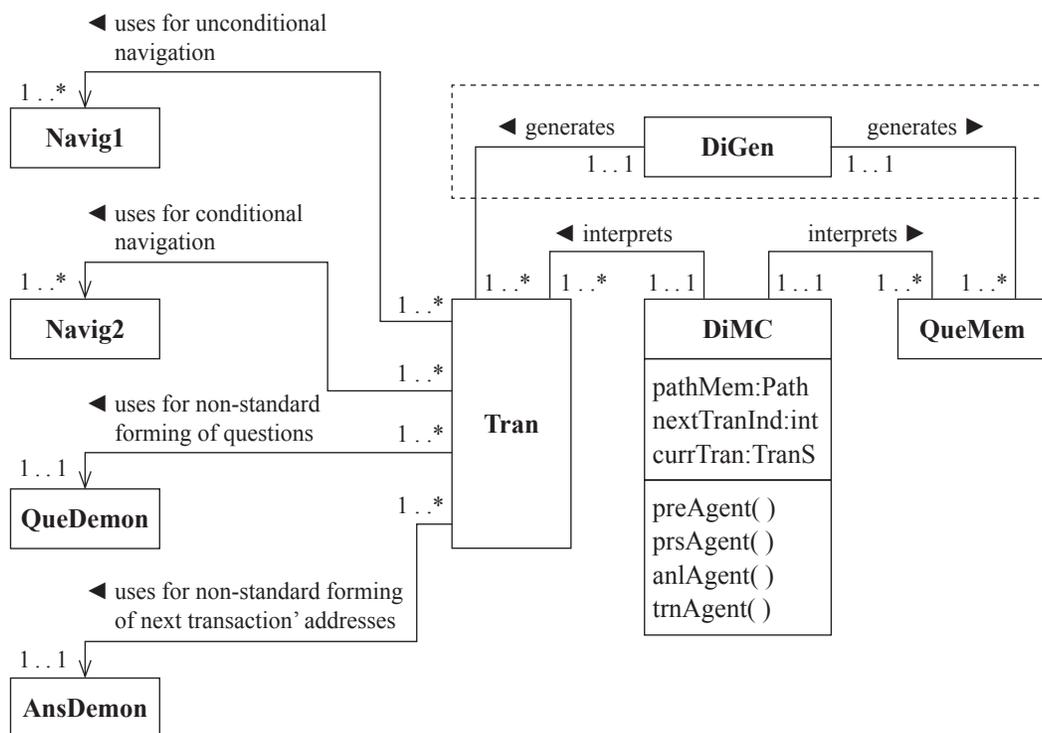


Рис. 7.2. Уточненная структура проблемно-независимого диалогового решателя задач

В класс *DiMC* введены поля, моделирующие «внутренний мир» решателя задач:

- поле *pathMem* моделирует аккумулятор порций декларативных знаний, соответствующих ответам реактивного агента *AnsChunk*;
- поле *currTrans* моделирует описание текущей транзакции;
- поле *nextTransInd* моделирует имя следующей транзакции.

Для класса *DiMC* принят децентрализованный способ управления вызовом методов-агентов. Предполагается, что методы-агенты сами управляют вызовом друг друга (см. рис. 6.8).

Приведенная структура проблемно-независимого диалогового решателя задач легла в основу разработки семейства пакетов программ под общим наименованием Процессор диалога [230–233]. Программы были разработаны в 90-е годы в Лаборатории диалоговых систем Одесского института низкотемпературной техники и энергетики. В приложении приведен пример диалогового приложения, созданного при помощи пакета программ Процессор диалога.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Weizenbaum J. ELIZA — a computer program for the study of natural language communications between man and machine / J. Weizenbaum // Communications of the ACM. — 1966. — Vol. 9. — № 1. — P. 36–45.
2. Вейценбаум Дж. Возможности вычислительных машин и человеческий разум. От суждений к вычислениям / Дж. Вейценбаум — М.: Радио и связь, 1982. — 367 с.
3. ELIZA: a skimmable report on the ELIZA conversational tutoring system / Taylor, Edwin F. (Ed.). Cambridge, Mass: MIT, 1968. — 87 p.
4. Ellis A. B. Can a machine counsel? / A. B. Ellis, D. V. Tiedeman // Proc. CEED-SSRC Conference on Computer-based Instruction. — Austin: University of Texas. 1968. — P. 134–142.
5. Colby K. M. Heuristic methods for computer understanding of natural language in context-restricted on-line dialogues / K. M. Colby, H. Enea // Math. Biosciences. — 1967. — № 1. — P. 1–25.
6. Colby K. M. Inductive Inference by Intelligent Machines / K. M. Colby, H. Enea // Scientia. — 1968. — Vol. 103. — № 1. — P. 169–172.
7. Tesler L. A directed graph representation for computer simulation of belief systems / L. Tesler, H. Enea, K. M. Colby // Math. Biosciences. 1968. — Vol. 21. — № 2 — P. 68–79.
8. Colby K. M. Dialogue between humans and an artificial belief system / K. M. Colby, D. C. Smith // Proc. Int. Conf. Art. Intel. — Washington D. C. 1969. — P. 319–324.
9. Schank R. A conceptual parser for natural language / R. Schank, L. G. Tesler // Proc. Int. Jt. Conf. Art. Intel. — Washington, D. C. 1969. — P. 569–578.
10. Шенк Р. Обработка концептуальной информации / Р. Шенк. — М.: Энергия, 1980. — 360 с.
11. Schank R. C. The cognitive computer on language, learning, and artificial intelligence / R. C. Schank, P. G. Childers. — N. Y.: Addison-Wesley, 1994. — 426 p.
12. Abelson R. P. Implicational molecules: a method for extracting meaning from input sentences / R. P. Abelson, C. M. Reich // Proc. Int. J. Conf. Art. Intel. — Washington, D. C. 1969. — P. 641–647.
13. Slagle J. R. Experiments with a deductive question-answering program / J. R. Slagle // Comm. ACM. — 1965. — Vol. 8. — № 12. — P. 792–798.
14. Слэйгл Дж. Искусственный интеллект / Дж. Слэйгл — М.: Мир, 1973. — С. 201–206.

15. Elliott R. W. A model for a fact retrieval system: Ph.D. Thesis: TNN-42: — Computation Center, University of Texas, Austin, 1965. — 176 p.
16. Green C. C. Research on intelligent question answering systems / C. C. Green, B. Raphael // Proc. ACM 23rd Nat. Conf. 1968. — N. J., Brandon Systems Press, Princeton, — P. 169–181.
17. Green C. Theorem-Proving by Resolution as a Basis for Question-Answering Systems / C. Green // Machine Intelligence. — Vol. 4, Edinburgh Univ. Press, 1969. — P. 183–205.
18. Quillian M. R. The teachable language comprehender / M. R. Quillian // Comm. ACM. — 1969. — Vol. 12. — № 8. — P. 459–476.
19. Quillian M. R. Word concepts: a theory and simulation of some basic semantic capabilities / M. R. Quillian // Behavior Science. — 1967. — № 12. — P. 410–430.
20. Simmons R. F. A computational model of verbal understanding / R. F. Simmons, J. F. Burger, R. M. Schwarcz // Proc. AFIPS 1968 Fall Joint Comput. Conf., Thomson Book Co., Washington, D. C., 1968. — Vol. 33. — P. 441–456.
21. Kay M. A parsing program for computational grammars: Scientific Report / M. Kay; Rand Corp. — RM-4283-PR. — Santa Monica, Calif., 1964. — 90 p.
22. Wenger E. Artificial Intelligence and Tutoring Systems. Computational and Cognitive Approaches to the Communication of Knowledge / E. Wenger. — Los Altos, California, Morgan Kaufmann Publishers., 1987. — 431 p.
23. Carbonell J. R. Mixed-Initiative Man-Computer Instructional Dialogues. Doctoral dissertation. Cambridge. — Massachusetts, Massachusetts Institute of Technology, 1970. — 234 p.
24. Collins A. Reasoning from incomplete knowledge / A. Collins, E. H. Warnock, N. Aiello, M. L. Miller // In Bobrow D. and Collins A. (Eds.) Representation and Understanding: Studies in Cognitive Science. — N. Y., Academic Press, 1975. — P. 18–30.
25. Collins A. Analysis and synthesis of tutorial dialogues / A. Collins, E. H. Warnock, J. Passafiume // In Bower G. (Ed.) The Psychology of Learning and Motivation. — N. Y., Academic Press, 1975. — Vol. 9. — P. 101–109.
26. Stevens A. L. The goal structure of a Socratic tutor / A. L. Stevens, A. Collins // Proceedings of the National ACM Conference in Seattle, Washington; N. Y., Association for Computing Machinery, 1977 — P. 256–263.
27. Collins A. Processes in Acquiring Knowledge / A. Collins // In Anderson R. C., Spiro R. J. and Montague W. E. (Eds.) Schooling and the Acquisition of Knowledge. — New Jersey, Lawrence Erlbaum Associates, 1977. — P. 25–39.
28. Collins A. Goals and strategies for inquiry teachers / A. Collins, A. L. Stevens // In Glaser R. (Ed.) Advances in Instructional Psychology — New Jersey, Lawrence Erlbaum Associates, 1982. — Vol. 2. — P. 165–181.
29. Stevens A. L. Misconception in students' understanding / A. L. Stevens, A. Collins, S. Goldin // Man-Machine Studies — 1979. — Vol. 11. — P. 146–156.
30. Stevens A. L. Multiple models of a complex system / A. L. Stevens, A. Collins // In Snow R., Frederico P. and Montague W. (Eds.) Aptitude, Learning, and Instruction — New Jersey, Lawrence Erlbaum Associates, 1980. — Vol. 2. — P. 225–238.

31. Evens M., W. et al. CIRCSIM-Tutor: An Intelligent Tutoring System Using Natural Language Dialogue // 12th Midwest AI and Cognitive Science Conference. — Oxford OH., 2001. — P. 16–23.
32. Freedman R. K. Plan-Based Dialogue Management in a Physics Tutor / R. K. Freedman // Proc. of the Sixth Applied Natural Language Processing Conference. — Seattle, WA., 2000. — P. 52–59.
33. Freedman R. K. Using a Reactive Planner as the Basis for a Dialogue Agent / R. K. Freedman // Proceedings of FLAIRS. — Orlando, FL., 2000. — P. 203–208.
34. Thomsen-Gray Z. Intelligent Tutoring for Non-Deterministic and Dynamic Domains / Z. Thomsen-Gray, K. Schultz, B. Clark, E. Bratt, S. Peters // Proc. of AI-ED., 2003. — P. 317–332.
35. Fried D. C. The Gerona Knowledge Ontology and Its Support for Spoken Dialogue Tutoring of Crisis Decision Making Skills / D. C. Fried, E. G. Wilkins, S. Peters, K. Schultz, B. Clark // Proc. of IJCAI., 2003. — P. 186–198.
36. Oliver L. The WITAS Multi-Modal Dialogue System I / L. Oliver, A. Bracy, A. Gruenstein, S. Peters // Proc. of EuroSpeech., 2001 — P. 62–71.
37. Pon-Barry H. Evaluating the Effectiveness of SCot: A Spoken Conversational Tutor / H. Pon-Barry, B. Clark, E. Bratt, K. Schultz, S. Peters // Proc. of the workshop: “Dialog-based Intelligent Tutoring Systems” held in conjunction with ITS 2004, the Seventh International Conference on ITS., 2004. — P. 23–32.
38. Alevin V. Towards Easier Creation of Tutorial Dialogue Systems: Integration and Authoring Environments for Tutoring and Dialogue Systems / V. Alevin, C. Rose // Proc. of the workshop: “Dialog-based Intelligent Tutoring Systems” held in conjunction with ITS 2004, the Seventh International Conference on ITS., 2004. — P. 1–7.
39. Wiener-Hastings P. The design and architecture of Research Methods Tutor, a second generation dialog-based tutor / P. Wiener-Hastings // Proc. of the workshop: “Dialog-based Intelligent Tutoring Systems” held in conjunction with ITS 2004, the Seventh International Conference on ITS., 2004. — P. 55–63.
40. Nakabayashi K., et al. Architecture of an Intelligent Tutoring Systems on the WWW / K. Nakabayashi et al. // Proc. of the 8th World Conference of the AIED Society, Kobe, Japan, 18–22 August 1997. — Kobe, Japan, 1997. — P. 214–221.
41. Brown J. S. SOPHIE: a sophisticated instructional environment for teaching electronic troubleshooting: BBN Report 2790 / J. S. Brown, R. R. Burton, A. G. Bell; Bolt Beranek and Newman Inc. — Cambridge, Massachusetts, 1974. — P. 14–21.
42. Brown J. S. SOPHIE: a step towards a reactive learning environment / J. S. Brown, R. R. Burton, A. G. Bell // Int Jnl Man-Machine Studies. — 1975. — Vol. 7. — P. 675–696
43. Brown J. S. Multiple representation of knowledge for tutorial reasoning / J. S. Brown, R. R. Burton // In Bobrow D. and Collins A. (Eds.) Representation and Understanding: Studies in Cognitive Science. — N. Y., Academic Press, 1975. — P. 64–72.
44. Brown J. S. Reactive learning environment for computer-assisted electronic instruction: BBN Report 3314 / J.S, Brown, R. Rubinstein, R. R. Burton; Bolt Beranek and Newman Inc. — Cambridge, Massachusetts, 1976. — 213 p.

45. Brown J. S. Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III / J. S. Brown, R. R. Burton, J. de Kleer // In Sleeman D. H. and Brown J. S. (Eds.) *Intelligent Tutoring Systems*. — London, Academic Press, 1982. — P. 264–271.
46. Burton R. R. *Semantically-centered parsing: Doctoral dissertation*; — University of California. Irvine, 1975. — 98 p.
47. Burton R. R. An investigation of computer coaching for informal learning activities / R. R. Burton, J. S. Brown // *Int. Jnl of Man-Machine Studies*. — 1979. — Vol. 11. — P. 5–24.
48. Yob G. Hunt the Wumpus / G. Yob // *Creative Computing* — 1975. September/October. — P. 51–54.
49. Russell J. S. *Artificial Intelligence. A Modern Approach* / J. S. Russel, P. Norvig — Prentice-Hall International, Inc. 1995. — 1080 p.
50. Stansfield J. C. Wumpus advisor I: a first implementation of a program that tutors logical and probabilistic reasoning skills / J. C. Stansfield, B. Carr, I. P. Goldstein. — Cambridge, Massachusetts, Massachusetts Institute of Technology, 1976. — 98 p. (AI Lab Memo 381.).
51. Carr B. Overlays: a theory of modeling for computer-aided instruction / B. Carr, I. P. Goldstein. — Cambridge, Massachusetts, Massachusetts Institute of Technology, 1977. — 115 p. (AI Lab Memo 406)
52. Goldstein I. P. The genetic graph: a representation for the evolution of procedural knowledge / I. P. Goldstein // *Int. Jnl Man-Machine Studies*. — 1979 — Vol. 11. — P. 51–57.
53. Brainerd C. J. *Piaget's Theory of Intelligence* / C. J. Brainerd. — Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1978. — 326 p.
54. Anderson J. R. *Language, memory, and thought* / J. R. Anderson. — Hillsdale, N. J.: Lawrence Erlbaum Associates, 1976. — 287 p.
55. Anderson J. R. *The Architecture of Cognition* / J. R. Anderson. — Cambridge, Massachusetts: Harvard University Press, 1983. — 311 p.
56. Anderson J. R. *Rules of the Mind* / J. R. Anderson. — Hillsdale, N. J.: Lawrence Erlbaum Associates, 1993. — 410 p.
57. Boyle C. F. Acquisition and automated instruction of geometry skills / C. F. Boyle, J. R. Anderson // *Annual Meeting of the American Educational Research Association*, New Orleans. — Wash., D. C., 1984. — 21 p.
58. Anderson J. R. The geometry tutor / J. R. Anderson, C. F. Boyle, G. Yost // *Proc. of the International Joint Conference on Artificial Intelligence*, Los Angeles. 1985. — P. 1–9.
59. Farrell R. G. An interactive computer-based tutor for LISP / R. G. Farrell, J. R. Anderson, B. J. Reiser // *Proc. of the National Conference on Artificial Intelligence*; Austin, Texas, 1984. — P. 106–109.
60. Reiser B. J. Dynamic student modeling in an intelligent tutor for LISP programming / B. J. Reiser, J. R. Anderson, R. G. Farrell // *Proc. of the Ninth International Joint Conference on Artificial Intelligence*; Los Angeles, 1985 — P. 8–14.

61. Corbett A. T. LISP Intelligent Tutoring System: Research in Skill Acquisition. Computer-Assisted Instruction and Intelligent Tutoring Systems. Shared Goals and Complementary Approaches / A. T. Corbett, J. R. Anderson // Jill H. Larkin and Ruth W. Chabay (Eds.). — N. J., Hillsdale, Lawrence Erlbaum Associates, 1992. — P. 48–55.
62. Anderson J. R. Cognitive Tutors: Lessons Learned / J. R. Anderson, A. T. Corbett, K. R. Koedinger. R. Pelletier // *The Journal of the Learning Sciences*. — 1995. Vol.4. — №2. — P. 167–207.
63. Baker R. S. Off-Task Behavior in the Cognitive Tutor Classroom: When Students “Game the System” / R. S. Baker, A. T. Corbett, A. Z. Wagner, K. R. Koedinger // *Proc. of the SIGCHI conference on human factors in computing systems*, 2004 — Vol.6. — №1. — P. 383–390.
64. Aleven V. Towards Tutoring Help Seeking: Applying Cognitive Modeling to Meta-Cognitive Skills / V. Aleven, B. McLaren, I. Roll, K. Koedinger // *Proc. of Intelligent Tutoring Systems Conference*, 2004 — P. 129–136.
65. Eysenck M. W. *Cognitive Psychology. A Student’s Handbook*. 3rd edition / M. W. Eysenck, M. T. Keane. — Lawrence Erlbaum Associates, 1995. — 542 p.
66. Солсо Р. Л. *Когнитивная психология* / Р. Л. Солсо. — М.: Триолта, 1996. — 598 с.
67. Kellogg R. T. *Cognitive Psychology* / R. T. Kellogg. — SAGE publications Ltd., 1995. — 523 p.
68. Neisser U. *Cognition and Reality. Principles and implications of cognitive psychology* / U. Neisser. — San Francisco: W. H. Freeman and Company. 1976. — 230 p.
69. Tolman E. C. Cognitive maps in rats and men / E. C. Tolman // *Psychological Review*. — 1975. — Vol.55. — P. 189–208.
70. Чмырь И. А. Учёт когнитивных свойств пассивного партнера в сценарно-ориентированных САИ-системах / И. А. Чмырь, А. И. Бондарь, М. Д. Жуковецкий // *Тепловые режимы и охлаждение радиоэлектронной аппаратуры (ТРИО)*. — 1995. — № 1–2. — С. 40–50.
71. Chimir I. New Visions of Old Models / I. Chimir, M. Horner // W. M. Beynon, C. L. Nehaniv, & K. Dautenhahn (Eds.), *Cognitive Technology: Instruments of Mind (Proc. 4th International Conference on Cognitive Technology, CT 2001 — University of Warwick, U. K.)*. Springer Lecture Notes in Artificial Intelligence, 2001. — Vol. 2117. — P. 157–163.
72. Чмырь И. А. Объектное моделирование ментальных процессов / И. А. Чмырь, А. П. Харьковская // *Зб. тез доповідей III Міжнародної наук.-практ. конф. студентів, аспірантів та молодих вчених «Системний аналіз та інформаційні технології»*. Київ, 1–3 липня 2001р. — К.: Ін-т прикладного системного аналізу при Національному технічному університеті України “Київський політехнічний інститут”, 2001. — С. 135–140.
73. Chimir I. Object modeling of cognitive processes / I. Chimir, A. Verlan, B. Zahidov, M. Sagatov // *Proc. of the First International Conference “Soft Computing and Computing with Words in System Analysis, Decision and Control”*. — Antalia (Turkey), 2001. — June 6–8. — P. 113–122.

74. Чмырь И. А. Применимость UML-ориентированных CASE-систем для программной симуляции ментальных процессов / И. А. Чмырь, А. П. Харьковская // Искусственный интеллект. — 2001. — №4. — С. 41–52.
75. Чмырь И. А. Когнитивные основы и концептуальный базис диалогового процесса / И. А. Чмырь, А. Ф. Верлань, М. Ф. Ус // УСиМ. — 2002. — № 6. — С. 54–60.
76. Chimir I. Psychological Foundations and Conceptual Basis of Dialogue Processes / I. Chimir, A. Verlan, M. Us, M. Sagatov // Yusupbekov N. R. and Aliev R. A. (eds.) proceedings of WCIS-2002. Second World Conference On Intelligent Systems For Industrial Automation. Tashkent, 4–5 June 2002. — Tashkent, Uzbekistan, 2002. — P. 356–363.
77. Верлань А. Ф. Объектно-ориентированное моделирование когнитивных процессов / А. Ф. Верлань, И. А. Чмырь // Электронное моделирование. — 2002. — Т.24. — №4. — С. 53–64.
78. Чмырь И. А. Когнитивные аспекты моделирования диалоговых агентов / И. А. Чмырь, М. Ф. Ус, Муса Раид Аль-Кавасми // Вісн. Черкаськ. технолог. Ун-ту. — 2002. — №4. — С. 7–12.
79. Чмырь И. А. Объектная модель цикла восприятия / И. А. Чмырь, Муса Раид Аль-Кавасми // Тези ХХІІ наук.-техн. конф. «Моделювання». Київ, 9–10 січня 2003р. — К.: Ін-т проблем моделювання в енергетиці ім. Г. Е. Пухова НАН України, 2003. — С. 11–12.
80. Чмырь И. А. Цикл перцепции Нейсера: формальная модель и её практическое применение / И. А. Чмырь, Муса Раид Аль-Кавасми // Тези міжнар. наук. конф. «Управління активними системами в бізнесі, освіті, техніці». — Черкаси: Черкаська Академія менеджменту, 2003. — С. 113.
81. Чмырь И. А. Цикл перцепции Нейсера: формальное представление и практическое применение / И. А. Чмырь, Муса Раид Аль-Кавасми // Искусственный интеллект. — 2003. — №1. — С. 107–116.
82. Чмырь И. А. Объектные модели фильтрации внимания: классификация и интеграция / И. А. Чмырь, И. А. Жирякова, Муса Раид Аль-Кавасми // Искусственный интеллект. — 2003. — №2. — С. 52–63.
83. Чмырь И. А. Объектное моделирование феномена внимания / И. А. Чмырь, Муса Раид Аль-Кавасми // Материалы международной научно-технической конференции. «Интеллектуальные и многопроцессорные системы» (ИМС'2003). — Дивноморское: Таганрогский радиотех. ун-т, Россия, 2003. — С. 148–150.
84. Chimir I. New visions of old models / I. Chimir, M. Horney // Проблеми бізнесу: Зб. наук. праць. — Черкаси: Черкаська академія менеджменту, 2003. — Вип. 7. — С. 93–97.
85. Чмырь И. А. Когнитивные модели для диалоговых агентов / И. А. Чмырь, М. Ф. Ус, З. М. Гадецкая, С. Ф. Акопф // Вісн. Академії дистанційної освіти. — 2004. — №2. С. — 41–46.
86. Chimir I. Object Modeling of Filter-Oriented Systems of Attention: Possibilities of Integration / I. Chimir, W. Abu-Dawwas, R. Alqawasmi // The International Arab Journal of Information Technology. — 2005. — Vol. 2, № 1. — P. 67–74.

87. Chimir I. A. Object-Oriented Modeling of Attentional Systems / I. A. Chimir, W. A. Abu-Dawwas, M. A. Horney // Proc. of the 2nd International Refereed Conference on Information Technology “Human Computer Interaction” (ICIT 2005). Amman, 3–5 May 2005. — Amman: Al-Zaytoonah University of Jordan, 2005. — P. 129–135.
88. Чмырь И. А. Программная реализация модели сфокусированного внимания Бродбента / И. А. Чмырь, И. А. Жирякова // Искусственный интеллект. — 2005. — №1. — С. 126–137.
89. Chimir I. A. Neisser’s Cycle of Perception: Formal Representation and Practical Implementation / I. A. Chimir, W. A. Abu-Dawwas, M. A. Horney // Journal of Computer Science. — 2005 — Special issue on Intelligence, Communication and Computer Interaction. — P. 106–111.
90. Верлань А. Ф. Компьютерное моделирование сфокусированного слухового внимания / А. Ф. Верлань, И. А. Чмырь, И. А. Жирякова // Электронное моделирование. — 2006. — Т.28, №1. — С. 19–31.
91. Жирякова И. А. Объектно-ориентированная модель селективного зрительного внимания / И. А. Жирякова, И. А. Чмырь // Искусственный интеллект. — 2006. — №1. — С. 12–21.
92. Чмырь И. А. Объектное моделирование ресурсной теории внимания / И. А. Чмырь, И. А. Жирякова // Системні дослідження та інформаційні технології. — 2006. — №1. — С. 31–40.
93. Платон. Сочинения: в 4 т.; под общ. ред. А. Ф.Лосева и В. Ф. Асмуса; пер. с древнегреч. — СПб.: Изд-во С.-Петербур. ун-та; «Изд-во Олега Абышко». СПб., 2006–2007.
94. Schaffer, J. Knowing the answer / J. Schaffer. // Philosophy and Phenomenological Research. — 2007. — Vol. LXXV No. 2, September 2007. — P. 383–403.
95. Kahneman, D. Attention and Effort / D. Kahneman. — Prentice-Hall, Inc., Englewood Cliffs, New Jersey. 1973. — 246 p.
96. Белнап Н. Логика вопросов и ответов / Н. Белнап, Т. Стил. — М.: Прогресс. 1981. — 288 с.
97. Chemyr I. A. SmartText: Using Agent Supporting Personalized Reading Comprehension. / I. A. Chimir, M. A. Horney, L. Andersen-Inman. // Personal Technologies. 1998. — Vol.2. — №3. — P. 152–161.
98. Chimir I. Modeling of a Procedural Knowledge by a Dialogue Knowledge Base / I. Chimir, W. Abu-Dawwas // The International Arab Journal of Information Technology. — 2004. — Vol. 1. — № 1. — P. 128–134
99. Uyemov, A. The Ternary Description Language as a Formalism for the Parametric General Systems Theory. Part I / A. Uyemov // Int. J. of General Systems. — 1999. — Vol. 28(4–5). — P. 351–366.
100. Uyemov, A. The Ternary Description Language as a Formalism for the Parametric General Systems Theory. Part II / A. Uyemov // Int. J. of General Systems. — 1999. — Vol. 31(2). — P. 131–155.

101. Uyemov, A. The Ternary Description Language as a Formalism for the Parametric General Systems Theory. Part III / A. Uyemov // *Int. J. of General Systems*. — 2003. — Vol. 32(6). — P. 583–623.
102. Чмырь И. А. Естественный диалог: моделирование диалоговой транзакции в контексте представления знаний / И. А. Чмырь // *Интеллектуальные системы*. — 2012. — Т.16. — Вып. 1–4. — С. 73–102.
103. Igor Chimir. Modeling Human Dialogue Transactions in the Context of Declarative Knowledge / Igor Chimir, Ameer Ghazi, Waheeb Abu-Dawwas // *The International Arab Journal of Information Technology*. — May 2013. — Vol. 10. — № 3. — P. 305–315.
104. Брауэр В. Введение в теорию конечных автоматов / В. Брауэр. — М.: Радио и связь, 1987. — 219 с.
105. Ричмонд У. К. Учителя и машины. Введение в теорию и практику программированного обучения / У. К. Ричмонд. — М.: Мир, 1968. — С. 57–64.
106. Волож Б. Б. ДИМО — гибкие сценарии диалога и их конструирование. / Б. Б. Волож // *Пакеты прикладных программ. Инструментальные системы*. — М.: Наука, 1987. — С. 18–29.
107. Чмырь И. А. Сценарий диалога и его структурные элементы / И. А. Чмырь, П. С. Попов // *Вопр. радиоэлектроники. Сер. ТРТО*. — 1991. — Вып. 3. — С. 87–95.
108. Чмырь И. А. Моделирование диалогового поведения средствами объектной базы данных / И. А. Чмырь, М. А. Клименко // *Вісн. Інженерної Академії України*. — 1997. — №2. — С. 51–52.
109. Чмырь И. А. Объектная модель для диалогового решателя задач / И. А. Чмырь, М. А. Клименко // *Зб. наук. праць Ін-ту проблем моделювання в енергетиці*. — К., 1998. — Вип. 4. — С. 152–158.
110. Чмырь И. А. Использование концепции объектно-ориентированных баз данных для проектирования системы поддержки диалоговых приложений / И. А. Чмырь, М. А. Клименко // *Вісн. Інженерної Академії України*. — 1998. — Спец. випуск. — С. 23–26.
111. Цикридис Д. Модели данных / Д. Цикридис, Ф. Лоховски. — М.: Финансы и статистика, 1985. — 254 с.
112. Питерсон Дж. Теория сетей Петри и моделирования систем / Дж. Питерсон. — М.: Мир, 1984. — 264 с.
113. Мурата Т. Сети Петри: свойства, анализ, приложения / Т. Мурата // *ТИИЭР*. — 1989. — Т.77. — №4. — С. 41–85.
114. Котов В. Е. Сети Петри. / В. Е. Котов. — М.: Наука, 1984. — 164 с.
115. Айтьян С. Х. Инструментальные средства разработки экспертных систем на ПЭВМ / С. Х. Айтьян, К. Р. Гуарян // *Микропроцессорные средства и системы*. — 1989. — №6. — С. 55–61.
116. Фурман А. В. Психодіагностика інтелекту в системі диференціації навчання / А. В. Фурман. — К.: Освіта, 1993. — 107 с.
117. Chmyr Igor. Dialogue of Partners as a Method For Non-Formal Problem Solving / Igor Chmyr // Maddy D. Brouwer-Janse and Thomas L. Harrington (Eds.) collection:

- Human-Machine Communication for Educational Systems Design, NATO ASI Series F129 Springer-Verlag, Berlin, 1994. — P. 221–228.
118. Мейер Д. Теория реляционных баз данных / Д. Мейер. — М.: Мир, 1987. — 331 с.
 119. Аверкин А. Нечёткие множества в моделях управления и искусственного интеллекта / А. Аверкин, И. Батыршин и др. — М.: Наука, 1986. — 219 с.
 120. Jang J.-S. R.-Fuzzy and Soft Computing / J.-S. R. Jang, Ch.-T. Sun, E. Mizutani. — N. J.: Prentice Hall, 1997–603 p.
 121. Чмырь И. А. Нечёткая автоматная модель диалогового поведения / И. А. Чмырь, А. И. Чмырь, Абу-Даввас // Тепловые режимы и охлаждение радиоэлектронной аппаратуры (ТРИО). — 1998. — № 1. — С. 84–90.
 122. Чмырь И. А. Применение нечёткой логики для определения когнитивного типа пользователя в интеллектуальных обучающих системах / И. А. Чмырь, М. Ф. Ус, А. В. Пискун // Моделювання та інформаційні технології: зб. наук. праць. — К.: Ін-т проблем моделювання в енергетиці НАН України, 2002. — Вип. 14. — С. 98–104.
 123. Martin J. Application Development Without Programmers / J. Martin. — Savant Institute, 1981. — 385 p.
 124. Embley D. W. NFQL: The Natural Forms Query Language / D. W. Embley // ACM Trans. on Database Systems. — 1989. — Vol.14. — P. 24–32.
 125. Santos A. Telephone Information System: Dialogue Specification Language / A. Santos, J. Colas, J. Lestani // Maddy D. Brouwer-Janse and Thomas L. Harrington (Eds.) Human-Machine Communication for Educational Systems Design, NATO ASI Series F129 Springer-Verlag, Berlin, 1994. — P. 315–324.
 126. Берт П. Интеллектуальное восприятие в пирамидальной зрительной машине / П. Берт // ТИИЭР. — 1988. — Т. 76. — №8, — С. 175–186.
 127. Залманзон Л. А. Преобразование Фурье, Уолша, Хаара и их применение в управлении, связи и других областях / Л. А. Залманзон. — М.: Наука, 1989. — 494 с.
 128. Харитоненко И. О. Организация средств визуализации изображений в компьютерных системах поддержки диалога: Диссертация кандидата технических наук: специальность 05.13.05 — Элементы и устройства вычислительной техники и систем управления. Защищена в Одесском политехническом университете. — Одесса, 1993. — С. 51–99.
 129. Чмырь И. А. “Процессор Диалога” для генерации обучающих курсов / И. А. Чмырь, П. С. Попов, М. С. Харьковская // Доклады междунар. симпозиума ИНФО’89. — Минск, 1989. — Т.1. — Ч.2. — С. 957–960.
 130. Чмырь И. А. Двухмашинная интеллектуальная САД-система / И. А. Чмырь, Ю. Н. Крапивный, О. П. Пилипенко // Тепловые режимы и охлаждение радиоэлектронной аппаратуры (ТРИО). — 1994. — № 1–2. — С. 17–24.
 131. Чмир І. О. Комп’ютерний тьютор української мови: теоретичні підвалини та досвід роботи / І. О. Чмир, О. І. Бондар, О. І. Чмир, В. А. Абу-Даввас // Наукове пізнання. Методологія та технологія. — Одеса: Південноукраїнський педагогічний ун-т ім. К. Д.Ушинського, 1998. — Вип. 1–2. — С. 84–90.

132. Ziv J. A Universal Algorithm for Sequential Data Compression / J. Ziv, A. Lempel // IEEE Trans. Info. Theory. — 1977. — Vol. IT — 23. — N3. — P. 337–343.
133. Fiala R. Data Compression with Finite Windows / R. Fiala, D. Green // Communication of the ASM. — 1989. — Vol. 32. — N4. — P. 490–505.
134. Харитоненко И. О. Метод адаптивной компрессии цветных изображений / И. О. Харитоненко // Вопр. радиоэлектроники. Сер. ТРТО. — 1992. — Вып. 3. — С. 102–107.
135. Newell A. Unified Theories of Cognition / A. Newell, Harvard University Press, 1994. — 395 p.
136. Чмырь И. А. Логические возможности построения системы тестирования, совмещенной с обучением / И. А. Чмырь, В. А. Абу-Даввас // Вимірювальна та обчислювальна техніка в технологічних процесах. — 1998. — № 2. — С. 104–110
137. Чмырь И. А. Моделирование процедурных знаний эксперта средствами диалоговой базы знаний / И. А. Чмырь, Аюкоф Хадж Сафуан, А. И. Чмырь // Вісн. Черкаськ. технолог. ун-ту. — 2003. — №2. — С. 89–95.
138. Чмырь И. А. Декларативные и процедурные знания в диалоговом процессе / И. А. Чмырь, Аюкоф Хадж Сафуан // Тези доп. учасників VI міжнар. наук.-практ. конф. студентів, аспірантів та молодих вчених «Системний аналіз та інформаційні технології». Київ, 1–3 липня 2004р. — К.: Національний технічний ун-т України, «Київський політехнічний інститут» Інститут прикладного системного аналізу, 2004. — С. 232–234.
139. Чмырь И. А. Диалоговый и дедуктивный процессы: возможность совмещения / И. А. Чмырь, Аюкоф Хадж Сафуан // Искусственный интеллект. — 2004. — №1. — С. 134–142.
140. Чмырь И. А. Способ совмещения диалогового и дедуктивного процессов / И. А. Чмырь, Аюкоф Хадж Сафуан // Тези міжнар. наук. конф. «Управління активними системами в бізнесі, освіті, техніці». — Черкаси: Східноєвропейський ун-т економіки і менеджменту, 2004. — С. 81–82.
141. Chimir I. A. Deductive Inference in The Context of The Dialogue Process / I. A. Chimir, W. A. Abu-Dawwas // Asian Journal of Information Technology. — 2006. — Vol. 5. — №3. — P. 263–271.
142. Янко Т. е. Информационная модель диалога / Т. е. Янко // Научно-техническая информация. Серия 2. Информационные процессы и системы. — 1990. — №12. — С. 30.
143. Дракин В. И. Общение конечных пользователей с системами обработки данных / В. И. Дракин, Э. В. Попов, А. Б. Преображенский. — М.: Радио и связь, 1988. — 286 с.
144. Петрушенко А. Н. О диалоговых вычислениях в алгоритмических алгебрах / А. Н. Петрушенко // Кибернетика. — 1990. — №1. — С. 13–20.
145. Констейбл Р. О классах схем программ / Р. Констейбл, Д. Грис // Кибернетический сборник. — 1977. — Вып. 14. — С. 122–177.
146. Кузнецов В. Е. Представление в ЭВМ неформальных процедур / В. Е. Кузнецов. — М.: Наука, 1989. — 202 с.

147. Harmon P. Expert Systems. Artificial Intelligence in Business / P. Harmon, D. King. — John Wiley, 1985. — 218 p.
148. Представление и использование знаний / Под ред. Х. Уэно и М. Исидзука. — М.: Мир, 1985. — 153 с.
149. Кондаков Н. И. Логический словарь-справочник / Н. И. Кондаков. — М.: Наука, 1976. — 586 с.
150. Winston P. H. Artificial Intelligence / P. H. Winston. — Addison-Wesley, 1984. — 359 p.
151. Попов Э. В. Экспертные системы / Э. В. Попов. — М.: Наука, 1987. 211 с.
152. Levin R. I. A Comprehensive Guide to AI and Expert Systems: Turbo Pascal Edition / R. I. Levin, D. E. Drang, B. Edelson. — McGraw-Hill, 1988. — 182 p.
153. Шеннон К. Символический анализ релейных и переключательных схем. Работы по теории информации и кибернетике / К. Шеннон. — М.: Изд. ин. лит., 1963. — 829 с.
154. Schneider W. Controlled and automatic human information processing: Detection, search, and attention / W. Schneider, R. M. Shiffrin // *Psychological Review*. — 1977. — № 92. — P. 424–428.
155. Shiffrin R. M. Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory / R. M. Shiffrin, W. Schneider // *Psychological Review*. — 1977. — № 84. — P. 127–190.
156. Cheng P. W. Pragmatic reasoning schemas / P. W. Cheng, K. J. Holyok // *Cognitive Psychology*. — 1985. — № 17. — P. 391–416.
157. Logan G. D. Toward an instance theory of automatization / G. D. Logan // *Psychological Review*. — 1988. — № 95. — P. 492–527.
158. Чмырь И. А. Экспертные системы на базе сетевой диалоговой модели / И. А. Чмырь, П. С. Попов // *Вопр. радиоэлектроники. Серия ТРТО*. — 1992. Вып. 3–4. — С. 42–51.
159. Франклин Дж. Э. Технология экспертных систем для военных применений: избранные примеры / Дж. Э. Франклин, К. Л. Кармоди, К. Келлер, Т. С. Левит, Б. Л. Бюто // *ТИИЭР*. — 1988. — Т. 76. — №10. — С. 18–69.
160. Чмырь И. А. Модели вопросно-ответных и диалоговых систем в искусственном интеллекте / И. А. Чмырь, Назир Двай Гайс // *Матеріали п'ятої міжнар. наук.-практ. конф. «Нові інформаційні технології в навчальному процесі»*. Одеса, 29 червня — 2 липня 2005р. — Одеса: Південноукраїнський державний педагогічний ун-т імені К. Д. Ушинського, 2005. — С. 114–117.
161. Чмырь И. А. Вопросно-ответные отношения в контексте логического вывода / И. А. Чмырь, Назир Двай Гайс, Аюпф Хадж Сафуан // *Искусственный интеллект*. — 2005. — №4. — С. 535–545.
162. Осука С. Обработка знаний / С. Осука. — М.: Мир, 1989. — 149 с.
163. Нильсон Н. Принципы искусственного интеллекта / Н. Нильсон. — М.: Радио и связь, 1985. — 252 с.
164. Newell A. Remarks on the relationship between artificial intelligence and cognitive psychology / A. Newell // R. Banerji and J. D. Mesarovich (eds.) *Theoretical*

- Approaches to Non-Numerical Problem Solving. — N. Y.: Springer-Verlag, 1970. — P. 363–399.
165. Абу-Даввас Вахееб Абдел-Вахаб. Моделирование диалогового процесса при разработке персонализированных компьютерных обучающих систем. Автореф. дис. на соискание учёной степени кандидата технических наук / Институт проблем моделирования в энергетике НАН Украины. — К., 1999. — 20 с.
166. Чмырь И. А. Учёт когнитивных особенностей обучаемого при проектировании персонализированных обучающих программ / И. А. Чмырь Ямен Хаджо // Матер. междунар. науч.-техн. конференции «Искусственный интеллект, интеллектуальные и многопроцессорные системы». Таганрог, 20–25 сент. 2004г. — Таганрог: Таганрог. радиотехн. ун-т, 2004. — Т.1. — С. 379–381.
167. Чмырь И. А. Многоагентная система моделирования диалогового поведения / И. А. Чмырь, А. Д. Бодарев // Моделювання соціально-економічних процесів в Україні. — К.: Державний науково-дослідний інститут інформатизації та моделювання економіки, 1997. — С. 63–69.
168. Чмырь И. А. Многоагентная система моделирования диалогового поведения / И. А. Чмырь, В. А. Абу-Даввас // Придніпровський науковий вісник. Технічні та сільськогосподарські науки. — 1998. — № 66 (133). — С. 27–36.
169. Chemyr I. Systems with embedded intelligence based on dialogue engine architecture / I. Chemyr, A. Verlan // Proc. of the Sixth International Conference “Advanced Computer Systems” 18 -19 November 1999. — Szczecin (Poland), 1999. — P. 164–169.
170. Верлань А. Ф. Системы со встроенным интеллектом на базе архитектуры машины диалога. А. Ф. Верлань, И. А. Чмырь // Электронное моделирование. — 2001. — Т.23. — №1. — С. 75–83.
171. Коноплев И. Д. Система диалогового формирования сеточной модели / И. Д. Коноплев, И. А. Чмырь, Е. М. Маньковская // Тезисы докладов научн.-техн. семинара «Математическое моделирование физических полей». Саратов, 22–23 сент. 1988г. — Саратов: НИИ механики и физики Саратов. ун-та, 1988. — С. 16–17.
172. Чмырь И. А. «Процессор диалога» для разработки приложений в области САПР // Тезисы докладов третьего научн.-техн. семинара «САПР и АСУТП в химической промышленности». — Черкассы: Черкасский филиал КПИ, 1989. — С. 6–7.
173. Математическое моделирование теплового состояния РЭА. Ч. 2. Разработка диалоговой системы автоматизированного формирования сеточной модели: Отчет по НИР / Одесский институт низкотемпературной техники и энергетики; N гос. регистрации 01880081870. — Одеса, 1990. — 130 с.
174. Чмырь И. А. Принципы организации интеллектуальной САД-системы / И. А. Чмырь, И. Д. Коноплев // Тепловые режимы и охлаждение радиоэлектронной аппаратуры (ТРИО). — 1993. — № 3–4. — С. 18–25.
175. Чмырь И. А. Разработать архитектуру системы диалогового автоматизированного проектирования водоочистных установок: Отчет по НИР. / И. А. Чмырь,

- Л. С. Остапенко. — Одесса: Одесский институт низкотемпературной техники и энергетики. — 1993. — 156 с.
176. Чмырь И. А. Разработать пакет программ диалогового проектирования принципиальных схем соединений: Отчет по НИР / И. А. Чмырь, Ю. Н. Крапивный. — Одесса: Одесский институт низкотемпературной техники и энергетики, 1994. — 144 с.
177. Чмырь И. А. Разработка методики непроцедурного диалогового взаимодействия с базой данных АСУ: Отчет по НИР / И. А. Чмырь, В. А. Трухнин. — Одесса: Одесский политехнический институт; № гос. регистрации 01870071480, 1987. — 161 с.
178. Чмырь И. А. Пакет программ на ПАСКАЛЕ, обеспечивающий непроцедурное общение с базой данных / И. А. Чмырь, П. С. Попов // Тезисы докл. Всесоюзной науч. конф. «Компьютеризация информационных процессов в управлении народным хозяйством». Москва, 3–5 окт. 1988 г. — М.: Экономико-статистический институт, 1988. — Ч. 1. — С. 129.
179. Чмырь И. А. Пакет программ для непроцедурного общения с базами данных / И. А. Чмырь, И. В. Португал, И. В. Соловьева // Тезисы докл. Всесоюзного семинара «Модели планирования и хозрасчета на предприятии». Москва, 13–15 сент. 1988г. — М.: Центральный экономико-математический институт АН СССР, 1988. — С. 144–145.
180. Чмырь И. А. Разработка системы непроцедурного диалогового взаимодействия с базой данных: Отчет по НИР / И. А. Чмырь, Ю. О. Руднева. — Одесса: Одесский технологический институт холодильной промышленности, № гос. Регистрации 01880019157, 1988. — 202 с.
181. Чмырь И. А. Логические основы интеррогативного взаимодействия с информационной базой / И. А. Чмырь, И. В. Сараева // Тезисы докл. и сообщений к 7 Всесоюзной конференции «Проблемы интеллектуального развития организационных систем». Новосибирск, 1–3 окт. 1991г. — Новосибирск, 1991. — С. 105–108.
182. Ус М. Ф. Организация системы программируемого диалога для общения с базой данных. Автореф. дис. на соискание учёной степени кандидата технических наук / Институт проблем моделирования в энергетике НАН Украины. — К., 1992. — 21 с.
183. Чмырь И. А. Логические основы построения сценариев экспертных систем управлением производством / И. А. Чмырь, М. Ф. Ус // Тезисы докл. Всесоюзной конф. «Организация управления производством в новых условиях хозяйствования». — Одесса: Одесский политехнический ин-т, 1990. — С. 91–93.
184. Pittman G. Teaching Structural English / G. Pittman. — Brisbane: Jacaranda, 1963. — 281 p.
185. Davies P. Situational Lesson Plans / P. Davis, J. Roberts, R. Rosner. — Mexico City: Macmillan, 1975. — 130 p.
186. Brooks N. Language and Language Learning: Theory and Practice 2nd ed. / N. Brooks — N. Y.: Harcourt Brace, 1964. — 213 p.

187. Asher J. Learning Another Language Through Action: The Complete Teacher's Guide Book — Los Gatos, Calif.: Sky Oaks Productions, 1977. — 98 p.
188. Francoise G. Developing Reading Skills: A practical guide to reading comprehension exercises / G. Francoise. — Cambridge University Press, 1992. — 201 p.
189. Watson B. Mini-Effort, Maxi>Returns / B. Watson. — Odessa: The British Council, English Teaching Resource Center, 1996. — 56 p.
190. Мельников В. М. Введение в экспериментальную психологию личности / В. М. Мельников, Л. Т. Ямпольский. — М.: Просвещение, 1985. — 319 с.
191. Общая психодиагностика. Основы психодиагностики, немедицинской психотерапии и психологического консультирования / под ред. А. А. Бодалева, В. В. Столина. — М.: Изд-во МГУ, 1987. — 304 с.
192. Пирьев Г. Д. Классификация методов в психологии / Г. Д. Пирьев. — Братислава: Психодиагностика в социалистических странах, 1985. — С. 19–25.
193. Бурлачук Л. Ф. Словарь-справочник по психологической диагностике / Л. Ф. Бурлачук, С. М. Морозов. — К.: Наук. думка, 1989. — 200 с.
194. Бобров А. Е. Алкогольный скрининг-тест: его валидность и структура / А. Е. Бобров, А. И. Шурыгин // Психологическая диагностика при нервно-психических и психосоматических заболеваниях. — Л.: НИИ психоневрологии им. В. М. Бехтерева, 1985. — С. 33–36.
195. Шмелев А. Г. Анализ пунктов при конструировании и применении тест-опросников: ручные и компьютерные алгоритмы / А. Г. Шмелев, В. И. Похилько // Вопр. психологии. — 1985. — № 4. — С. 126–134.
196. Березин Ф. Б. Методика многостороннего исследования личности / Ф. Б. Березин, М. П. Мирошников, Р. В. Рожанец. — М.: Медицина, 1976. — 99 с.
197. Собчик Л. Н. Пособие по применению ММРІ Л. Н. Собчик. — М.: МЗ РСФСР, 1971. 86 с.
198. Ханин Ю. Л. Шкала Марлоу-Крауна для исследования мотивации одобрения / Ю. Л. Ханин. — Л.: НИИ физической культуры, 1976. — 10 с.
199. Wechsler D. The Measurement and Apraisal of Adult Intelligence / D. Wechsler. — Baltimore: Williams and Wilkins, 1958. — 297 p.
200. Румянцев Г. Г. Опыт применения метода незаконченных предложений в психиатрической практике / Г. Г. Румянцев // Исследование личности в клинике и в экспериментальных условиях. — Л.: НИИ психоневрологии им. В. М. Бехтерева, 1969. — С. 266–275.
201. Беспалько И. Г. Проективные методы / И. Г. Беспалько, И. Н. Гильяшева // Методы психологической диагностики и коррекции в клинике. — Л.: Медицина, 1978. — С. 116–144.
202. Норакидзе В. Г. Методы исследования характера личности В. Г. Норакидзе. — Тбилиси: Мецниереба, 1975. — 243 с.
203. Патрик Э. Основы теории распознавания образов / Э. Патрик. — М.: Сов. радио, 1970. — 408 с.
204. Фу К. Последовательные методы в распознавании образов и обучении машин / К. Фу. — М.: Наука, 1971. — 255 с.

205. Загоруйко Н. Г. Методы распознавания образов и их применение / Н. Г. Загоруйко. — М.: Сов. радио, 1972. — 206 с.
206. Вапник В. Н. Теория распознавания образов (статистические проблемы обучения) / В. Н. Вапник, А. Я. Черновенкис. — М.: Наука, 1974. — 415 с.
207. Дуда Р. Распознавание образов и анализ сцен / Р. Дуда, П. Харт. — М.: Мир, 1976. — 511 с.
208. Фукунага К. Введение в статистическую теорию распознавания образов / К. Фукунага. — М.: Наука, 1979. — 368 с.
209. Горелик А. Л. Методы распознавания / А. Л. Горелик, В. А. Скрипник. — М.: Высшая шк., 1989. — 232 с.
210. Crane, G. R. (ed.) The Perseus Project, <http://www.perseus.tufts.edu>, September, 1997.
211. Britt A. M. Learning from history texts: From causal analysis to argument models / A. M. Britt, J. F. Rouet, M. C. Georgi, G. Perfetti // G. Leinhart, I. L. Beck, C. Stainton, Eds. Teaching and Learning in History. — N. J. Hillsdale: Lawrence Erlbaum Associates, 1994. — P. 47–84.
212. Anderson-Inman L. Transforming text for at-risk readers / L. Anderson-Inman, M. Horney // D. Reinking, L. Labbo, M. McKenna & R. Kieffer (Eds.). Handbook of literacy and technology: Transformations in a post-typographic world Mahwah. — N. J.: Lawrence Erlbaum Associates, 1998. — P. 15–43.
213. Anderson-Inman L. Hypertext Literacy: Observations From the ElectroText Project / L. Anderson-Inman, M. A. Horney, D. T. Chen, L. Lewin // Language Arts. — 1994. — Vol. 71. — №4. — P. 279–287.
214. Horney M. A. The ElectroText Project: Hypertext Reading Patterns of Middle School Students / M. A. Horney, L. Anderson-Inman // Journal of Educational Multimedia and Hypermedia. — 1994. — Vol. 3. — № 1. — P. 71–91.
215. Hawley M. R. Things That Think / M. R. Hawley, D. Poor, M. Tuteja. // Personal Technologies. — 1997. — Vol. 1. — №1. — P. 13–20.
216. Hayes P. Changes in human-computer interfaces on the space station: why it needs to happen and how to plan for it. / P. Hayes // Humans in Automated and Robotic Space Systems. — Wash., D. C. National Academy of Sciences, 1985. — P. 34–48.
217. Minsky M. The Society of Mind / M. Minsky. — Simon and Schuster, 1988. — 303 p.
218. Математика и САПР. — М.: Мир, 1988. — Кн. 1. — 190 с.
219. Математика и САПР. — М.: Мир, 1989. — Кн. 2. — 215 с.
220. Коноплёв И. Д. Применение диалоговых методов при моделировании явлений теплопереноса / И. Д. Коноплёв, Л. С. Остапенко, И. В. Соловьева // Тепловые режимы, термостатирование и охлаждение радиоэлектронной аппаратуры (ТРТО). — 1992. — Вып. 3–4. — С. 93–101.
221. Коздоба Л. А. Электрическое моделирование явлений тепло- и массопереноса / Л. А. Коздоба. — М.: Энергия, 1972. — 98 с.
222. AUTOCAD, версия 10. Руководство пользователя / пер. с англ. London: Autodesk Limited, 1989. — 331 с.

223. Дешко А. И. Непроцедурные языки манипулирования данными. Перспективы развития и классификация / А. И. Дешко, Б. В. Игнатенко, В. И. Павловский // Программирование. — 1985. — №4. — С. 18–31.
224. Грэй П. Логика, алгебра и базы данных / П. Грэй. — М.: Машиностроение. 1989. — 240 с.
225. Дейт К. Руководство по реляционной СУБД В2 / К. Дейт. — М.: Финансы и статистика, 1988. — 271 с.
226. Файнберг В. Базы данных типа «клиент-сервер» / В. Файнберг // Компьютер-Пресс. — 1990 — №7. — С. 24–34.
227. Zloof M. M. QBE/OBE; a Language for Office and Business Automation / M. M. Zloof // Computer. — 1981. — Vol.14. — №5. — P. 81–94.
228. Embley D. W. NFQL: The Natural Forms Query Language / D. W. Embley // ACM Transactions on Database Systems. — 1989. — Vol.14. — №2. — P. 45–54.
229. Clifton L. Barbary Database Primer on Natural Language / L. Clifton // Journal of System Management. — 1987. — №2. — P. 15–25.
230. Чмырь И. А. Пакет программ для непроцедурного общения с базами данных / И. А. Чмырь, И. В. Португал, И. В. Соловьева // Тезисы докл. всесоюзн. семинара «Модели планирования и хозрасчета на предприятии». Москва, 13–15 сент. 1988г. — М.: Центральный экономико-математический институт АН СССР, 1988. — С. 144–145.
231. Чмырь И. А. «Процессор диалога» — инструментальная система для разработки диалоговых приложений / И. А. Чмырь, М. Ф. Ус, И. О. Харитоненко // Тез. докл. междун. конф.-ярмарки «Технология программирования 90-х». Киев, 14–17 мая 1991г. — К.: Ин-т кибернетики им. В. М. Глушкова, 1991. — С. 57–58.
232. Чмырь И. А. Проблемы разработки высокопроизводительных программно-аппаратных средств компрессии/декомпрессии изображений / И. А. Чмырь, И. О. Харитоненко // Тез. докл. семинара «Инструментальные средства программирования». Одесса, 12–16 окт. 1992г. — Одесса: Ин-т кибернетики им. В. М. Глушкова, 1992. — С. 83–84.
233. Чмырь И. А. Конвертация интерпретирующих программ из среды MS DOS в среду WINDOWS / И. А. Чмырь, О. В. Хороших // Материалы 60-й учебно-метод. и научно-техн. конф. проф.-препод. состава, научных сотр. и студентов Одесской государственной академии холода «Теория и практика вузовской науки». — Одесса: Одесская государственная академия холода, 1995. — С. 16–17.

ПРИЛОЖЕНИЕ

Настоящее приложение содержит краткое описание диалоговой программы English Grammar in Use, которая была разработана в лаборатории Диалоговых систем Одесского института низкотемпературной техники и энергетики в начале 90-х годов. Программа разрабатывалась в рамках серии проектов, имеющих лингводидактическую направленность. Целью этой серии проектов было внедрение результатов теоретического моделирования эротетического диалога, а также проверка применимости диалоговых процессов для решения плохо формализуемой задачи передачи знаний в ходе обучения тьюторского типа.

В процессе работы над проектами осуществлялось тестирование инструментального пакета программ Процессор диалога. Серия пакетов программ под общим наименованием Процессор диалога является реализацией идеи проблемно-независимого диалогового решателя задач, описанной в седьмом разделе. В процессе работы над проектами также проверялись основные положения теории обучающего стимула и способность диалоговой базы знаний сохранять специфическую методику обучения того или иного преподавателя.

Одной из причин, по которой описание диалоговой программы English Grammar in Use включено в качестве приложения к книге, является желание проиллюстрировать эффективность развиваемого диалогового подхода к моделированию и проектированию диалоговых программ. При работе над программой English Grammar in Use не было написано ни одной строчки программного кода. Все особенности, как обучающих стимулов, так и методики обучения реализуются исключительно средствами диалоговой базы знаний.

Программа English Grammar in Use является электронной версией известного британского учебника *“English Grammar in Use. A self-study reference and practice book for intermediate students”*, разработанного Раймондом Мэрфи и впервые изданного в Cambridge University Press в 1988 году. Выбор отмеченного учебника в качестве прототипа для программы был обусловлен тем фактом, что его материал хорошо структурирован и поэтому относительно легко разбивается на обучающие эпизоды.

Основное внимание при разработке программы “English Grammar in Use” уделялось проектированию обучающих эпизодов-сцен. Обучающие сцены представляют собой последовательности видеообъектов, построенных в соответствии с принципами диалогического представления эротетической транзакции, описанными в четвертом разделе.

Логика связи обучающих эпизодов в диалоговый обучающий сценарий относительно проста и определяется следующими двумя возможностями формирования множества ожидаемых ответов реактивным агентом:

- (1) альтернативным выбором одного из элементов субъекта сцены;
- (2) заполнением одного или нескольких каркасов видеообъектов последовательностью символов.

Ниже приведены примеры сцен, демонстрирующие различные варианты организации обучающих эпизодов. Весь материал электронного учебника сгруппирован в 12 основных разделов, охватывающих все грамматические конструкции и группы английского языка.

На рис. П.1 приведена сцена, позволяющая выбрать один из разделов учебника, а на рис. П.2 сцена, позволяющая выбрать один из модулей (unit) раздела Tenses.

Программа English Grammar in Use является сильно структурированной в том смысле, что состоит из множества модулей, каждый из которых содержит материал примерно одинакового уровня сложности. Модули слабо связаны между собой и могут изучаться независимо.

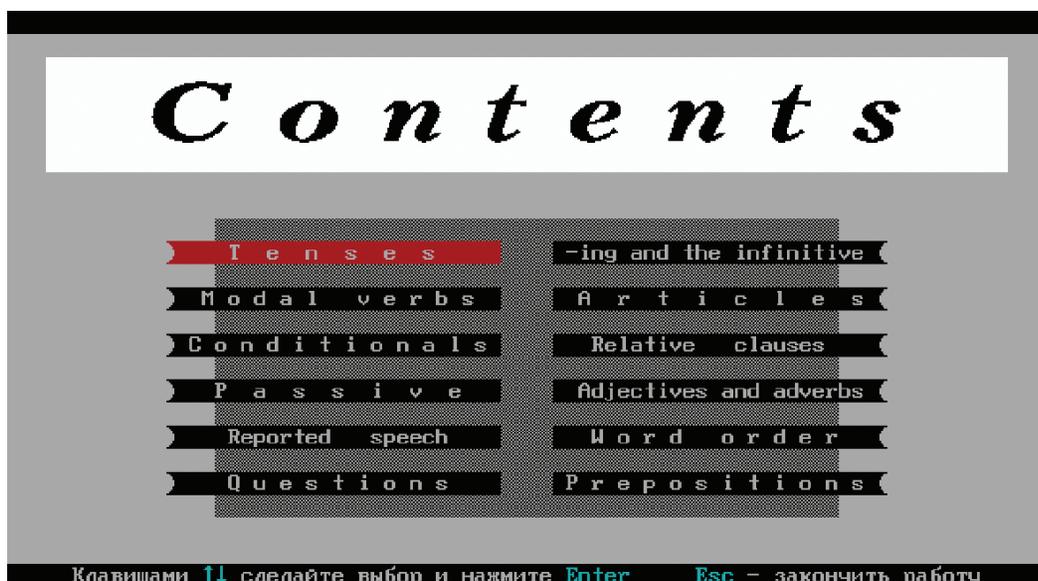


Рис. П.1. Сцена, в которой предлагается сделать выбор одного из разделов учебника “English Grammar in Use”

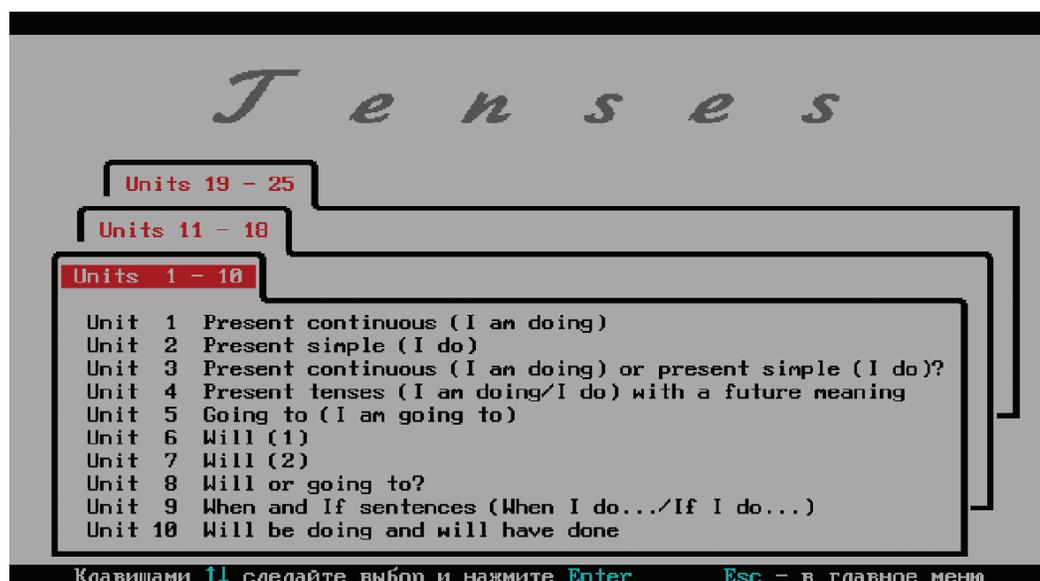


Рис. П.2. Сцена, в которой предлагается сделать выбор одного из модулей раздела Tenses

На рис. П.3 и П.4 приведены примеры начальных сцен двух различных модулей.

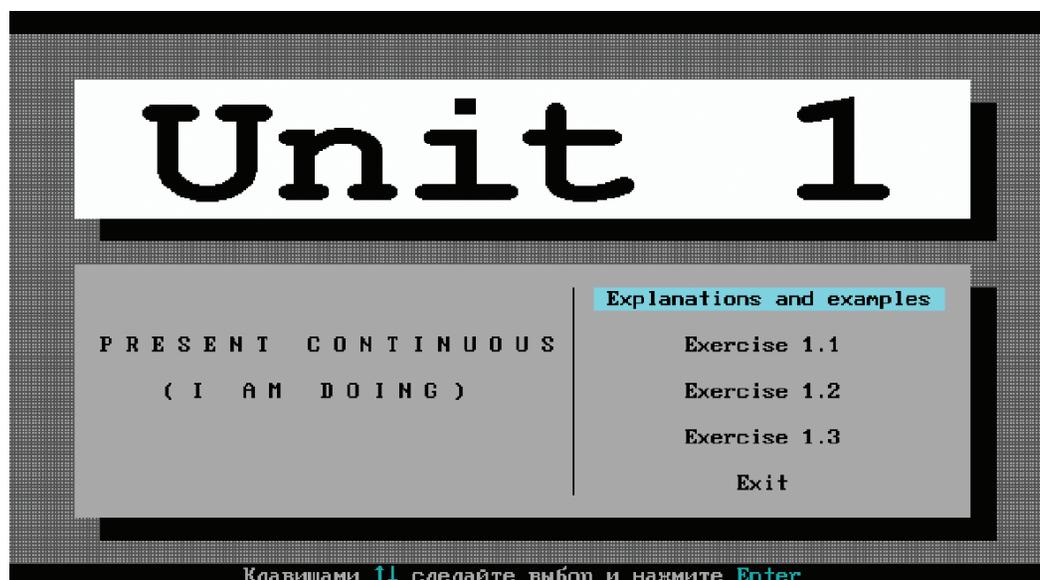


Рис. П.3. Первая сцена модуля 1

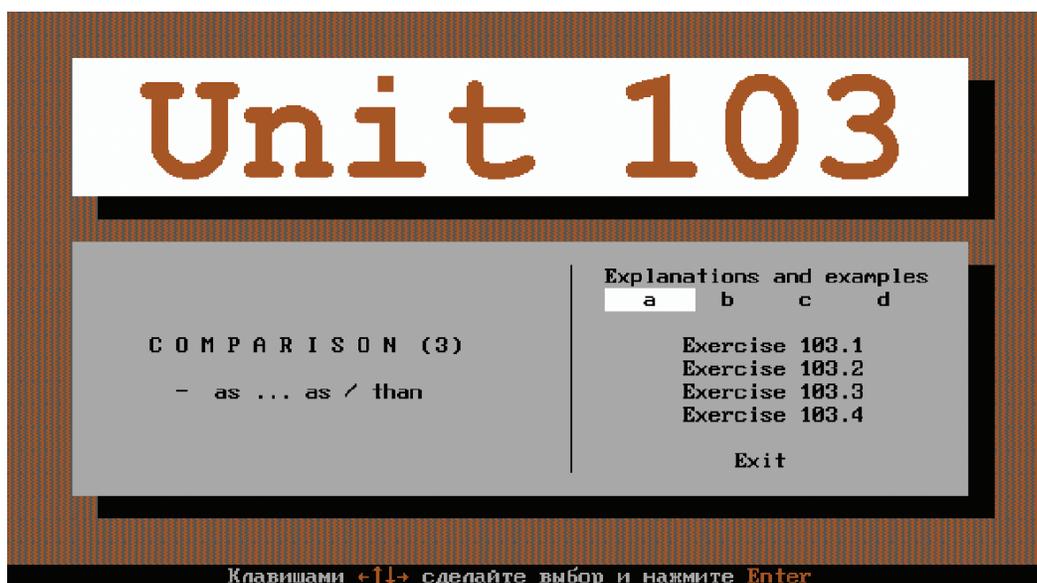


Рис. П.4. Первая сцена модуля 103.
Раздел explanations and examples включает четыре подраздела

Как видно из приведенных выше примеров сцен, программа English Grammar in Use не имеет стандартизованного интерфейса. На каждом шаге она предоставляет реактивному агенту «самодостаточную» информацию в виде сцены, отражающей логическую структуру субъекта и требования вопроса. Такой подход к разработке интерфейса может рассматриваться как вариант интерфейса, адаптирующегося к текущей транзакции диалогового процесса. Вместо стандартизованного интерфейса в программе используется стандартизованный стиль сцены, заключающийся в том, что субъект вопроса всегда располагается в верхней части экрана, а требование, специфицирующее ожидаемый ответ реактивного агента, — в нижней части экрана. Как видно из примеров сцен, приведенных в настоящем приложении, проектирование сцен в виде структуры «субъект — требование» исключает появление тупиковых ситуаций, когда реактивный агент не знает, как продолжить диалоговый процесс.

Одной из особенностей программы English Grammar in Use является широкое использование динамических сцен. Рис. П.5–П.8 иллюстрируют динамику интерпретации сцены. С целью уменьшения количества скриншотов ряд промежуточных фаз интерпретации пропущен.

Динамические сцены проектируются таким образом, чтобы темп появления видеообъектов сцены и темп заполнения объектов-окон информацией был психологически комфортен для реактивного агента. Для этой цели используются соответствующие значения атрибутов поступательного перемещения.

Unit 103 C O M P A R I S O N (3) - as ... as / than



Рис. П.5. Начальная фаза интерпретации динамической сцены.

Unit 103 C O M P A R I S O N (3) - as ... as / than



Joe, Henry and Arthur are all millionaires. They are all very rich. Joe has \$5 million, Henry has \$4 million and Arthur has \$2 million.

Рис. П.6. Промежуточная фаза интерпретации динамической сцены

На рис. П.7 проинтерпретированы все объекты, составляющие субъект сцены. Текст в правом окне появляется порционно в темпе, достаточном для его чтения и восприятия. Темп интерпретации выбран в расчёте на способность к восприятию “слабым” реактивным агентом. Поскольку темп интерпретации, ориентированный на “слабого” реактивного агента, может оказаться слишком медленным для более подготовленных студентов, в программе предусмотрена возможность управления темпом интерпретации сцены самим реактивным агентом. Для “мгновенной” интерпретации текущего окна необходимо нажать комбинацию клавиш: “Alt-W”, а для мгновенной интерпретации всей сцены — комбинацию клавиш: “Alt-S”.

Unit 103 COMPARISON (3) - as ... as / than

Joe Henry Arthur

\$5m \$4m \$2m

Joe, Henry and Arthur are all millionaires. They are all very rich. Joe has \$5 million, Henry has \$4 million and Arthur has \$2 million.

So:

- + Henry is rich.
- + He is richer than Arthur.
- + But he isn't as rich as Joe. (= Joe is richer than he is)

Рис. П.7. Ещё одна промежуточная фаза интерпретации динамической сцены

Метод динамически разворачивающейся сцены использовался также для проектирования сцен-упражнений. В ряде упражнений пользователю предлагается заполнить недостающие фрагменты предложений, используя конечный набор слов, предлагаемый в виде списка.

На рис. П.9 приведена начальная фаза динамической сцены-упражнения этого типа. Требование сцены, кроме альтернативного набора ожидаемых реакций в нижней части экрана, включает также текст в верхней части экрана: “Read this conversation between Brain and Steve. Put each verb into the correct form”. В правой части экрана приведен список слов, который необходимо использовать для заполнения недостающего фрагмента текущего предложения.

Unit 103 C O M P A R I S O N (3) - a s ... a s / t h a n



Joe, Henry and Arthur are all millionaires. They are all very rich. Joe has \$5 million, Henry has \$4 million and Arthur has \$2 million.

So:

- + Henry is rich.
- + He is richer than Arthur.
- + But he isn't as rich as Joe.
(= Joe is richer than he is)

Enter – продолжение работы, F2 – словарь, Esc – к началу блока.

Рис. П.8. Заключительная фаза интерпретации динамической сцены. Сцена дополнена требованием в нижней строке

UNIT 1 Present continuous (I am doing)
Exercise 1.3

Read this conversation between Brain and Steve.
Put each verb into the correct form.

Brain and Steve meet in a restaurant:
Brain: Hello, Steve. I haven't seen you for ages.
What (you/do) these days?

an
are
building
doing
enjoying
helping
is
not
training
working
you

Клавишами ↑↓ сделайте выбор и нажмите Enter

Рис. П.9. Начальная фаза динамической сцены-упражнения.
Заполнение недостающего фрагмента предложения осуществляется путем выбора слов из предлагаемого списка

В том случае, когда реактивный агент делает правильный выбор, слово перемещается по экрану и занимает нужное положение в предложении.

На рис. П.10 показана работа реактивного агента над упражнением. Первое выбранное слово “are” уже заняло своё место в предложении, а только что выбранное слово “you” перемещается по экрану.

На рис. П.11 показан момент завершения работы реактивного агента с первым предложением. Словосочетание “are you” заняло свое место в предложении, а заключительное слово “doing” перемещается по экрану.

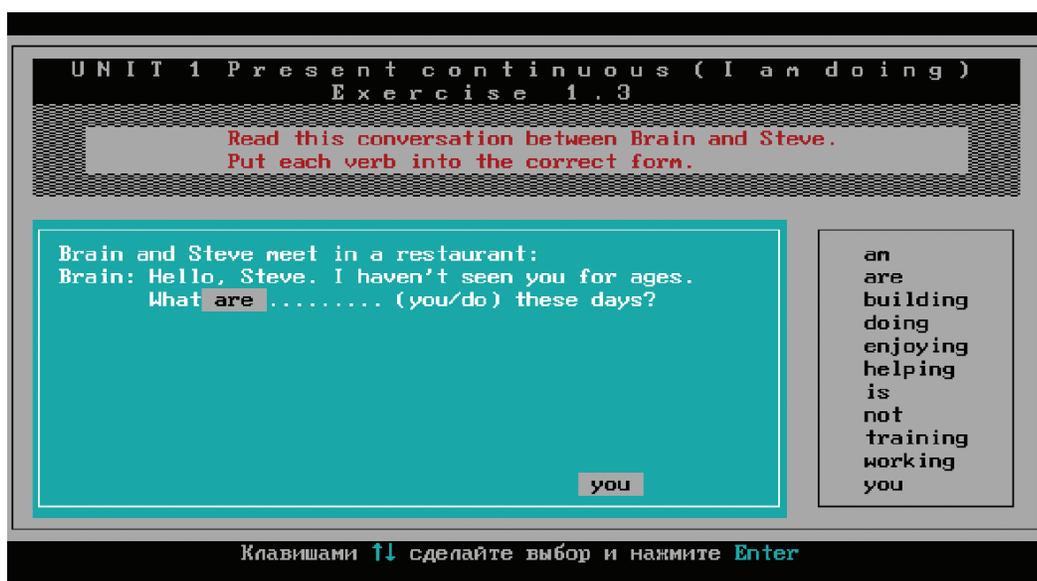


Рис. П.10. Работа реактивного агента с динамической сценой-упражнением.

Ранее выбранное слово “are” заняло своё место в предложении.

Только что выбранное слово “you” перемещается по экрану

В том случае, если во время выполнения упражнения рассматриваемого типа реактивный агент сделал ошибку выбора, он получает контекстно зависимое замечание.

Рис. П.12 и П.13 иллюстрируют «обработку» сценарием ошибки реактивного агента.

При работе над программой соблюдались общие соглашения о логической структуре сцены, способах организации «теоретических» сцен и сцен-упражнений. Однако дизайн сцен от раздела к разделу изменяется, оставаясь однотипным в пределах одного раздела. Рис. П.14 и П.15 иллюстрируют иной вариант дизайна сцены — упражнения и “обработки” ошибки.

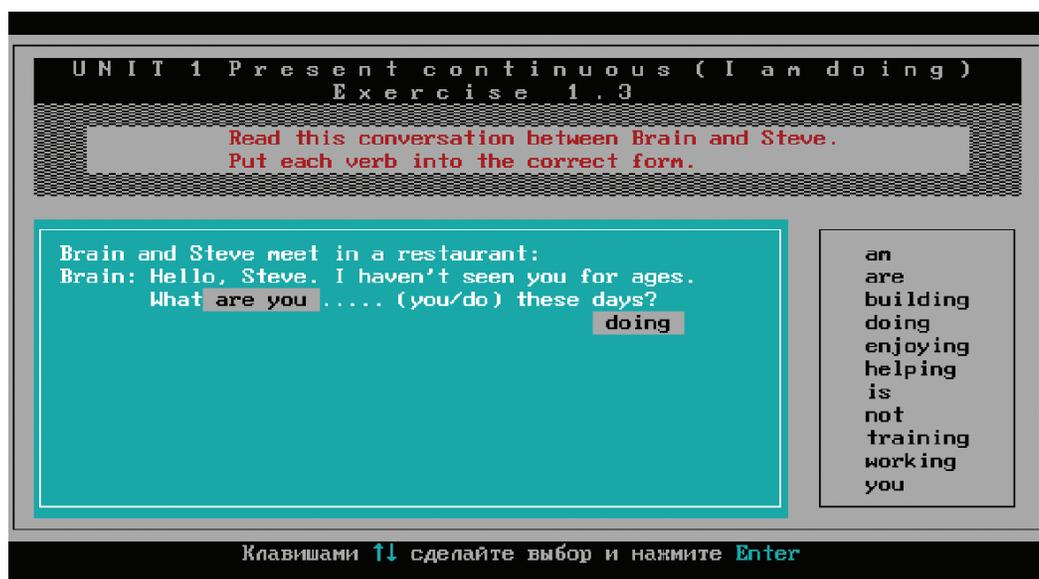


Рис. П.11. Завершение работы реактивного агента с первым предложением. Словосочетание “are you” заняло своё место в предложении, а заключительное слово “doing” перемещается по экрану

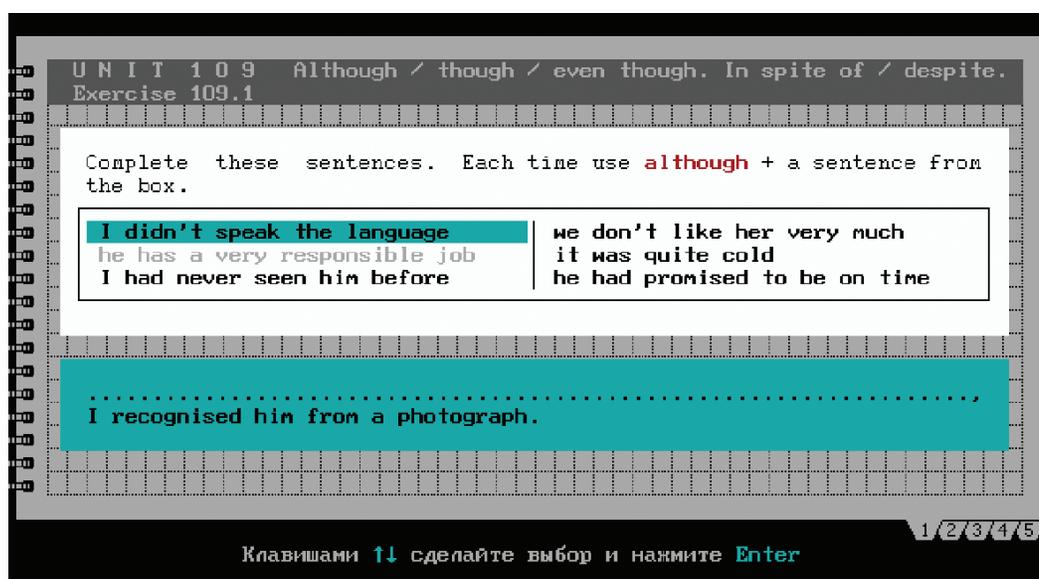


Рис. П.12. Сцена-упражнение, в которой реактивный агент совершает ошибку

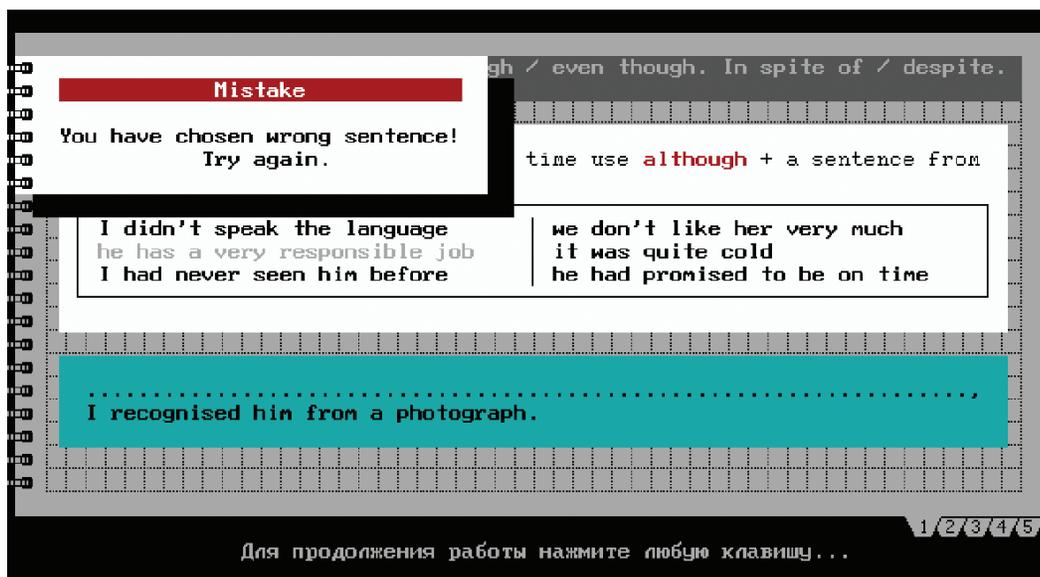


Рис. П.13. Сцена-упражнение, приведенная на рис. П.12, дополненная сообщением об ошибке



Рис. П.14. Сцена, в которой обучаемый совершает ошибку

UNIT 48 Reported speech (2)
Exercise 48.2

Now you have to complete these sentences
with **said**, **told** or **talked**.

Example

- Tom **said** that he didn't like Brian.

2

Ошибка
Вы сделали неправильный выбор. Попробуйте еще раз!

said
told
talked

Для продолжения работы нажмите любую клавишу...

Рис. П.15. Сцена-упражнение, приведенная на рис. П.14,
дополненная сообщением об ошибке