

**БАЛАН В.П.
ДУШКИН А.В.
НОВОСЕЛЬЦЕВ В.И.
СУМИН В.И.**

**ВВЕДЕНИЕ В СИСТЕМНОЕ
ПРОЕКТИРОВАНИЕ
ИНТЕЛЛЕКТУАЛЬНЫХ
БАЗ ЗНАНИЙ**

ББК 66.4(0)
УДК 316.285
Б 74

Рецензенты: начальник кафедры высшей математики Воронежского института МВД России д.ф.-м.н., проф. Меньших В. В.; декан факультета прикладной математики, информатики и механики Воронежского государственного университета д.т.н., проф. Шашкин А. И.

Б 74

Балан В.П., Душкин А.В., Новосельцев В.И., Сумин В.И. **Введение в системное проектирование интеллектуальных баз знаний** / Под ред. В.И. Новосельцева – М.: Горячая линия – Телеком, 2016. – 107 с.

ISBN

Монография вводит читателя в новую область теории и практики информатизации человеческой деятельности – системное проектирование интеллектуальных баз знаний. Содержит семь разделов, раскрывающих системные принципы построения интеллектуальных баз знаний; языковые средства представления знаний; технологии представления знаний; нейросетевые методы представления знаний; типовые методы поиска решений в интеллектуальных базах знаний; оценку качества проектов по созданию интеллектуальных баз знаний; основы управления проектами по созданию интеллектуальных баз знаний.

Основу монографии составляют авторские методики, успешно апробированные при проектировании и разработке ряда информационно-телекоммуникационных систем в различных отраслях народного хозяйства.

Адресована научным работникам и специалистам в области автоматизированных систем управления. Будет полезна аспирантам и студентам старших курсов, специализирующихся на проблемах информатизации социальных, экономических, технологических и других процессов.

ISBN

ББК 66.4(0)
УДК 316.285

© Балан В.П., Душкин А.В.,
Новосельцев В.И., Сумин В.И., 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1. СИСТЕМНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ	9
1.1. ПОНЯТИЕ ИНТЕЛЛЕКТУАЛЬНОЙ БАЗЫ ЗНАНИЙ	9
1.2. ПРИНЦИП ОРГАНИЗАЦИИ ОБЩЕНИЯ МЕЖДУ ПОЛЬЗОВАТЕЛЕМ И ИБЗ	10
1.3. МОДЕЛИ МАШИННОГО ПОНИМАНИЯ ВХОДНЫХ ТЕКСТОВ	13
1.4. ТИПОВАЯ СТРУКТУРА ИНТЕЛЛЕКТУАЛЬНОЙ БАЗЫ ЗНАНИЙ	18
1.5. РЕЖИМЫ РАБОТЫ ИНТЕЛЛЕКТУАЛЬНОЙ БАЗЫ ЗНАНИЙ ...	23
2. ЯЗЫКОВЫЕ СРЕДСТВА ПРЕДСТАВЛЕНИЯ ЗНАНИЙ	24
2.1. ОБЗОР ЯЗЫКОВ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ	24
2.2. СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ЯЗЫКОВ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ	42
3. ТЕХНОЛОГИИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ	44
3.1. ПРЕДСТАВЛЕНИЕ ЗНАНИЙ РОЛЕВЫМИ ФРЕЙМАМИ	47
3.2. ПРЕДСТАВЛЕНИЕ ЗНАНИЙ КОНЦЕПТУАЛЬНЫМИ СЕМАНТИЧЕСКИМИ СЕТЯМИ	51
3.3. ПРЕДСТАВЛЕНИЯ ЗНАНИЙ	54
3.4. ТЕХНОЛОГИЯ ПОСТРОЕНИЯ ПРОЦЕДУРНОЙ КОМПОНЕНТЫ БАЗЫ ЗНАНИЙ	56
4. НЕЙРОСЕТЕВЫЕ МЕТОДЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ	61
4.1. ИДЕИ НЕЙРОИНФОРМАТИКИ	61
4.2. БАЗОВЫЕ ЭЛЕМЕНТЫ И АРХИТЕКТУРА НЕЙРОСЕТЕЙ	63
5. ТИПОВЫЕ МЕТОДЫ ПОИСКА РЕШЕНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ БАЗАХ ЗНАНИЙ	67
5.1. ПОИСК РЕШЕНИЙ НА СЕМАНТИЧЕСКИХ СЕТЯХ	67
5.2. СИТУАЦИОННЫЙ ПОИСК РЕШЕНИЙ	74
5.3. МНОГОКРИТЕРИАЛЬНЫЙ ПОИСК РЕШЕНИЙ	77
5.4. ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ	80
6. МЕТОДЫ ОЦЕНКИ КАЧЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ	84
6.1. КРИТЕРИИ КАЧЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ	84
6.2. ИНТЕГРАЛЬНАЯ ОЦЕНКА КАЧЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ	86
6.3. ОЦЕНКА КАЧЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ ПО ЛОКАЛЬНЫМ КРИТЕРИЯМ	94

7. ОСНОВЫ УПРАВЛЕНИЯ ПРОЕКТАМИ ПО СОЗДАНИЮ ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ	97
7.1. ГЛАВНЫЕ ПОНЯТИЯ	97
7.2. ОСОБЕННОСТИ УПРАВЛЕНИЯ ПРОЕКТАМИ	97
7.3. КОНЦЕПЦИЯ УПРАВЛЕНИЯ ПРОЕКТАМИ	98
7.4. ТЕХНОЛОГИЯ УПРАВЛЕНИЯ ПРОЕКТАМИ	99
7.5. ВИДЫ НЕОПРЕДЕЛЕННОСТЕЙ ПРИ УПРАВЛЕНИИ ПРОЕКТАМИ И СПОСОБЫ ИХ УЧЕТА	100
7.6. ПРИЕМКА ПРОЕКТА.....	103
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	105

ВВЕДЕНИЕ

Наше время – это эпоха интенсивного внедрения информационных технологий практически во все сферы человеческой деятельности. Ожидается, что такая тенденция не только сохранится, но будет прогрессировать в обозримом будущем. В практическом плане эти технологии реализуются в виде различных информационно-коммуникационных систем, типа ГАС РФ «Выборы» и «Правосудие», разнообразных отраслевых АСУ, действующих в общесоюзных, республиканских, краевых и областных министерствах и ведомствах, многочисленных информационных и экспертных систем предприятий, организаций и фирм.

Неотъемлемой составной частью, а точнее – душой, всех систем этого типа становятся интеллектуальные базы знаний, которые в первом приближении можно определить как особого рода информационно-программные комплексы, способные напрямую (без посредников) вести диалог с человеком и оперировать при этом не только данными, но главным образом знаниями (метаданными).

В настоящее время, для построения интеллектуальных баз знаний используются готовые программные продукты типа Oracle, MSSQL, SyBASE и другие, дополняемые различными программными модулями, в которые по схеме «естественный язык → компьютерная программа» закладываются знания о проблемной области. В результате получается некий бессистемный конгломерат, в котором первичной выступает программная среда, отражающая в основном предметные и частично лингвистические данные, а декларативные, процедурные и лингвистические знания присутствуют постольку, поскольку это допускают возможности данного программного продукта и квалификация программистов-разработчиков. Другими словами, основу существующей технологии поддержки проектных решений при создании интеллектуальных баз знаний составляет принцип: *«делаем то, что можем, а не то, что нужно»*. Негативные последствия такого подхода очевидны: конечный пользователь, владеющий максимальными знаниями о предметной области, фактически исключается из процесса проектирования и разработки баз знаний. В результате на выходе проекта он получает не то, что ему нужно для обеспечения профессиональной деятельности, а то, что могут сделать программисты-разработчики, используя имеющиеся программные платформы. И обусловлено это не квалификацией программистов и не низкими функциональными возможностями используемых программных продуктов, а самой технологией поддержки проектных решений по созданию интеллектуальных баз знаний.

Если исключить из рассмотрения многочисленные подробности «технического» плана, то смысл и сущность традиционной технологии

проектирования баз знаний заключается в непосредственном переводе описаний предметной области с естественного языка на математико-программный язык, «понятный» компьютеру. А такой перевод фактически исключает конечного пользователя из процесса проектирования баз знаний, оставляя ему только «начало» – формирование замысла проекта и «конец» – приемку готового проекта. Именно ограниченность палитры используемых языковых средств обуславливает все те трудности, которые приходится преодолевать разработчикам проектов по созданию интеллектуальных баз знаний, и предопределяет те неудачи, которые фактически превращают компьютеры не в интеллектуальных партнеров человека, а в «хранилище» данных, печатающие машинки или в быстродействующие логарифмические линейки.

Таким образом, в настоящее время имеет место и все более прогрессирует реальное противоречие между насущной потребностью широкого внедрения и использования интеллектуальных баз знаний и несовершенством применяемых технологий их проектирования. При этом негативное последствие заключается в том, что конечный пользователь, оставаясь в стороне от процесса проектирования баз знаний, в конечном счете получает не актуального помощника, а некоторый программный продукт, кое-как подстроенный под его нужды.

Вместе с тем в современной теории искусственного интеллекта происходит интенсивное развитие новых языковых средств, в частности реляционного и ролевого типов. Эти языки позволяют записывать и генерировать правила логического вывода (то есть работать с декларативными знаниями), а также создавать управляющие структуры, обеспечивающие вывод новых знаний (то есть оперировать с процедурными знаниями). Их использование открывает новые возможности по описанию фактов и закономерностей предметной области, и позволяет предложить более совершенную технологию поддержки принятия проектных решений в процессе проектирования и создания интеллектуальных баз знаний, свободную от указанных выше недостатков.

Однако специалисты в области информатизации еще не полностью преодолели инерцию традиционного мышления. Многие из них по-прежнему рассматривают проектирование интеллектуальных баз знаний как искусство программирования с ориентацией на существующие программные продукты (языки программирования, стандартные базы данных, типовые интерфейсы и т.п.), но не как науку, опирающуюся на системную методологию, технологию, модели анализа и оптимизации проектных решений. В преодолении этой порочной традиции авторы видели свою основную задачу при написании монографии.

Монография содержит семь разделов, в которых обобщен опыт более чем двадцатилетней деятельности авторов в области системного

проектирования интеллектуальных баз знаний в составе объектов автоматизации различного (в том числе и военного) назначения.

Первый раздел посвящен описанию системных принципов построения интеллектуальных баз знаний, включая такие вопросы как: организация общения между пользователем и базой знаний; модели машинного понимания входных текстов; структура интеллектуальной базы знаний и режимы ее работы.

Во втором разделе проводится анализ существующих и перспективных языковых средств представления знаний. В качестве объектов анализа рассматриваются: язык Кодда; логические языки; язык нечетких множеств; контекстно-свободный плекс-язык; язык RX-кодов; язык семантических сетей; язык ролевых фреймов; тензорный язык Крона. Суть анализа заключалась в упорядочении указанных языков по двум критериям: выразительности и мощности его инструментальных средств.

Третий раздел посвящен технологиям представления знаний. Рассматривается общая технология, базирующаяся на принципах этапности, последовательного наращивания уровня формализации представления знаний при переходе от этапа к этапу, поддержки проектных решений путем использования четырех частных технологий, а именно: представления знаний ролевыми фреймами, на основе концептуальных и терминальных семантических сетей, а также построения процедурной компоненты базы знаний. Раскрывается содержание этих технологий и показывается, что они обладают по сравнению с традиционно используемой процедурой «естественный язык → компьютерная программа» тем преимуществом, что позволяют заменить эвристические соображения строго формальными методами задания единиц знаний, снизить требования к языкам программирования, а так же более полно использовать знания конечного пользователя о предметной области.

В четвертом разделе рассматриваются нейросетевые методы представления знаний, включая такие вопросы как: основные идеи нейроинформатики применительно к построению интеллектуальных баз знаний; базовые элементы и архитектура нейросетей.

В пятом разделе описываются типовые методы поиска решений, которые, целесообразно реализовывать при системном проектировании интеллектуальных баз знаний, а именно: поиск решений на семантических сетях, ситуационный поиск решений, многокритериальный поиск решений и поиск решений на основе генетических алгоритмов. Первые два метода применяются в тех предметных областях, в которых оперируют не столько числами, сколько понятиями и нестрогими категориями. Два других метода эффективны в тех случаях, когда предметную область удастся описать числами и строгими математическими соотношениями.

Шестой раздел посвящен описанию методов оценки качества интеллектуальных баз знаний. В отличие от существующих фрагментарных подходов предлагаемые в монографии методы обеспечивают всесторонний и комплексный учет социальных, функциональных, эргономических, прагматических, экономических, технических и технологических критериев качества интеллектуальных баз знаний, выраженных как в числовых, так и в понятийных категориях.

В седьмом разделе рассматриваются вопросы управления проектами по созданию интеллектуальных баз знаний, включая особенности, концепцию и технологию управления данными проектами, а также неопределенности при управлении проектами и способы их учета.

1. СИСТЕМНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ

1.1. ПОНЯТИЕ ИНТЕЛЛЕКТУАЛЬНОЙ БАЗЫ ЗНАНИЙ

Под интеллектуальной базой знаний (ИБЗ) понимается управляемый комплекс языковых, алгоритмических, программных и технических средств, предназначенных для восприятия, обработки, хранения и выдачи (отображения) знаний о предметной области, включающий четыре компонента:

а) упорядоченные каким-либо способом факты и данные, отражающие модель профессиональной сферы (*предметные данные*);

б) правила, модели, алгоритмы и программы, позволяющие рассчитывать определенные показатели объектов профессиональной сферы, строить цепочки логических выводов и на этой основе делать обобщения и заключения, а так же вызывать определенные ассоциации (*декларативные знания*);

в) управляющая и интерпретирующая структура, определяющая порядок и способы применения моделей и правил логического вывода для получения или трансформации информации (*процедурные знания*);

г) правила морфологического, синтаксического и семантического анализа входных и выходных текстов, а также списки основ слов, которые используются для организации диалога между базой знаний и пользователем (*лингвистические знания*).

Если оставить в стороне практически бесплодные рассуждения о том, чем данные отличаются от знаний, то с точки зрения системотехники различия между понятиями «база данных» и «база знаний» можно выразить в виде следующих формул:

«база данных» = «предметные данные» + «управление данными» + «лингвистические единицы»;

«интеллектуальная база знаний» = «предметные данные» + «декларативные знания» + «процедурные знания» + «лингвистические знания».

Как видно из приведенного определения ИБЗ, охватывая традиционное понимание баз данных, представляют собой их дальнейшее развитие и наращивание по следующим главным направлениям:

- интеллектуализации общения с конечным пользователем в форме, не требующей участия в диалоге посредника-программиста;

- формирование моделей проблемных ситуаций по их содержательному описанию, представленному пользователем;

- расширения понятия «данные» от уровня чисел, текстов, схем и других простейших информационных атрибутов до уровня закономерностей, правил, алгоритмов и других операций, обеспечивающих ин-

теллектуальную поддержку принятия управленческих решений в условиях неопределенности;

- обеспечение семантической целостности знаний и данных, их проверку на полноту, корректность и определенность;

- усложнения алгоритмов обработки информации до уровня имитации таких интеллектуальных механизмов мышления человека как: анализ и обобщение информации, построение логических цепочек вывода новых утверждений, генерация альтернативных вариантов решений, поиск оптимальных решений и других;

- планирование вычислений и логической обработки данных с целью решения сформулированных пользователем задач количественного и качественного анализа моделей проблемных ситуаций;

- оформление, интерпретация и отображение результатов решения задач в удобном для пользователя виде.

Таким образом, ИБЗ, будучи реализованной в компьютере, предоставляет пользователю следующие возможности. Пользователь на своем профессиональном языке (разумеется, по необходимости ограниченном и формализованном) вводит в компьютер описание проблемной ситуации, ставит задачи ее анализа и задает исходные данные. После чего компьютер сам формирует программу решения этих задач, решает задачи, производит анализ результатов и выдает ответы в удобном для пользователя виде (графическом, текстовом, картографическом и др.).

При этом основной эффект перехода от баз данных к интеллектуальным базам знаний заключается в более полном удовлетворении потребностей пользователя информацией, необходимой ему для принятия решений, за счет повышения статуса компьютера и компьютерных сетей, поставляющих конечному пользователю не только информацию к размышлению, но и мотивированные варианты возможных решений. Естественно, что речь идет не об однократном решении какой-либо задачи, а о диалоговом общении пользователя с ИБЗ при многократном повторении цикла: подготовка исходных данных, решение задачи, анализ решения, коррекция исходных данных.

1.2. ПРИНЦИП ОРГАНИЗАЦИИ ОБЩЕНИЯ МЕЖДУ ПОЛЬЗОВАТЕЛЕМ И ИБЗ

Под общением понимается процесс обмена информацией между пользователем и ИБЗ, обусловленный необходимостью последовательного и(или) параллельного выполнения человеком и машиной действий по совместному решению какой-либо задачи.

Наиболее распространенным и интересным с позиции проектирования видом общения является диалог, инициируемый ИБЗ при реше-

нии задач, поставленных пользователем. Традиционная схема такого диалога века показана на рис. 1.1.

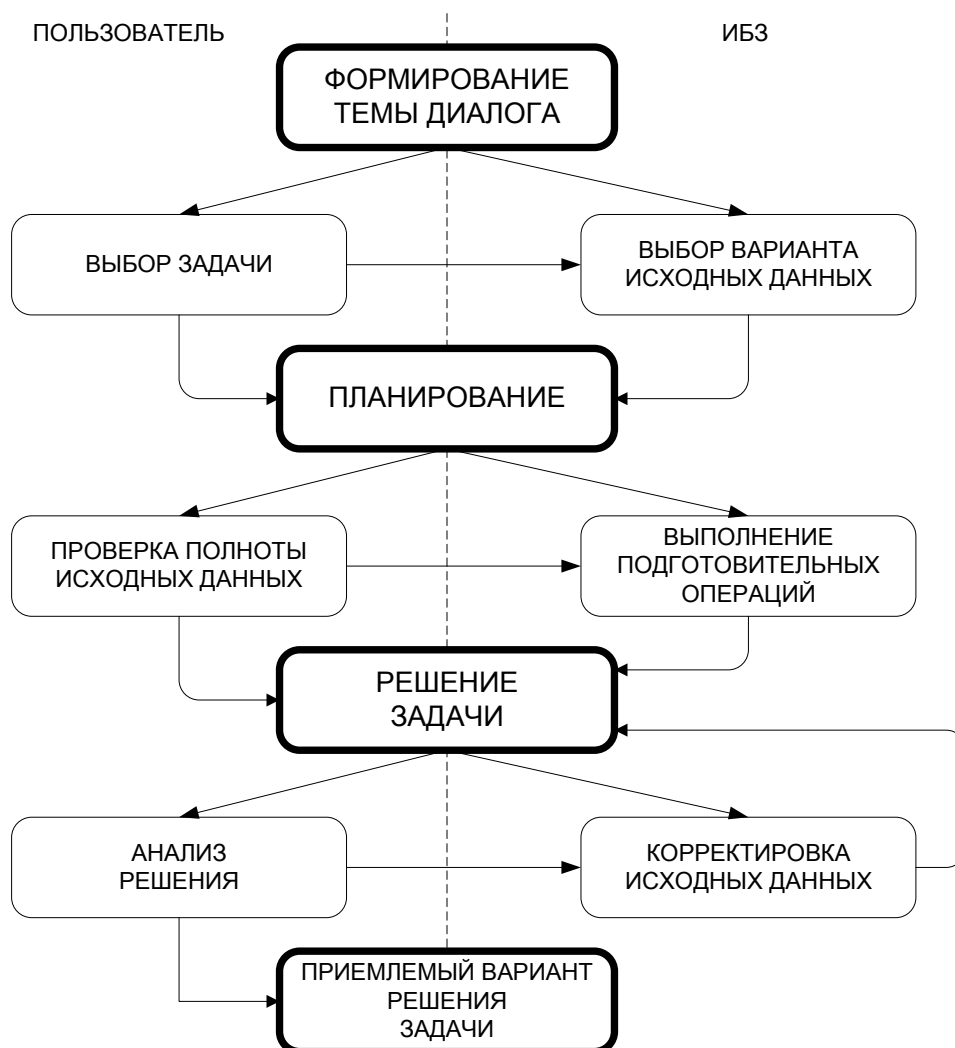


Рис. 1.1. Схема диалога, инициируемого ИБЗ при решении задач, поставленных пользователем

Здесь предполагается, что вся совокупность информации, необходимой для решения прикладной задачи доступна пользователю. Однако в действительности каждого пользователя интересует сравнительно небольшое количество так называемых персонально ориентированных задач, и он в состоянии варьировать лишь некоторой частью параметров, описывающих предметную область и критериев выбора решений.

В результате при полном доступе пользователю может поступать излишняя информация, на которую тратятся вычислительные ресурсы, а время между введением новой корректировки параметров и получением решения может оказаться неприемлемо большим. Этого можно

избежать, если при построении ИБЗ использовать принцип гибкого формирования персонально ориентированных задач с соответствующим уровнем детализации и специализацией информации, предоставляемой пользователю, как показано на рис. 1.2.

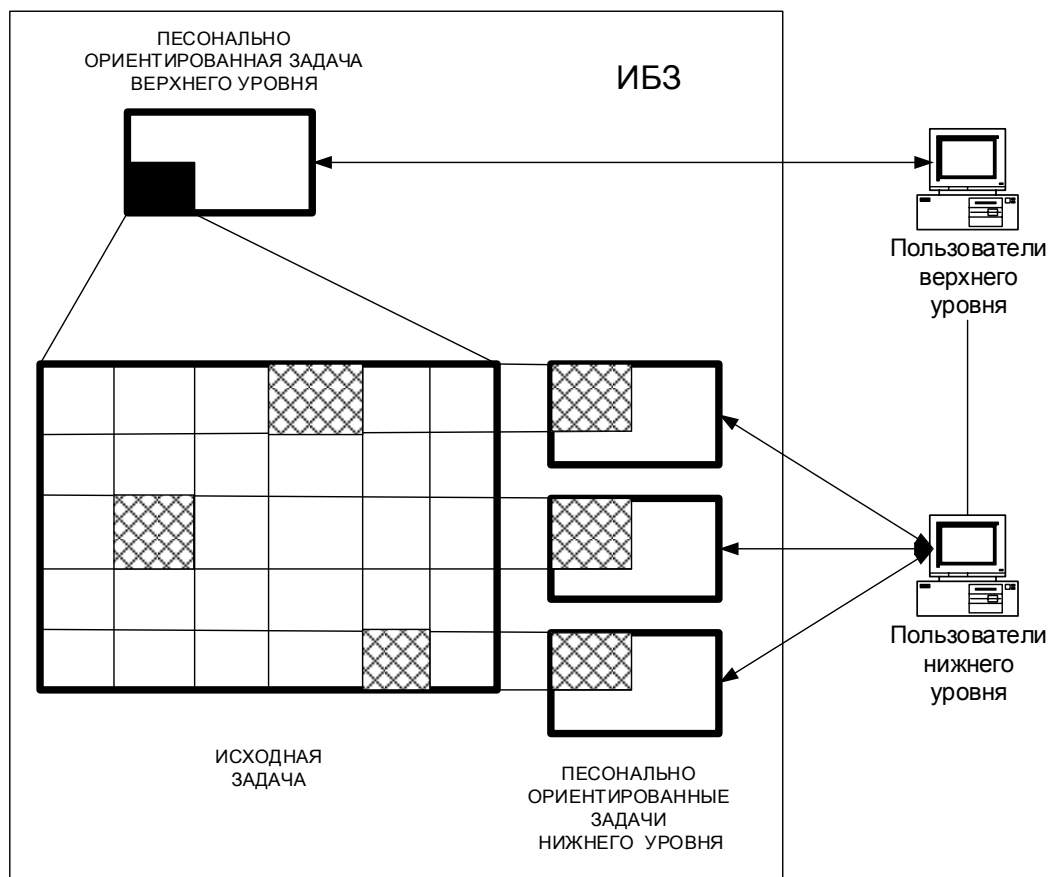


Рис. 1.2. Принцип формирования персонально ориентированных задач при построении ИБЗ

На этом рисунке в персонально ориентированных задачах нижнего уровня оставлены без изменения некоторые параметры исходной задачи, а остальные параметры подвергнуты агрегированию. Такие задачи можно назвать частично агрегированными. Они позволяют конкретному пользователю, работая с ИБЗ, оперировать только тем набором информации, который его интересует и входит в его компетенцию. Внесение изменений в частично агрегированные задачи позволяет исследовать определенный аспект проблемной области, проанализировать внутренние взаимосвязи и выбрать наиболее эффективные решения.

Персонально ориентированная задача верхнего уровня строится в терминах агрегированных параметров, состав и способ построения которых может меняться в зависимости от изучаемого вопроса, от желания пользователя проследить связь между теми или иными агрегиро-

ванными параметрами. Такую задачу следует назвать полностью агрегированной, тем самым подчеркивается, что речь идет об укрупненном описании поведения всех параметров исходной задачи.

Таким образом, при реализации ИБЗ на основе принципа гибкого формирования персонально ориентированных задач предполагается, что каждая исходная задача большой размерности является тем материалом, из которого строятся частично агрегированные задачи, ориентированные на потребности конечных пользователей. Тем самым решается одна из центральных задач проектирования ИБЗ – сокращения времени их реакции на запросы пользователей. И осуществляется это за счет перехода от одноуровневого к иерархическому принципу организации общения между ИБЗ и коллективным пользователем. Кроме того, по сравнению с традиционным принципом «все для всех» реализация данного принципа повышает информационную безопасность ИБЗ и делает процесс их проектирования более технологичным.

1.3. МОДЕЛИ МАШИННОГО ПОНИМАНИЯ ВХОДНЫХ ТЕКСТОВ

Непременным требованием к ИБЗ является требование ведения диалога с пользователем на проблемно ориентированном естественном языке. Соответственно при проектировании ИБЗ на первый план выходит проблема машинного понимания естественно-языковых текстов.

Традиционно при решении этой проблемы использовалась трехкомпонентная модель «текст – ИБЗ – смысл», представленная на рис. 1.3.

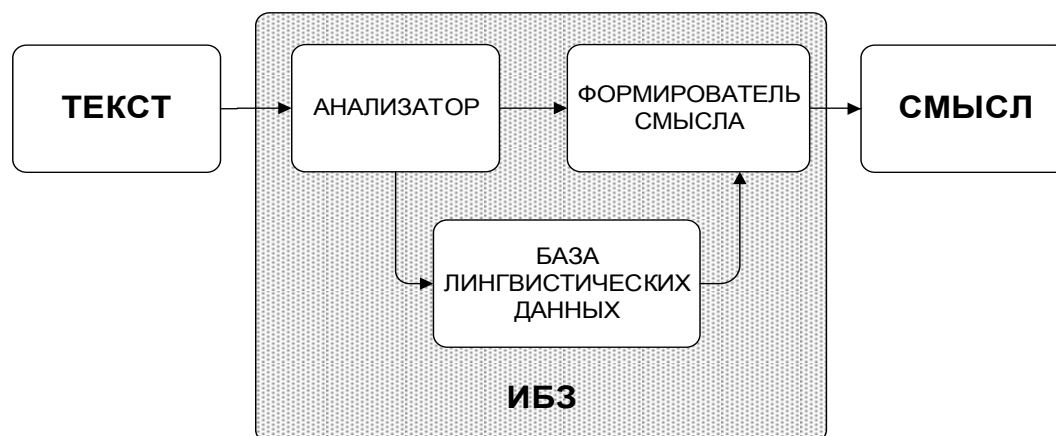


Рис. 1.3. Традиционная трехкомпонентная модель машинного понимания естественно-языковых текстов («текст-ИБЗ-смысл»)

В этой модели имеется «текст» как некоторая данность, ИБЗ, снабженная средствами лингвистического анализа текста, базой линг-

вистических данных и алгоритмами формирования (экспликации) смысла текста, и собственно «смысл», как результат работы ИБЗ. Такая модель равносильна предположению, что о тексте ничего не может быть сказано заранее; поэтому при анализе текстовых предложений фактически ищется ответ лишь на абстрактный вопрос «что есть в предложении?» при том предположении, что в нем может быть все, что угодно.

Очевидно, что с системной точки зрения эта модель неполна: в ней отсутствуют два важных компонента – пользователь, решающий с помощью ИБЗ проблемно ориентированные задачи, и реакция ИБЗ на запрос пользователя. Без дополнения модели рис. 1.3 этими компонентами синтез процесса машинного понимания естественно-языковых текстов является неправомерным, а сколь угодно искусная реализация этой модели никогда не завершится созданием эффективных компьютерных программ понимания входных текстов.

Структура расширенной пятикомпонентной модели «пользователь – текст – ИБЗ – реакция – смысл» представлена на рис. 1.4.

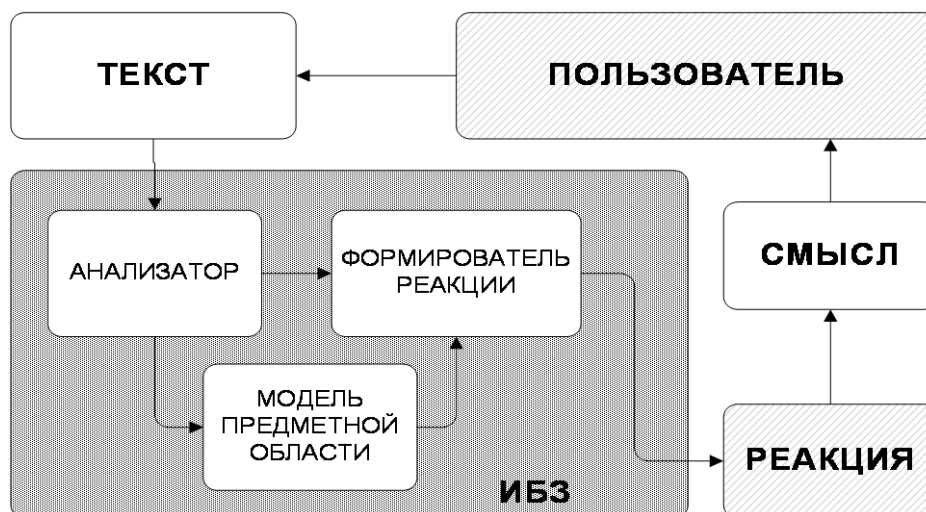


Рис. 1.4. Расширенная пятикомпонентная модель машинного понимания естественно-языковых текстов («пользователь-текст-ИБЗ-реакция-смысл»)

В этой модели критерием понимания поступающего от пользователя текста является не расшифровка его абстрактного содержания (как это предполагалось в предыдущем случае), а адекватная реакция ИБЗ на проблемно ориентированный вопрос пользователя. При таком подходе оказывается возможным построить априорные утверждения о некоторых целевых объектах, из которых состоит входной текст, с помощью которых анализ может быть ориентирован не на пассивную регистрацию того, «что есть в предложении», а на активный поиск в тексте

ключевых объектов, стимулирующих ИБЗ к той или иной реакции, и тем самым определяющим смысл входного текста. Таким образом, если ИБЗ вырабатывает в ответ на входящий текст ожидаемую пользователем реакцию, то и имеет место понимание ИБЗ смысла вопроса.

Следует отметить, что использование расширенной пятикомпонентной модели в рассмотренном выше понимании накладывает определенные ограничения на содержание и структуру входных естественно-языковых текстов. Во-первых, такой текст должен содержать информацию двоякого рода: а) информацию для выбора ИБЗ алгоритма реагирования, адекватного запросу пользователя; б) данные для запуска этого алгоритма, то есть конкретные значения его параметров. Во-вторых, во входном тексте должны обязательно содержаться лингвистические объекты, для которых в ИБЗ в числе возможных реакций имеется хотя бы одна реакция, адекватная данному запросу пользователя при какой-либо совокупности обстоятельств.

Естественно, что указанные ограничения в определенной мере формализуют общение пользователя с ИБЗ, но одновременно делают однозначным понятие «смысл текста», связывая его с целью, которую преследует пользователь, общаясь с ИБЗ.

Для более подробного рассмотрения обсуждаемой модели машинного понимания естественно-языковых текстов введем следующие объекты:

$T^{(p)}$ – множество всех доступных ИБЗ предложений входного проблемно ориентированного текста;

r_i – i -й класс однотипных реакций ИБЗ на запрос пользователя;

R – множество всех классов однотипных реакций ИБЗ, которые могут потребоваться пользователю;

f_i – алгоритм формирования реакции из класса r_i ;

$F^{(p)}$ – множество всех возможных алгоритмов формирования реакции;

W – алгоритм выбора алгоритма из $F^{(p)}$;

$C^{(p)}$ – интерпретация ИБЗ конкретной совокупности обстоятельств, в рамках которых анализируется запрос пользователя и выдается ответ;

$C^{(s)}$ – интерпретация пользователем совокупности обстоятельств, в условиях которых функционирует ИБЗ;

t – предложение входного текста из $T^{(p)}$;

t_w – часть («субтекст») текста t , предназначенная для обеспечения выбора алгоритма из $F^{(p)}$ («код» выбора алгоритма реагирования);

t_v – часть («субтекст») текста t , содержащая значения («код» выбора конкретной реакции в классе r_i однотипных реакций);

α – алгоритм членения текста входного предложения на субтексты t_w и t_v .

Используя введенные обозначения, процесс понимания ИБЗ предложений входного текста t в формальном представлении выглядит следующим образом:

$$\begin{aligned} 1. \alpha(t) &\Rightarrow t_w, t_v, \quad t \in T^{(p)}; \\ 2. w(t_w, C^{(p)}, C^{(s)}) &\Rightarrow f_i, \quad f_i \in F^{(p)}; \\ 3. f_i(t_v) &\Rightarrow r_i', \quad r_i' \in r_i \in R. \end{aligned} \tag{1.1}$$

На первом шаге алгоритм членения α выделяет в тексте входного предложения t субтексты t_w и t_v . На втором шаге алгоритм W производит по данным $t_w, C^{(p)}, C^{(s)}$ выбор алгоритма реагирования f_i из множества $F^{(p)}$. На третьем шаге алгоритм f_i с учетом субтекста t_v вырабатывает конкретную реакцию ИБЗ r_i' , выбирая ее из класса однотипных реакций r_i .

В графическом виде описанный выше процесс машинного понимания естественно-языковых текстов с учетом того, что каждый из указанных шагов может завершиться неудачей, представлен на рис. 1.5.

Сопоставим рассмотренную и традиционную модели понимания естественно-языковых текстов. Как уже отмечалось, традиционная ненаправленная лингвистическая обработка текстовых предложений состоит в выявлении и экспликации его «смысла». Вместе с тем, конструктивного определения понятия «смысл предложения» (как и понятия «смысл» вообще) не существует. Поэтому, реализация традиционной модели «текст – ИБЗ – смысл» при разработке ИБЗ, приводит, в конце концов, к тому, что проектировщики «заставляют» компьютер искать «то – не знаю, что». В рамках же расширенной пятикомпонентной модели «пользователь – текст – ИБЗ – реакция – смысл» совершенно точно и определенно известно, что ищется: ищутся объекты t_w и t_v , первый из которых доставляет конкретные значения параметров в алгоритме выбора алгоритма реагирования ИБЗ, второй – конкретные значения параметров алгоритма реагирования. Правильное выявление объектов t_w и t_v при работоспособном алгоритме W и верных $C^{(p)}, C^{(s)}$ как раз и равносильно пониманию ИБЗ смысла входного предложения, поскольку в этом случае обеспечивается выработка ожидаемой реакции.

Нетрудно заметить, что основные трудности практической реализации расширенной пятикомпонентной модели «пользователь – текст – ИБЗ – реакция – смысл» машинного понимания естественно-языковых текстов связаны с тремя обстоятельствами. Во-первых, алгоритм α , выделяя в тексте входного предложения t субтексты t_w и t_v , может допускать ошибки, что обусловлено неоднозначностью и расплывчатостью понятий естественного языка.

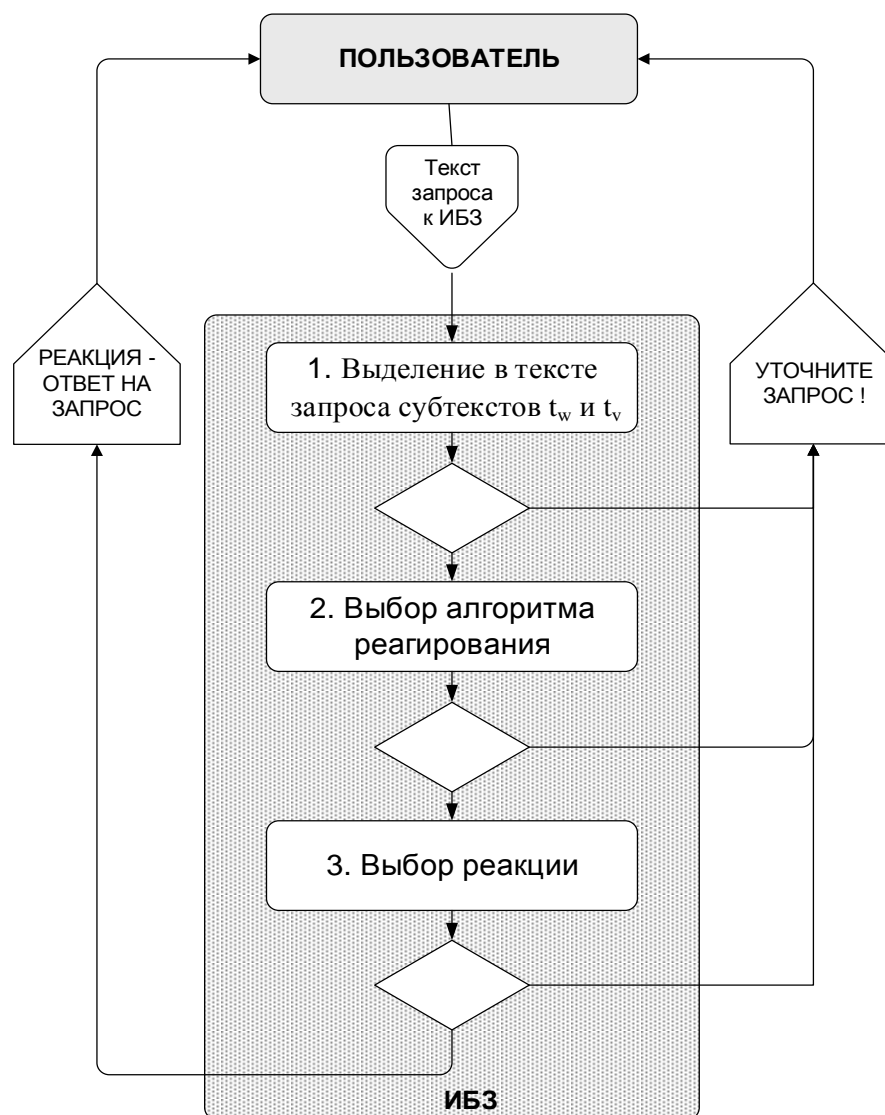


Рис. 1.5. Целенаправленный процесс машинного понимания естественно-языковых текстов

Во-вторых, отображение множества всех субтекстов t_w на множество $F^{(p)}$, которое осуществляет алгоритм W , в общем случае может быть неоднозначным: при разных комплексах обстоятельств $C^{(p)} \in C^{(s)}$ одному и тому же входному предложению (и в его составе одному и тому же субтексту t_w) могут быть адекватны различные типы реакций. В-третьих, алгоритм f_i , вырабатывая конкретную реакцию r_i из класса однотипных реакций r_i , так же может допускать ошибки, обусловленные тем, что одно и то же входное предложение (и в его составе один и тот же субтекст t_v) может ассоциировать разные реакции ИБЗ.

Преодоление этих трудностей при системном проектировании ИБЗ возможно двумя способами. Во-первых, путем жесткой формализации предложений входных текстов, до уровня, когда все запросы

пользователя к ИБЗ составляются по заранее определенным формам и с использованием унифицированного тезауруса. Одновременно эти формы и тезаурус заносятся в память компьютера и образуют базу лингвистических знаний, которая пополняется и модифицируется в процессе эксплуатации ИБЗ. Во-вторых, путем организации специальной процедуры обучения ИБЗ, заключающейся в том, что в память компьютера заранее записываются корреляционные правила вида:

$$\begin{aligned} (C^{(p)}, C^{(s)}) : f_w^i &\Rightarrow r_k \in R; \\ (t_v) &\Rightarrow r_k' \in r_k. \end{aligned} \quad (1.2)$$

Правила (1.2) называются предписывающими и читаются так: если имеет место $(C^{(p)}, C^{(s)})$, то при наличии субтекста f_w^i из множества возможных типов реакций R следует выбирать реакцию из класса r_k ; если имеет место субтекст t_v , то из класса r_k следует выбирать конкретную реакцию r_k' .

Помимо предписывающих, возможно использование рекомендуемых (например, если t_w^1 , то лучше r_1 , но можно r_2 или r_3) и запрещающих корреляционных правил (например, если t_w^2 , то недопустимы r_1, r_2, \dots, r_N), а также их всевозможных комбинаций.

Из этих правил формируется процедурная база знаний, которая после заполнения в режиме обучения, может пополняться и корректироваться в процессе эксплуатации ИБЗ.

1.4. ТИПОВАЯ СТРУКТУРА ИНТЕЛЛЕКТУАЛЬНОЙ БАЗЫ ЗНАНИЙ

Если рассматривать ИБЗ как систему, обладающую свойствами расчлененности, целостности, связанности и неадитивности, то в своем типовом варианте она должна состоять из трех основных компонентов: технического; информационного и операционно-лингвистического, а также включать в себя комплекс обеспечивающих подсистем (рис. 1.6). Рассмотрим содержание этих компонентов.

Технический компонент ИБЗ – это комплекс средств автоматизации, на котором реализуется данная база знаний, включающий: компьютеры, принтеры, сканеры, средства защиты от несанкционированного доступа и утечки информации, сетевые устройства и другие элементы.

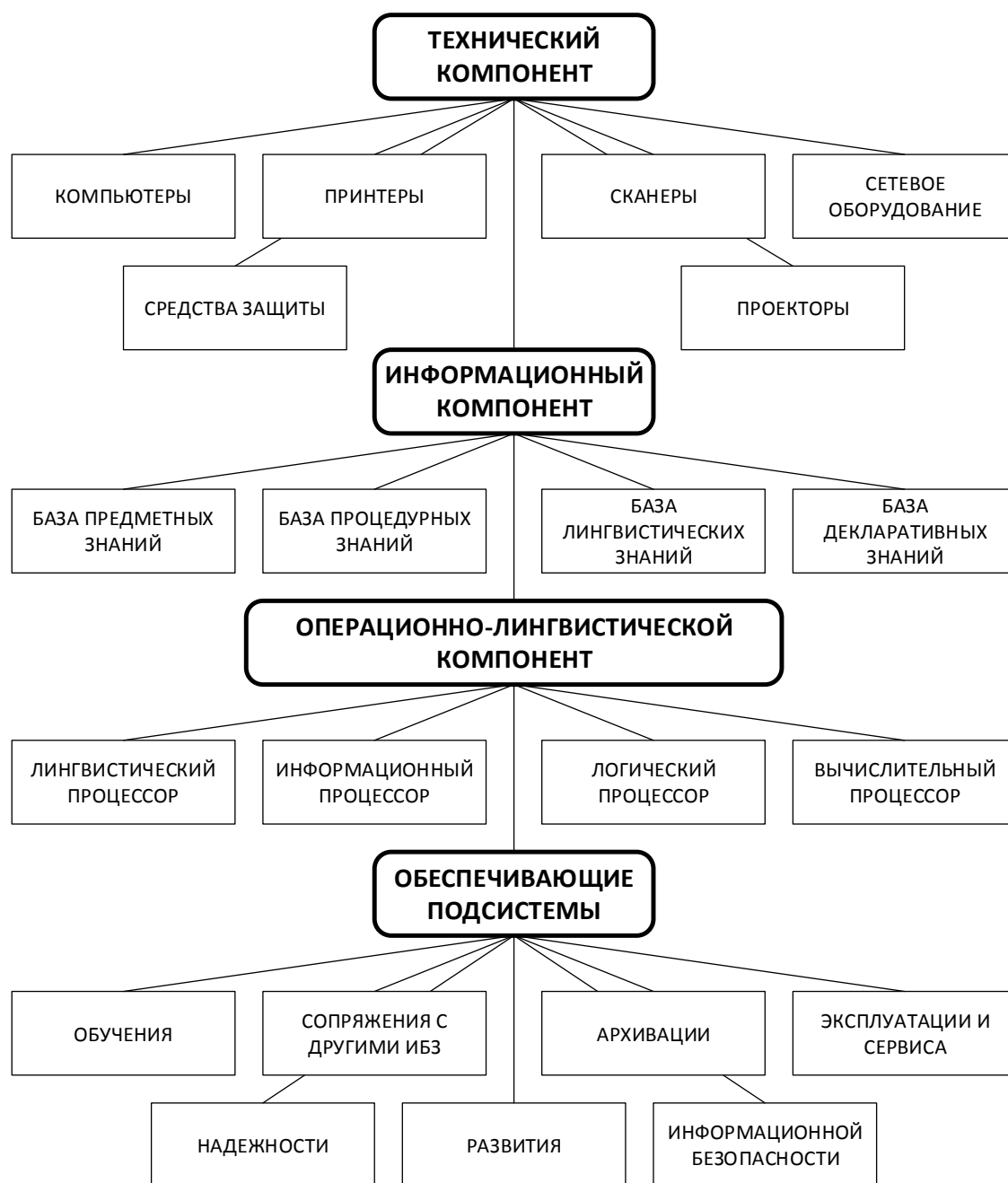


Рис. 1.6. Типовая структура интеллектуальной базы знаний

Информационный компонент представляет собой упорядоченные сведения о внешнем и внутреннем мире ИБЗ. Он состоит из баз предметных, процедурных знаний, декларативных и лингвистических знаний.

База предметных знаний, отражая проблемную среду, внешний и внутренний мир ИБЗ, дифференцируется на пользовательский и прагматический уровни. Пользовательский уровень включает: словари терминов, описания используемых переменных и накладываемых на них ограничений, возможные критерии выбора решений, а также базовые

аксиомы – утверждения, которые априори считаются истинными и из которых по определенным правилам выводятся новые утверждения. Прагматический уровень содержит данные о характеристиках информационных объектов и используемой программной среде.

Фактически, этот уровень представляет собой базу данных в ее традиционном понимании, построение которой можно осуществлять с использованием СУБД Oracle, относящейся к разряду современных многопользовательских СУБД клиент-серверной архитектуры и ориентированной на работу с интегрированными базами данных.

База процедурных знаний содержит совокупность правил, определяющих порядок и способы применения предметных и декларативных знаний для ответа на вопрос, поставленный пользователем. Ее центральным компонентом является дерево целей и задач, определяющее алгоритм поиска ответа на запрос пользователя, а также правила вывода, в совокупности позволяющие формировать алгоритм поиска ответа методом резолюций:

Если **«ЗАПРОС ПОЛЬЗОВАТЕЛЯ»**, то **«ОТВЕТ ИБЗ»**.

Базу декларативных знаний образуют правила вывода, на основе которых ИБЗ делает обобщения и заключения. Она имеет также два уровня: пользовательский и прагматический. Пользовательский уровень содержит набор сценариев для анализа запроса пользователя, а также набор типовых ответов, привязанных к этим запросам:

«ЗАПРОС ПОЛЬЗОВАТЕЛЯ» → «ТИПОВОЙ ОТВЕТ ИБЗ».

Прагматический уровень содержит прикладные программные модули, предназначенные для проведения статистических расчетов и вычисления обобщенных показателей судебной практики и судебных решений:

**«ЗАПРОС ПОЛЬЗОВАТЕЛЯ» → «АКТИВИЗАЦИЯ
МОДУЛЯ» → «ПРОВЕДЕНИЕ РАСЧЕТОВ» → «ОТВЕТ ИБЗ»**.

База лингвистических знаний содержит правила морфологического, синтаксического и семантического анализа входных и выходных текстов, а также списки основ слов, которые используются для организации диалога.

Операционно-лингвистический компонент ИБЗ представляет собой структурированную совокупность программ, выполняющих процедуры анализа и синтеза текстов естественного языка, запоминания, забывания и извлечения информации, ее интерпретации, производства логических выводов и принятия решений, а также планирования и оптимизации вычислительного процесса. Операционный компонент, как правило, состоит из четырех основных блоков, которые по сложившейся традиции называют процессорами.

Лингвистический процессор – это комплекс программ, осуществляющих перевод входных текстов естественного языка на язык пред-

ставления знаний и обратный перевод. Перевод осуществляется в три этапа. На первом этапе производится морфологический анализ входного предложения с целью идентификации образующих его слов с терминами, используемыми в модели. Для этого используются списки основ слов (слова за вычетом аффиксов – суффиксов и окончаний), которые хранятся в лингвистической базе знаний. Морфологический анализ не является обязательной операцией. В системных моделях, работающих с небольшим словарным запасом входного языка, используют функциональную лексику, когда каждой фразе присваивается определенный символ, который однозначно воспринимается лингвистическим процессором. На втором этапе производится синтаксический анализ входной фразы с целью получения формализованной записи синтаксической структуры входного выражения. Для проведения анализа используются синтаксические правила, хранящиеся в лингвистической базе знаний, которые позволяют идентифицировать структуру входной фразы со словарем ИБЗ. На третьем этапе производится семантический анализ входной фразы, то есть идентификация ее значения со знаниями, хранящимися в предметной и декларативной базах. При нахождении идентичной конструкции входная фраза считается понятной, и лингвистический процессор работу заканчивает. Если такой конструкции в указанных базах знаний не содержится, то лингвистический процессор инициирует диалог с пользователем для пополнения или изменения знаний. Лингвистические процессоры обычно строятся таким образом, чтобы они удовлетворяли принципу обратимости. Суть этого принципа состоит в том, что все процедуры, обеспечивающие перевод фраз естественного языка в выражения языка представления знаний, должны быть обратимыми. Другими словами, если выходы этих процедур считать их входами и все операции заменить обратными, то те же самые процедуры можно использовать для обратного перевода.

Информационный процессор – это комплекс программ, осуществляющих операции запоминания, забывания и извлечения информации о фактах внешнего и внутреннего миров ИБЗ. Знания на пользовательском уровне обрабатываются с помощью классификатора понятий, который одновременно участвует в идентификации конструкций входных фраз. Кроме того, информационный процессор реализует функции управления базой данных.

Логический процессор – это комплекс программ, осуществляющих классификацию и обобщение текущих ситуаций в объекте исследования, логический поиск рациональных решений и их координацию. Логический процессор участвует в идентификации ситуаций, описываемых входной фразой, с типовыми ситуациями, хранящимися в базе декларативных знаний. При успешной идентификации найденная типовая ситуация передается в решатель, который отыскивает среди эталонных

решений (хранящихся в декларативной базе) наилучшее решение для идентифицированной ситуации. Если идентификация ситуации невозможна, производится ее расчленение на более простые, которые могут быть сведены к типовым ситуациям, описанным в базе знаний, либо, напротив, ее обобщение с той же целью. Если не удастся выделить типовую ситуацию, то процессор формирует новую ситуацию и передает ее для запоминания в базу знаний.

Вычислительный процессор – это комплекс программ, планирующих и выполняющих различные вычисления. В типовом варианте он состоит из трех программных модулей: анализатора, планировщика и программатора. Анализатор на основании описания вычислительной задачи определяет, какие прикладные программные модули необходимо привлечь для ее решения. Кроме того, анализатор определяет состав исходной информации и аппаратных средств, которые должны участвовать в формировании ответа. После этого планировщик составляет план вычислительного или логического процесса, оптимальный относительно некоторого критерия, например, минимума времени производства вычислений. Завершает работу вычислительного процессора программатор, который на основании плана формирует программу вычислений и следит за ходом ее реализации. После завершения программы полученная информация передается в лингвистический процессор, а оттуда – пользователю.

Помимо указанных выше компонентов в состав ИБЗ могут входить различные **обеспечивающие подсистемы**, в частности такие как:

- подсистема обучения кадров для подготовки и переподготовки специалистов и пользователей ИБЗ, проведения сертификации специалистов;
- подсистема сопряжения для обеспечения информационного взаимодействия со сторонними объектами (другими ИБЗ, автоматизированными системами управления, экспертными системами и т.п.);
- подсистема архивации для ведения и использования рабочих архивов;
- подсистема обеспечения эксплуатации и сервисного обслуживания для поддержки процессов технического и сервисного обслуживания комплексов средств автоматизации;
- подсистема обеспечения надежности функционирования в критических ситуациях, в частности при отключении энергопитания или нарушения связи;
- подсистема развития, обеспечивающая изменение конфигурации ИБЗ без переделки программно-технических комплексов;
- подсистема обеспечения информационной безопасности для защиты конфиденциальной информации от несанкционированного доступа, контроля целостности и подлинности электронных документов,

обеспечения безопасного межсетевого взаимодействия, антивирусной защиты.

1.5. РЕЖИМЫ РАБОТЫ ИНТЕЛЛЕКТУАЛЬНОЙ БАЗЫ ЗНАНИЙ

Принципиальная особенность ИБЗ как системного информационного объекта состоит в необходимости его обучения. Исходный вариант ИБЗ содержит лишь операционный компонент, который реализует «интеллектуальные» механизмы восприятия, представления и выдачи информации по запросу пользователя. Информационный же компонент системы пуст. Для того чтобы работать с ИБЗ, надо сообщить ей факты и закономерности, касающиеся предметной области. В качестве учителя, как правило, выступает пользователь, который через лингвистический процессор сообщает ИБЗ все необходимые знания. В режиме обучения информационный и логический процессоры работают только на прием информации, а вычислительный процессор вообще не участвует в работе.

После обучения ИБЗ может работать в вопросно-ответном и интерактивном режиме.

В вопросно-ответном режиме пользователь формулирует и вводит в ИБЗ текст, содержащий тему и условия вопроса. Выходом являются сведения, удовлетворяющие условиям вопроса и оформленные в виде некоторого текста на естественном языке либо в виде документа (таблицы, анкеты).

В интерактивном режиме участвуют все компоненты ИБЗ. В этом режиме на вход ИБЗ поступают описания сценариев, содержащие сведения о целях и задачах данного эксперимента, условиях и графике его проведения и т.д. ИБЗ инициирует диалог с пользователем до тех пор, пока не получит от него ответы, снимающие всякую неопределенность относительно выполнения предписанных операций. После снятия вопросов производятся необходимые расчеты, логические выводы и другие операции, позволяющие сформировать на выходе модели рекомендации по изменению параметров объекта. Эта информация в виде таблиц и графиков, сопровождаемых текстами на естественном языке, выдается исследователю. После анализа полученной информации пользователь вводит в ИБЗ необходимые корректировки, изменяет условия задачи, и модельный эксперимент продолжается.

2. ЯЗЫКОВЫЕ СРЕДСТВА ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

В проблеме представления знаний много трудных вопросов, конструктивные ответы на которые до сих пор не получены. Среди них кардинальный вопрос о структуре самого языка представления знаний. На вопросы о том, как и в какой языковой форме хранятся у человека знания о внешнем мире, прошлом опыте и своих возможностях, каким образом осуществляется обработка знаний, как на базе старых происходит формирование новых знаний, современная наука пока не дает исчерпывающих ответов. Есть только гипотезы и предположения. Это вынуждает разработчиков ИБЗ использовать паллиативный подход к решению проблемы представления знаний. Сущность этого подхода заключается в том, что язык для представления знаний рассматривается как инструментальное средство, аналогичное естественному языку, с тем существенным отличием, что в нем алгоритмы выявления смысла сообщений и текстов зафиксированы в более строгой (естественно, и более ограниченной) форме по сравнению с естественным языком.

В силу такого подхода выбор языков представления знаний при принятии проектных решений по созданию ИБЗ всегда сталкивается с двумя противоречивыми требованиями. С одной стороны, требуется, чтобы такой язык содержал развитую систему формальных эквивалентных преобразований, обеспечивающую возможность определения синтаксической и семантической правильности выражений и формального построения выводов. С другой стороны, такой язык должен позволять адекватно описывать ситуации в проблемной области во всем их многообразии, то есть обеспечивать некоторое отображение естественного языка на формально-логический язык.

Эта двойственность приводит к тому, что выбор того или иного варианта языка представления знаний и соответственно метода представления знаний фактически сводится к поиску компромисса между указанными выше требованиями.

2.1. ОБЗОР ЯЗЫКОВ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

В настоящее время при создании ИБЗ наибольший практический интерес представляют следующие языки: язык Кодда; логические языки; язык нечетких множеств; контекстно-свободный плекс-язык; язык RX-кодов; язык семантических сетей; язык ролевых фреймов; тензорный язык Крона.

Дадим этим языкам краткую характеристику с целью выявления их достоинств и недостатков с точки зрения использования при выборе проектных решений по созданию ИБЗ.

Язык Кодда. С точки зрения записи и обработки знаний этот язык представляет сегодня скорее исторический, нежели теоретический интерес, но на практике он используется, и довольно широко, выступая базисом для построения различных баз данных (типа *Microsoft Office Access*) и развития языков более высокого уровня. В этом языке данные о предметной области формально задаются на множестве

$$H = I \times H_1 \times \dots \times H_k, \quad (2.1)$$

где H_1 – множества значений атрибутов j_1 , а I – множество имен сущностей, данных в виде 1-арных отношений ($1 \leq k$).

На заре информатизации язык Кодда с табличной формой представления информации составлял теоретическую основу почти всех методов разработки всевозможных баз данных. Однако с переходом к созданию баз знаний выявилась его ограниченность, выражающаяся в том, что конструкции вида (2.1) позволяют записывать и обрабатывать только данные, но не знания.

Логические языки. Эти языки образуют достаточно большой класс формализованных языковых средств пригодных для представления знаний, наибольшее практическое применение из которых нашли язык предикатов первого порядка и язык исчисления высказываний. Для этих языков существуют достаточно мощные процедуры эквивалентных преобразований (правила дистрибутивности, идемпотентности и т.п.) и получения новых утверждений и фактов, самой известной из которых является процедура резолюции «если..., то...». Однако с точки зрения проектирования ИБЗ их описательные возможности ограничены:

- в логических языках невозможно выразить многие квантификаторы естественного языка (например, «часто», «сильно», «почти всегда»), а так же модальности (в частности такие как «желательно», «необходимо») и императивы («иди», «сделай» и т.п.), присущие естественно-языковым текстам;

- набор текстов на естественном языке, для которых имеется полная аналогия с логическим представлением, резко сужается из-за необходимости точного указания области действия предиката или высказывания.

Из этого следует, что логические языки из-за «жесткой» семантики не обладают достаточными описательными возможностями для передачи всех оттенков и тонкостей смысла естественного языка. Вместе с тем мощность процедур эквивалентных преобразований обуславливает их широкое применение в узко ориентированных базах знаний, входной язык которых является небольшим подмножеством естественного языка.

Разновидностью логических языковых средств представления знаний является язык программирования Пролог (от фр. *programmation*

en Logique), построенный на исчислении предикатов первого порядка. Основными понятиями в этом языке являются факты, правила логического вывода и запросы, позволяющие описывать факты, процедуры логического вывода и принятия решений. Факты в языке Пролог описываются логическими предикатами с конкретными значениями. Правила записываются в форме правил логического вывода с логическими заключениями и списком логических условий. Особую роль в интерпретаторе Пролога играют конкретные запросы к базам знаний, на которые система логического программирования генерирует ответы «истина» и «ложь». Для обобщённых запросов с переменными в качестве аргументов система Пролог выводит конкретные данные в подтверждение истинности обобщённых сведений и правил вывода.

Будучи декларативным языком программирования, Пролог воспринимает в качестве программы некоторое описание задачи или баз знаний и сам производит логический вывод, а также поиск решения задач, пользуясь механизмом поиска с возвратом и унификацией. В настоящее время Пролог, несмотря на неоднократные пессимистические прогнозы, продолжает развиваться и вбирать в себя новые технологии и концепции, а также парадигмы императивного программирования. В частности, одно из направлений развития языка Пролог реализует концепцию многоагентного проектирования баз знаний.

Язык нечетких множеств. Этот язык представляет собой естественное расширение классического теоретико-множественного языка, построенного на дихотомии (от греч. *dichotomía* – разделение надвое), то есть элемент « x » может либо принадлежать множеству A , либо не принадлежать ему. В языке нечетких множеств дихотомия отсутствует. Вместо нее принадлежность элемента « x » множеству A задается функцией принадлежности $\mu_A(x)$.

Нечетким множеством A на множестве X называется множество упорядоченных пар $A = \{\mu_A(x), x\}$, составленных из элементов x универсального (полного) множества X , и соответствующих функций принадлежности $\mu_A(x)$. Переменная x называется базовой. Обычно на нечеткое множество ссылаются либо по его имени A , либо по функции принадлежности.

Если универсальное множество X состоит из конечного числа элементов x_1, x_2, \dots, x_n , то нечеткое множество A можно представить в следующем виде:

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n = \sum_{i=1}^n \mu_A(x_i)/x_i. \quad (2.2)$$

В данном случае знак « $+$ » не есть сложение в его традиционном понимании, им обозначается совокупность элементов множества (числитель) с их принадлежностью (знаменатель).

Функция принадлежности может выражаться как числами из интервала $[0, 1]$, так и в виде лингвистических переменных, то есть переменных, значениями которых выступают не числа, а слова и словосочетания. Например, лингвистическая переменная O = «ошибка» может иметь значения $O_{об}$ = «отрицательно большая», $O_{ом}$ = «отрицательно малая», O_n = «нуль», $O_{пм}$ = «положительно малая», $O_{пб}$ = «положительно большая».

Таким образом, по определению нечеткие множества представляют собой расширение обычных (строгих или четких) множеств, которые есть ни что иное как частный случай нечетких множеств.

Содержательная трактовка функции принадлежности зависит от задачи, в которой используется нечеткое множество. Возможные трактовки функции принадлежности: степень соответствия понятию A ; вероятность; возможность; полезность; истинность; правдоподобность; значение функции.

Для каждой трактовки функции принадлежности разрабатываются свои методы их построения. В ряде моделей мягких вычислений функции принадлежности задаются в параметрическом виде. Например, функции принадлежности нечетких множеств $O_{об}$, $O_{ом}$, O_n , $O_{пм}$, $O_{пб}$ могут изначально задаваться в модели так, чтобы они равномерно покрывали область определения X , а затем настраивались в результате изменения их параметров в процессе отладки модели.

В случае, когда функция принадлежности задается числами, наиболее распространенными в приложениях являются линейные, треугольные, трапециевидные, гауссовы и колоколообразные функции принадлежности.

Язык нечетких множеств имеет достаточно развитую систему операций для формальных эквивалентных преобразований, сравнимую по мощности с аналогичной системой операций традиционных логических языков.

Простейшей является операция дополнения нечеткого множества, выступающая аналогом связки «не» (отрицания) в математической логике: $\forall x \in X: \bar{\mu}(x) = 1 - \mu(x)$.

Следующей является операция разности нечетких множеств: $(\mu_1 - \mu_2)(x) = \max \{0, \mu_1(x) - \mu_2(x)\}$, где индекс 1 относится к уменьшаемому, а индекс 2 – к вычитаемому.

Объединение нечетких множеств – максимум (аналог логическому «или» как «либо-либо»), записывается так: $(\mu_1 \cup \mu_2)(x) = \max \{\mu_1(x), \mu_2(x)\}$.

Пересечение нечетких множеств (минимум, аналог логическому «и» как «и-и») записывается так: $(\mu_1 \cap \mu_2)(x) = \min \{\mu_1(x), \mu_2(x)\}$.

Ограниченная сумма нечетких множеств (логическое «или»): $(\mu_1 \oplus \mu_2)(x) = \min \{1, \mu_1(x) + \mu_2(x)\}$.

Ограниченное произведение нечетких множеств (логическое «и»): $(\mu_1 \otimes \mu_2)(x) = \max \{0, \mu_1(x) + \mu_2(x)\}$.

Алгебраическое произведение нечетких множеств: $(\mu_1 \bullet \mu_2)(x) = \mu_1(x) \cdot \mu_2(x)$.

Алгебраическая сумма нечетких множеств (логическое «или»): $(\mu_1 + \mu_2)(x) = \{\mu_1(x) + \mu_2(x)\} - (\mu_1 \bullet \mu_2)(x)$.

Операция концентрирования, для гауссовой функции принадлежности имеющая вид: $\mu(x, \vartheta) = \exp[-0,5(x - a)^2/\vartheta b^2]$, где $\vartheta = (0, \dots, 1)$ – коэффициент концентрирования.

Операции над нечеткими множествами можно интерпретировать простейшим абстрактным автоматом – преобразователем информации, который выдает определенный выходной сигнал в ответ на каждый входной сигнал. В качестве примера на рис. 2.1 изображен абстрактный

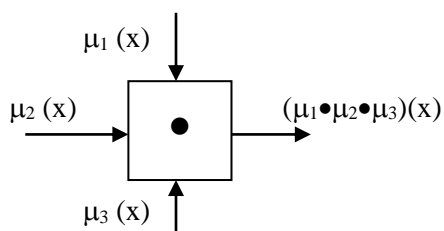


Рис. 2.1. Пример автоматного представления алгебраического произведения нечетких множеств

автомат, имеющий три входа и один выход и соответствующий операции алгебраического произведения трех нечетких множеств $\mu_1(x)$, $\mu_2(x)$ и $\mu_3(x)$. Работает такой автомат просто: в каждый момент t_i дискретного времени он либо принимает входные сигналы $\mu(x)$, либо порождает выходной сигнал $(\mu_1 \bullet \mu_2 \bullet \mu_3) = \mu_1(x) \cdot \mu_2(x) \cdot \mu_3(x)$. Автоматное представление нечетких мно-

жеств удобно для составления нечетких алгоритмов.

Нечеткими алгоритмами называются упорядоченные совокупности операций над функциями принадлежности нечетких множеств. Они описывают сложные отношения между нечеткими множествами и бывают с фиксированной и нефиксированной структурой. Алгоритмом с фиксированной структурой однозначно предписывается последовательность выполнения элементарных операций над функциями принадлежности нечетких множеств, выступающими исходными данными.

Пример нечеткого алгоритма «А» с фиксированной структурой приведен на рис. 2.2, а его формальная запись представляется в следующем виде: $\mu(A)(x) = \min \{1, \min [\mu_1(x), \mu_2(x)] + \max [0, \mu_1(x) - \mu_3(x)]\}$. При заданных функциях принадлежности $\mu_1(x)$, $\mu_2(x)$ и $\mu_3(x)$ такая запись однозначно определяет порядок выполнения элементарных операций и позволяет установить совокупное нечеткое множество $\mu(A)(x)$. Алгоритмы с четко фиксированной структурой используются для представления в ИБЗ детерминированных процессов.

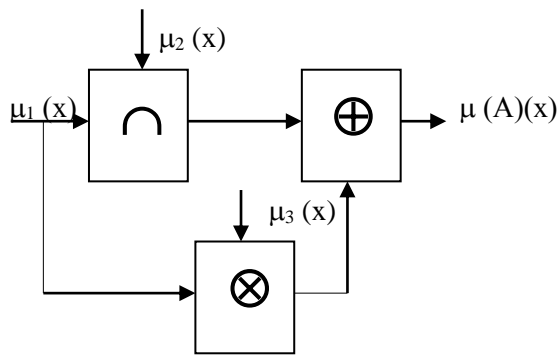


Рис. 2.2. Пример нечеткого алгоритма с фиксированной структурой

Простой условный оператор имеет вид, представленный на рис. 2.3, и формально записывается так: $O(x, \theta, \lambda): \lambda \Rightarrow \{[y_1 = \theta_1 \mu(x)], \dots, [y_N = \theta_N \mu(x)]\}, \sum \theta_i = 1$. Он

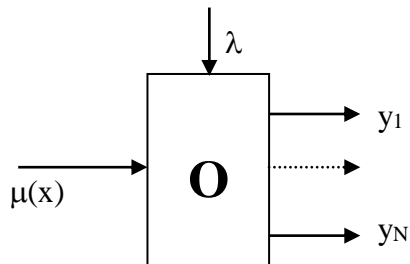


Рис. 2.3. Простой условный оператор

имеет тот смысл, что если выполняется некоторое условие λ , то на выходе i ($i = 1, \dots, N$) вырабатывается сигнал $y_i = \mu(x)$ с некоторой вероятностью θ_i от 0 до 1. Если условие λ не выполняется, то на всех выходах сигнал отсутствует, $y_1 = \dots = y_N = 0$. В частном случае, когда λ принимает только два значения, например 0 и 1, простой условный оператор будет де-

терминированным.

Условные операторы с обратной связью используются при разработке алгоритмов, представляющих в ИБЗ адаптивные и самоорганизующиеся процессы. На рис. 2.4 приведен один из возможных вариантов такого алгоритма, реализующего обратную связь типа ограниченного произведения \otimes по выходу y_N . Изменяя конфигурацию такого алгоритма и варьируя операциями, образующими контур обратной связи, можно имитировать функционирование систем как с положительными, так и с отрицательными обратными связями (детерминированными, случайными, запаздывающими и другими).

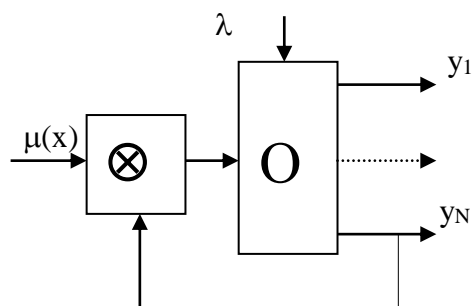


Рис. 2.4. Условный оператор с обратной связью

Метод представления знаний с использованием нечетких алгоритмов образно можно сравнить с построением различных фигур из «кубиков» конструктора. Операция в принципе простая, но для того чтобы в итоге конструирования получилась требуемая конфигурация, необходимо: а) иметь конструкторскую схему, то есть достаточно полную понятийную (вербальную) модель объекта; б) располагать полным набором совмещаемых между собой «кубиков», то есть задать функции принадлежности нечетких множеств, которые будут использоваться при конструировании алгоритма.

Зачастую выполнить эти условия не представляется возможным именно потому, что нет полной ясности относительно свойств и возможных вариантов поведения объекта предметной области. В таких случаях организуется пошаговая разработка алгоритма и его поэтапная апробация с «обучением». Учителем выступает пользователь ИБЗ, а обучение заключается в уточнении и видоизменении первоначально выбранных функций принадлежности, а также структуры алгоритма (используемых операторов, вариантов их соединения и т.п.). Процесс обучения может завершиться тем, что построенный алгоритм будет неадекватным реальному процессу. Отрицательный результат – тоже результат, добавляющий знания. Во всяком случае, даже неудачные попытки представления в ИБЗ объектов предметной области с помощью нечетких алгоритмов полезнее, чем поверхностные наблюдения, сомнительные аналогии и умозрительные заключения.

Контекстно-свободный плекс-язык. Основными компонентами этого языка являются: плекс-элементы, операция канкатенации и КСП-грамматика. Плекс-элемент – это абстрактный объект, имеющий определенное количество контактов (входов и выходов) для соединения с

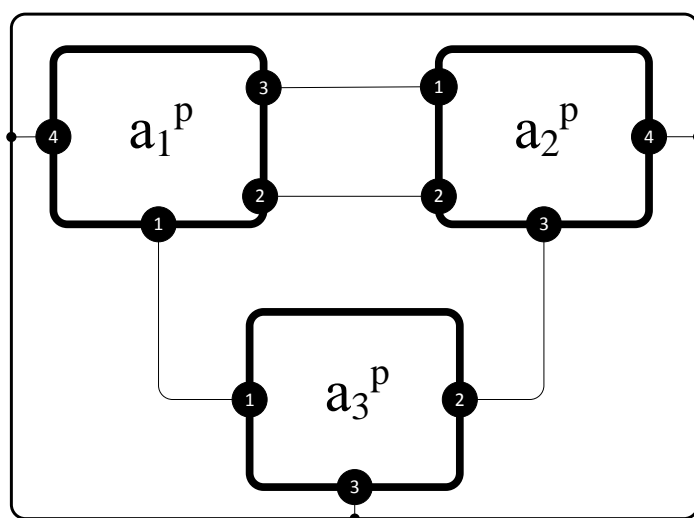


Рис. 2.5. Представление трехкомпонентной системы с помощью КСПЯ

другими плекс-элементами. Множество различающихся плекс-элементов, из которых может быть образована некоторая структура, образуют алфавит КСПЯ. Конкатенацией называется соединение плекс-элементов (их контактов) из алфавита, что формально задается матрицей $\tilde{A} = \|\gamma_{ij}\|$, элементы которой отражают связи между контактами

плекс-элементов. КСП-грамматика – это правила соединения плекс-элементов между собой.

В качестве иллюстративного примера на рис. 2.5 дано графическое представление некоторой системы, описанной на КСПЯ. Эта система состоит из трех компонентов или в терминах КСПЯ – трех плекс-элементов: a_1^p , a_2^p , и a_3^p . В каждом плекс-элементе контакты выделены точками на его контуре и перенумерованы. При этом часть контактов замкнута на контакты других плекс-элементов (внутренность системы), а другая часть контактов замкнута на контакты внешнего контура поверхности системы. Матрица конкатенации для этого плекса (системы) представляется следующим образом:

$$\tilde{A} = \begin{vmatrix} 0 & (2-2,3-1) & (1-1) \\ (2-2,1-3) & 0 & (3-2) \\ (1-1) & (2-3) & 0 \end{vmatrix}. \quad (2.3)$$

Элемент γ_{ij} , расположенный над главной диагональю матрицы, отличается от элемента γ_{ji} , стоящего ниже диагонали, только порядком записи индексов для каждой пары, что отражает направленность связей между плекс-элементами.

Грамматикой КСПЯ называется четверка:

$$G = \langle v_0, A, V, R^* \rangle, \quad (2.4)$$

где v_0 – начальный символ; $A = \{a_i^p\}$ – алфавит терминальных плекс-элементов; $V = \{v_k\}$ – алфавит вспомогательных плекс-элементов; R^* – множество правил вывода.

Правила вывода имеют вид

$$v \rightarrow \Psi PI, \quad (2.5)$$

где Ψ ($\Psi \subseteq a_i^p \times v_k$) – подстановка, указывающая порядок конкатенации контактов при подстановке вспомогательного плекса вместо терминального; символика $PI(A \vee V)^+$ означает, что плекс PI есть подмножество объединения алфавитов A и V , значок «+» указывает на то, что при таком объединении должны учитываться все возможные в конкретной грамматике индексы.

Иллюстрация сказанного дается рис. 2.6 для грамматики со следующими правилами вывода:

$$\begin{aligned} v_0 &\rightarrow \Psi_1 \Gamma_1 v_1 v_2; \\ v_1 &\rightarrow \Psi_3 \Gamma_3 a_1^p a_2^p a_3^p; \\ v_2 &\rightarrow \Psi_6 \Gamma_6 a_4^p a_5^p. \end{aligned} \quad (2.6)$$

Эти правила построены на двух вспомогательных плекс-элементах v_1 и v_2 , пяти терминальных плекс-элементах $a_1^p, a_2^p, a_3^p, a_4^p, a_5^p$ при подстановках Ψ_1, Ψ_3, Ψ_6 . Начальный символ v_0 в этой грамматике представляет некоторый объект предметной области (систему) в целом, то есть на нулевом уровне членения. Вспомогательные плексы v_1 и v_2 заданы, например, альтернативными графами, вершинами которых служат функции, представляющие подпроцессы функционирования некоторого объекта (системы) как целого, а ребрами – отношения между функциями. Заданы и терминальные плексы $a_1^p, a_2^p, a_3^p, a_4^p, a_5^p$ (ими могут быть также альтернативные графы).

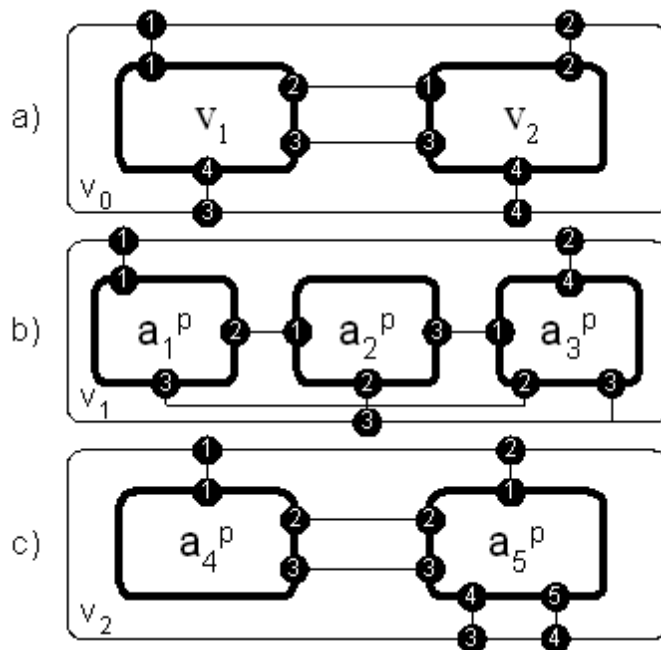


Рис. 2.6. Иллюстративный пример представления объекта с помощью КСП-грамматики

ми – отношения между функциями. Заданы и терминальные плексы $a_1^p, a_2^p, a_3^p, a_4^p, a_5^p$ (ими могут быть также альтернативные графы).

На рис. 2.6b и 2.6c показаны альтернативные варианты заданной через v_0 структуры объекта (системы) с применением различных комбинаций терминальных плекс-элементов.

Таким образом символика (2.6) означает (на примере первой строки) «плекс v_0 образован плексами v_1 и v_2

по правилам подстановки Ψ_1 и матрицы конкатенации Γ_1 ».

Матрицы конкатенации и подстановки в рассматриваемом примере выглядят так:

$$\tilde{A}_1 = \begin{vmatrix} 0 & (2-1, 3-3) \\ (1-2, 3-3) & 0 \end{vmatrix};$$

$$\Psi_1 = ((1-1, 4-3), (2-2, 4-4));$$

$$\tilde{A}_3 = \begin{vmatrix} 0 & (2-1) & (3-2) \\ (1-2) & 0 & (3-1) \\ (2-3) & (1-3) & 0 \end{vmatrix};$$

$$\Psi_3 = ((1-1), (2-3), (3-4, 4-2));$$

$$\tilde{A}_6 = \left\| \begin{array}{cc} 0 & (2-2, 3-3) \\ (2-2, 3-3) & 0 \end{array} \right\|;$$

$\Psi_6 = ((1-1), (1-2), (2-2), 4-3, 5-4))$.

В этой грамматике возможен вывод «r» такого вида:

$$r: v_0 \rightarrow \Psi_1 \Gamma_1 (\Psi_3 \Gamma_3 a_1^p a_2^p a_3^p) (\Psi_6 \Gamma_6 a_4^p a_5^p). \quad (2.7)$$

Этот вывод записан в так называемой неприведенной форме, позволяющей определять элементный состав плекса и, вместе с этим, содержащий указатели матриц конкатенации и подстановки, по которым можно выявить способы соединения плкс-элементов между собой. С точки зрения построения ИБЗ это означает, что, используя записи типа (2.7), можно строить дерево логических выводов.

Основным достоинством КСПЯ является возможность его использования как достаточно гибкой формы представления структур различных объектов в базах знаний. Причем, когда появляется некоторая новая структура, всегда можно установить, достаточно ли алфавита и грамматики данного языка для ее адекватного представления в базе знаний, или необходимо дополнять алфавит новыми плкс-элементами и модифицировать КСП-грамматику. Основные ограничения данного языка связаны со следующими обстоятельствами:

- в КСПЯ практически отсутствуют средства формальных эквивалентных преобразований, что не позволяет установить синтаксическую и семантическую правильность выражений и формального построения логических выводов (эти функции, важные для построения интеллектуальных баз знаний, как и при использовании текстов естественного языка, остается за человеком);

- с помощью КСПЯ достаточно трудно описывать свойства и качества реальных объектов предметной области, поскольку он ориентирован на описание преимущественно отношений «вход-выход»;

- пока отсутствуют компьютерные программы-трансляторы, обеспечивающие перевод текстов, написанных на КСПЯ, в машинные коды или на языки программирования высокого уровня.

Язык RX-кодов. Появление этого языка связано с развитием методов обработки знаний на компьютерах. Он явился результатом последовательных попыток усовершенствовать и расширить автоматизацию тех видов обработки информации, которые выполняются человеком. Базовыми единицами этого языка выступают отношения, которые обозначаются буквой R_i с различными нижними индексами, и понятия, обозначаемые буквой X_j^m с верхними и нижними индексами. В качестве правильных синтаксических выражений используются двойки $R_i X_j^m$, любые конъюнкции таких двоек и конъюнкции, получающиеся путем замены любого X в правильном синтаксическом выражении

правильным выражением. Знак конъюнкции, если это не приводит к путанице, обычно опускается, а для указания зоны действия отношения используются круглые скобки. Эти правила и есть синтаксис языка RX-кодов.

Поясним сказанное. Пусть множество понятий состоит из X_1 и X_2 , а множество отношений – их R_1 и R_2 . Тогда синтаксически правильными будут, например, выражения

$$(R_1 X_2), (R_2 X_2)(R_1 X_2)(R_1 X_1), \quad (2.8)$$

а также выражение

$$((R_2 X_2)(R_1 (R_2 X_1)(R_2 X_2)))(R_1 X_1), \quad (2.9)$$

полученное из конъюнкции $(R_2 X_2)(R_1 X_2)(R_1 X_1)$ заменой X_2 на $(R_2 X_1)(R_2 X_2)$.

Для интерпретации выражений языка RX-кодов всем понятиям и отношениям жестко предписываются некоторые семантические значения. В этом случае оказывается возможным производить вывод новых понятий через ранее определенные, которые для этого нового понятия выступают в качестве определяющих признаков.

Естественная иерархия, присущая RX-кодам, позволяет описывать знания о предметной области методом древовидных структур, однако одновременно накладывает существенные ограничения на их описательные возможности. Дело в том, что определяемое в этом языке понятие всегда является корнем дерева, а отношения между понятиями одного уровня не учитываются. Для сочленения одноуровневых понятий требуется введение дополнительных (внеязыковых) средств. Вместе с тем именно это ограничение позволяет весьма эффективно использовать язык RX-кодов при организации баз знаний, в которых родовидовые отношения играют определяющую роль. Для естественного языка древовидные конструкции не столь характерны. В нем отношения между понятиями имеют, как правило, сетевую структуру и содержат многочисленные циклы, в которых понятия определяются через самих себя. Поэтому при использовании RX-кодов для записи естественно-языковых текстов возникают определенные трудности. Фактически, RX-коды и построенные на их основе методы представления знаний не могут адекватно отражать тексты, написанные на естественном языке.

Язык семантических сетей. Понятие семантической сети основано на гипотезе, что человеческая память формируется через ассоциации между понятиями. Понятие «ассоциативная память» появилось еще во времена Аристотеля и вошло в информатику в связи с работами по использованию простых ассоциаций для представления значения слов в базе знаний. В настоящее время этот формализм всесторонне развит и используется для представления социально-экономических, управленческих, медицинских, военных и других видов знаний. Функ-

циональным базовым элементом семантической сети служит тройная структура $(x_i \text{ r } x_j)$, состоящая из двух «узлов» (x_i, x_j) и связывающей их «дуги» (r) . Каждый узел представляет некоторое понятие, а дуга – отношение между парами понятий. Можно считать, что каждая такая тройка выражает простой факт. Например, дом (x_1) стоит на (r_1) горе (x_2) . Дуга r имеет направленность, благодаря чему между понятиями «х» в рамках определенного факта выражается отношение «субъект – объект». Кроме того, любой из узлов может быть соединен с любым числом других узлов, в результате чего существует возможность формировать сеть фактов.

Формально семантические сети задаются тремя классами термов: понятиями $X = \{x_1, x_2, \dots, x_n\}$, именами $I = \{i_1, i_2, \dots, i_m\}$ и отношениями $R = \{r, r_1, r_2, \dots, r_k\}$. Отношение r – особое. Оно означает «иметь имя», остальные отношения подразделяются на падежные, характеристические, причинно-следственные, иерархические, временные и топологические.

Падежные отношения связывают предикат как основу действия, определяемого предложением, с остальными словами. Например, в предложении «вынуть мяч из корзины» слово «вынуть» – предикат. Им устанавливается предикатное отношение «вынимать» между двумя понятиями – «мяч» и «корзина».

Характеристические отношения связывают характеристику с характеризуемым объектом и характеристику со значением характеристики. Например, в предложении «температура воздуха 5°C », характеризуемым объектом является «воздух», характеристикой – «температура», а значением характеристики – « 5°C ». Они связаны характеристическими отношениями «быть характеристикой» и «иметь значение».

Причинно-следственные отношения связывают понятия, одно из которых является причиной, а другое следствием. Эти отношения выражаются импликацией вида: $A \rightarrow B$, которые читаются так: если «А», то «В», или из «А» следует «В».

Иерархические отношения указывают на то, что один объект является составной частью другого объекта или – одно понятие определяется через другие понятия. Эти отношения выражаются словами «принадлежать к классу», «быть частью» и т.п.

Временные отношения бывают двух типов: абсолютные и относительные. Абсолютные временные отношения устанавливаются между объектами и отрезками (точками) временной оси, а относительные – это связи «быть раньше», «одновременно», «быть позже» и другие, которые устанавливаются между объектами предметной сферы.

Топологические отношения связывают объекты с точками какой-либо системы координат либо указывают на их взаимное расположение: «быть впереди», «быть сзади», «располагаться левее» и так далее.

Синтаксически правильными выражениями в языке семантических сетей считаются тройки

$$(x_i \text{ r}_j \text{ x}_l), (x_i \text{ p } i_j), (r_i \text{ p } i_j), \quad (2.10)$$

где $x_i \in X$, $i_j \in I$, $p, r_i \in R$.

Правильное выражение может быть образовано конъюнкцией (\wedge) и дизъюнкцией (\vee) правильных выражений, а также операцией отрицания (\neg), которая применима как к элементам из R , так и к правильным выражениям. Кроме того, выражение, полученное заменой терма в правильном выражении правильным выражением, также является правильным.

При разумном выборе обозначений с помощью семантических сетей можно выразить очень сложные совокупности фактов. В отличие от RX-кодов семантические сети позволяют описывать не только постоянные отношения, присущие совокупности объектов, но и временные (ситуативные) отношения между объектами. Важная особенность языка семантических сетей заключается в его способности выводить новые понятия (обобщать ситуации) за счет выделения типовой структуры отношений. Эта особенность не только используется для описания ситуаций – она оказалась весьма продуктивной при ситуационном поиске решений. Постепенное обобщение описаний в языке семантических сетей приводит, в конце концов, к построению так называемого минимального описания с максимальной свободой в заполнении переменных конкретными понятиями и со свободными параметрами. Это – самое важное свойство семантических сетей, позволяющее использовать их при создании ИБЗ и разработке процедур поддержки проектных решений.

Язык ролевых фреймов. В этом языке понятия определяются совокупностью семантических (смысловых) ролей, которые выражают их сущность, и задаются строкой:

$$\langle i; p_1, p_2, \dots, p_m, p_{m+1}, p_{m+2}, \dots, p_n \rangle, \quad (2.11)$$

где i – имя понятия, p_1, p_2, \dots, p_m – обязательные роли, а $p_{m+1}, p_{m+2}, \dots, p_n$ – необязательные роли. Множество необязательных ролей может быть пустым.

Ролями в каждом фрейме выступают другие фреймы, называемые слотами, что позволяет конструировать иерархические фреймовые описания рекурсивного вида:

$$\langle i; p_1(\langle i; p^1_1(\langle i; p^2_1(\dots p^M_1) \dots \rangle), \dots, p_N(\langle i; p^1_N(\langle i; p^2_N(\dots p^M_N) \dots \rangle)) \rangle \rangle \rangle. \quad (2.12)$$

Рассмотрим в качестве примера ролевой фрейм, описывающий понятие «управление»:

⟨**управление**: субъект управления, задача подчиненным, порядок их взаимодействия, мероприятия по обеспечению, мероприятия по организации управления⟩.

Указанные роли в свою очередь, являются сложными понятиями, представляемыми фреймами следующего уровня. Так, например, понятие «задача подчиненным» представляется следующим фреймом:

⟨**задача подчиненным**: цель, время действия, объект воздействия, место действия, способ действия, необходимый ресурс⟩.

Основное достоинство такого представления знаний о предметной области заключается в том, что с помощью фреймов можно описывать информационные структуры произвольной сложности и достаточно адекватно отображать семантику моделируемых предметных областей. Кроме того, представление знаний в виде фреймов подразумевает довольно полную и строгую их формализацию. Вместе с тем весьма трудно (даже для заранее известной предметной области) выделить сами фреймы. Какими по форме и структуре они должны быть? Сколько их нужно для адекватного описания ситуации? Как фреймы должны связываться друг с другом и передаются ли их свойства при сцеплении в сеть? Эти и другие вопросы пока не нашли исчерпывающих ответов и решаются на основе опыта разработчиков и пользователя.

Тензорный язык Крона. Обычно специалисты в области информационных систем воспринимают тензоры как чисто абстрактные математические объекты. Однако тензоры Крона – это физические объекты, то есть объекты, отражающие знания о реальных процессах и системах. Обладая инвариантностью и экономичностью записи, универсальностью и наращиваемостью аппарата преобразований, тензорные описания, предложенные Кроном, позволяют эффективно осуществлять декомпозицию систем и описывать их динамику в различных координатах, и именно в этом заключается их привлекательность для построения баз знаний.

Определение 2.1. Составной объект, у которого каждый компонент представляет собой объект той же валентности, что и он сам, называется компаунд-объектом.

$Z =$

	a	b	c	d	e	f	g	h	i	j	k
p											
q											
r											
s											
t											
u											
v											
w											

Рис. 2.7

При этом исходный расчленяемый объект может описываться тензором или в частном случае n-матрицей. Таким же образом компоненты каждой части могут представляться тензорами и n-матрицами.

На рис. 2.7 представлен двухвалентный объект Z, заданный матрицей с фиксированными индексами a, b, c, d, e, f, g, h, i, j, k, разделенный горизонтальными и вертикальными

жирными линиями на 16 частей. Для представления его компаунд-объектом введены компаунд-индексы p, q, r, s по вертикали и t, u, v, w по горизонтали. На рис.2.8 компаунд-объект Z представлен в координатах p, q, r, s, t, u, v, w как объект, состоящий из 16 частей $Z_i, i = \overline{1,16}$. Когда работают с компаунд-объектами (компаунд-тензорами, компаунд-матрицами), в отличие от компаунд-индексов «прежние» индексы a, b, c, \dots для компаунд-объекта называют отдельными индексами. Каждый фиксированный или переменный компаунд-индекс может представлять несколько фиксированных или переменных отдельных индексов. Так, на рис. 2.8 компаунд-индекс t представляет собой отдельные индексы a, b, c .

$$Z = \begin{array}{c|cccc} & t & u & v & w \\ \hline p & Z_1 & Z_2 & Z_3 & Z_4 \\ q & Z_5 & Z_6 & Z_7 & Z_8 \\ r & Z_9 & Z_{10} & Z_{11} & Z_{12} \\ s & Z_{13} & Z_{14} & Z_{15} & Z_{16} \end{array}$$

Рис. 2.8

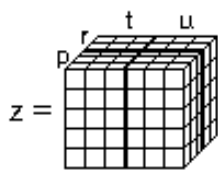


Рис. 2.9

На рис. 2.9 графически представлено расчленение трехвалентного объекта двумя ортогональными друг к другу вертикальными плоскостями. В этом случае сохраняются описанные для двухвалентного объекта правила разделения отдельных и компаунд-индексов (отдельные индексы на рисунке не показаны).

С компаунд-матрицей не связывается никакая система координат, тогда как у компаунд-тензора обязательно наличие фиксированных индексов. Если компаунд-матрица не имеет никакой формулы преобразования, то каждый компаунд-тензор имеют свою формулу преобразования. Отдельные индексы компонентов тензора преобразуются при помощи тензора преобразования $C_1 = C_{\alpha}^{\alpha'}$. Для преобразования компаунд-индексов того же тензора нужно пользоваться другим тензором преобразования $C_2 \neq C_1$.

Из приведенного видно, что компаунд-тензоры могут использоваться для представления некоторого объекта его составными частями. При этом привлекательным, с точки зрения представления и преобразования знаний о составе и структуре системы, является то, что над тензорными описаниями выполнимы операции, аналогичные арифметическим и алгебраическим операциям. Но такие операции имеют специфику. Основное правило выполнения операций над тензорами состоит в том, что никакие соотношения между тензорами недопустимы в разных точках пространства.

Простейшая такая операция – равенство тензоров, выполняема только для тензоров одинаковых типа и ранга в определенной точке (x^1, x^2, \dots, x^n) пространства. Тензоры одного типа и одного ранга A и B равны, если в этой точке равны соответствующие компоненты тензоров относительно некоторой координатной системы. Это записывается как

равенство соответствующих компонентов тензоров A и B в каждой координатной системе в таком виде

$$A_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r} (x^1, x^2, \dots, x^n) = B_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r} (x^1, x^2, \dots, x^n). \quad (2.13)$$

Равенство тензоров симметрично, рефлексивно и транзитивно.

Тензор, у которого все компоненты в любой координатной системе равны нулю, называется нуль-тензором.

$$A_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r} (x^1, x^2, \dots, x^n) = 0. \quad (2.14)$$

Сумма $C = A + B$ тензоров одного и того же типа и ранга есть тензор, компоненты которого в каждой координатной системе равны сумме соответствующих компонентов тензоров A и B

$$C_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r} = A_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r} + B_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r}. \quad (2.15)$$

Сложение векторов коммутативно и ассоциативно.

Произведением $B = \alpha A$ тензора A на скаляр α называется тензор, компоненты которого в каждой координатной системе равны произведениям компонентов тензора A на скаляр α

$$B_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r} = \alpha A_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r}. \quad (2.16)$$

Умножение тензора на скаляр коммутативно, ассоциативно и дистрибутивно относительно как тензоров, так и скаляров (например, $(-1)A = -A$ есть тензор, противоположный тензору A , то есть $A + (-A) = 0$).

Свертывание, применимое только к смешанным тензорам, определяется следующим образом. Если в заданном тензоре $A_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r}$ выбрать какой-нибудь верхний индекс и какой-нибудь нижний индекс и просуммировать все компоненты такого тензора с совпадающими значениями выбранных индексов, то полученные суммы будут компонентами нового тензора A , $s-1$ раз ковариантного и $r-1$ раз контравариантного. Кроме того, что тензор может быть свернут различными способами (выбором разных пар индексов), свертывание может быть повторено несколько раз (и каждое свертывание снижает на единицу ранг тензора).

Для тензоров различают внешнее и внутреннее произведения. Внешним произведением $C = AB$ двух тензоров A и B веса W и W' , соответственно, определяемых компонентами $A_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r}$ и $B_{k_1, k_2, \dots, k_q}^{k_1, k_2, \dots, k_p}$, называется тензор с компонентами

$$C_{i_1, i_2, \dots, i_s, k_1, k_2, \dots, k_q}^{i_1, i_2, \dots, i_r, k_1, k_2, \dots, k_p} = A_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r} B_{k_1, k_2, \dots, k_q}^{k_1, k_2, \dots, k_p}. \quad (2.17)$$

Частным случаем внешнего произведения является произведение тензоров. Произведение AB является тензором веса $W+W'$, $r+p$ раз кон-

травариантным и $s+q$ раз ковариантным. Умножение тензоров ассоциативно и дистрибутивно относительно сложения, но, в общем случае, не коммутативно в связи с тем, что порядок следования индексов менять нельзя. Примеры произведений: $a^i b^k = A^{ib}$; $a^i b_k = A_k^i$.

От произведения и внешнего произведения тензоров отличается внутреннее произведение двух тензоров. Оно определяется для таких двух тензоров, которые можно свернуть таким образом, что в каждом из слагаемых один или несколько верхних индексов компонента $A_{i_1, i_2, \dots, i_s}^{i_1, i_2, \dots, i_r}$ будут совпадать с одним или несколькими нижними индексами компонента $B_{k_1, k_2, \dots, k_q}^{k_1, k_2, \dots, k_p}$. В этом случае полученные суммы будут служить компонентами нового тензора, называемого внутренним произведением тензоров A и B . Примеры: $a^i b_i = \gamma$; $A_k^i a^k = c^i$.

Каждое внутреннее произведение тензоров A и B является тензором того же веса, что и AB . Ранг внутреннего произведения представляется разницей между рангом внешнего произведения AB и числом попарно взятых индексов, по которым производилось суммирование.

Выше представление о компаунд-тензоре формировалось на примере расчленения некоторого исходного тензора на части, имеющие ту же, что и исходный тензор, валентность. Нетрудно представить и обратный порядок формирования компаунд-тензора из нескольких тензоров той же валентности.

Когда речь идет о совокупности тензоров разной валентности, используется понятие мультитензора.

Определение 2.2. Тензор, содержащий два или более множества индексов, когда каждое множество индексов относится к различным множествам систем координат, называется мультитензором. При этом базовая буква тензора может иметь различное число индексов в различных координатах.

Например, $Z_{\alpha\beta}^{pq\gamma}$ является ковариантным тензором валентности два в α -координатах и контравариантным тензором валентности три в β -координатах. Соответственно, матрицы преобразования $C_{\alpha'}^{\alpha}$ и $C_{\beta}^{\beta'}$ относятся к разным группам. Важно иметь в виду, что основная буква может быть тензором в одних координатах и n -матрицей в других координатах. Так, например, $A_{\alpha(p)(q)}$ является вектором в α -координатах и, одновременно, 2-матрицей в β -координатах (заметим при этом, что α -координатная система может быть преобразована к α' с помощью $C_{\alpha'}^{\alpha}$, но закрытые индексы при этом не преобразуются).

Мультитензоры представляются так же, как и обычные тензоры,

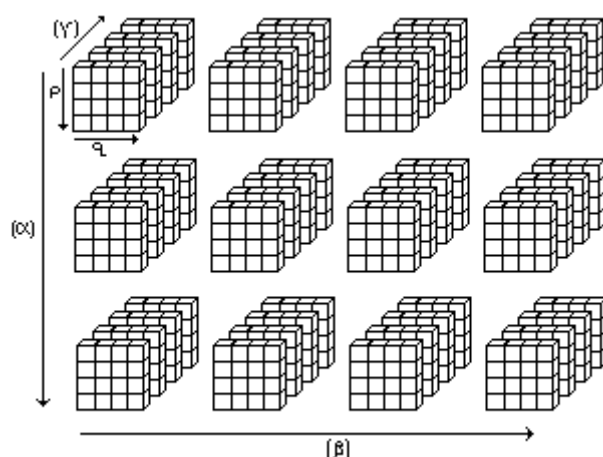


Рис. 2.10

но у них вдоль различных направлений число фиксированных индексов различно.

Например, если для $A^{\alpha pq}$ (см. рис. 2.10) существует семь осей в α -координатах и пять осей в p -координатах, то число компонентов $A^{\alpha pq}$ $7 \times 5 \times 5 = 175$. При этом, когда α -координаты временно не меняются (когда α - закрытый индекс, то есть $A^{(\alpha)pq}$), тогда тензор является

совокупностью семи 2-тензоров A^{pq} , расположенных в столбец, а когда индексы p и q рассматриваются как закрытые индексы ($A^{\alpha(p)(q)}$), тогда тензор – множество из $5^2 = 25$ векторов. Соответственно, на рис. 4 представлена совокупность k^3 2-тензоров $A^{\alpha\beta\gamma(p)(q)}$, а на рис. 2.11 – совокупность k^2 2-тензоров $A^{\alpha\beta\gamma(p)(q)}$.

Мультитензор можно расчленить на составляющие тензоры и прийти к компаунд-мультитензору. Достоинством мультитензоров и компаунд-мультитензоров (с точки зрения представления знаний) является то, что они обладают общностью описания и удобством выполнения промежуточных преобразований, число и сложность которых существенно сокращается по сравнению с традиционным описанием систем в основных переменных.

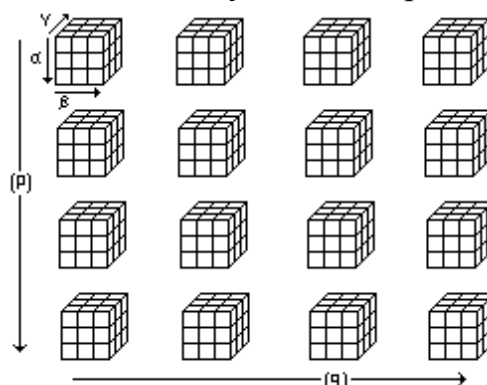


Рис. 2.11

Использование компаунд- и мультитензоров Крона можно рассматривать как перспективу развития методологии представления знаний в ИБЗ. Несомненным достоинством такого подхода является то, что тензоры обладают общностью описания и удобством выполнения промежуточных преобразований, число и сложность которых существенно сокращается по сравнению с традиционным описанием систем в основных переменных. Другое достоинство такого представления знаний заключается в том, что имеется возможность формального описания процесса функционирования систем в виде тензорных уравнений, решение которых дает возможность изучать и прогнозировать динамику развития систем.

2.2. СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ЯЗЫКОВ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

На рис. 2.12 приведена эвристическая диаграмма, отражающая результаты такого анализа. Язык Кодда исключен из рассмотрения, как непригодный для представления знаний.

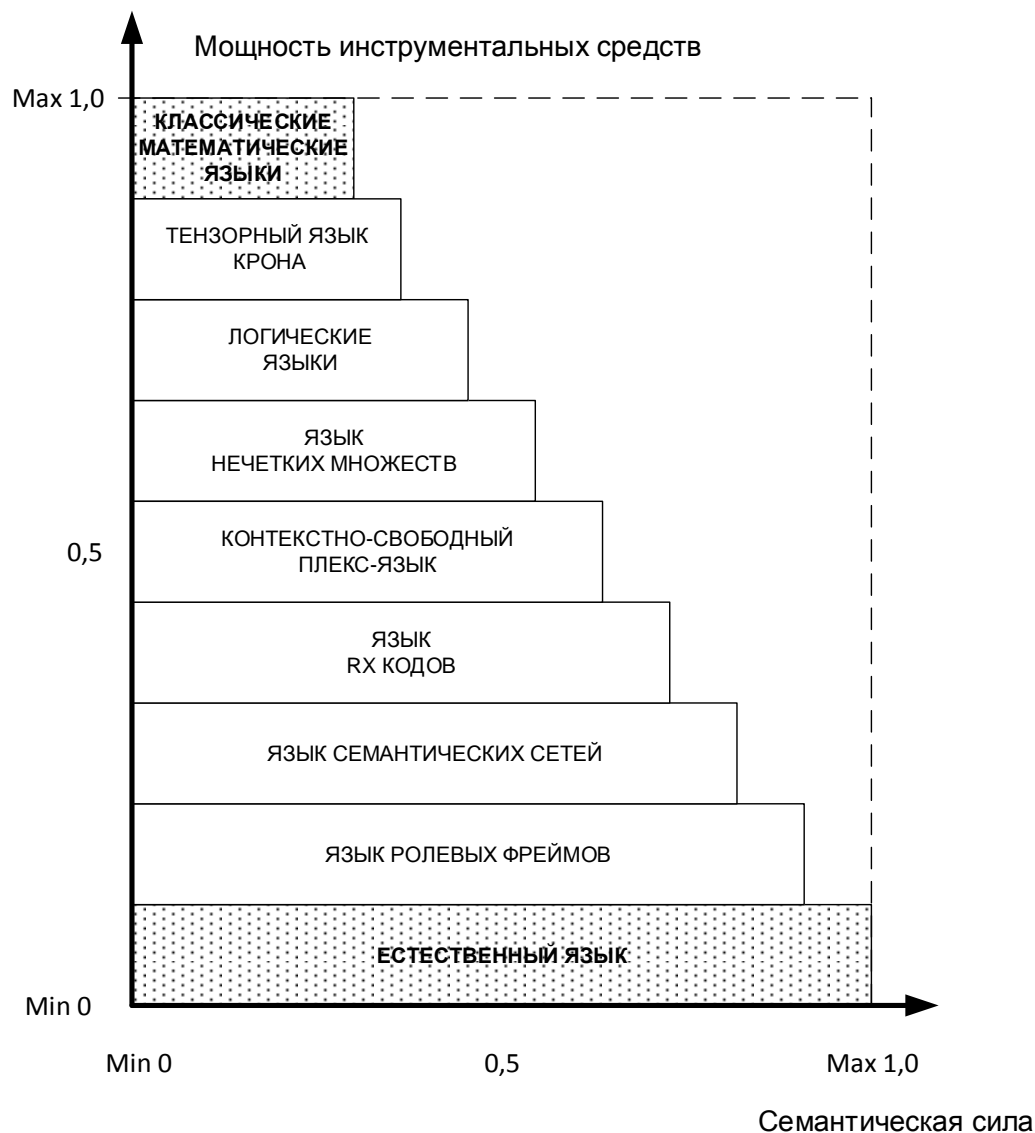


Рис. 2.12. Сравнительная характеристика языков представления знаний по мощности инструментальных средств и семантической силе

На этой диаграмме языки представления знаний упорядочены по двум характеристикам, определяющим возможности их практического использования при проектировании ИБЗ. Первая характеристика, названная семантической силой языка, отражает его описательные возможности, то есть возможности адекватного и полного описания про-

блемных областей. Эта характеристика отложена на горизонтальной оси диаграммы.

Вторая характеристика, названная мощностью инструментальных средств языка, отражает его возможности по построению эффективных систем эквивалентных преобразований предложений языка, то есть аппарата, позволяющего (за конечное число шагов) однозначно определять синтаксическую и семантическую правильность предложений. Эта характеристика отложена на вертикальной оси диаграммы. Там же приведены примеры математических средств и теорий, характеризующих мощность инструментальных средств анализируемых языков.

Кроме того, на диаграмме помимо анализируемых языков представления знаний приведены математические (алгебраические, теоретико-множественные и др.) и естественные языки, имеющие предельные значения указанных характеристик. Из этой диаграммы видно, что требования выразительности и мощности инструментальных средств языка являются противоречивыми: чем выше семантическая сила языка, тем ниже мощность его инструментальных средств, и, наоборот, с ростом мощности инструментальных средств семантическая сила языка падает. Отсюда следует, что при практическом проектировании и создании ИБЗ невозможно выбрать какой-либо один язык, адекватно удовлетворяющий указанным требованиям. Речь может идти о некоторой совокупности языковых средств, которые совместно могут обеспечить как требуемый уровень выразительных возможностей ИБЗ, так и необходимую «суммарную» мощность ее инструментальных средств.

3. ТЕХНОЛОГИИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

В большинстве практических случаев при проектировании ИБЗ представление знаний осуществляется по так называемой прямой технологии: «знания пользователя» → «компьютерная программа». Такая технология слабоуправляема и обладает рядом других существенных недостатков.

Во-первых, в ней затруднен перенос знаний от пользователя к проектировщику. Естественно, что конечный пользователь знает о предметной области несоизмеримо больше чем проектировщик ИБЗ. На то, собственно, он и конечный пользователь. Однако его знания о предметной области, как правило, неупорядочены, фрагментарны, многое в его мыслях и словесных текстах содержится за кадром (подразумевается), но напрямую не выражается. Многие термины, употребляемые пользователем, аллегоричны, ассоциативны и синонимичны. Переносить такие сведения в компьютерные программы напрямую чрезвычайно трудно. Более того, конечный пользователь зачастую сам не представляет, чего он знает и чего не знает о предметной области, и без применения специальной технологии невозможно выявить его знания об этой области в полном или хотя бы в частичном объеме.

Во-вторых, конечных пользователей ИБЗ много и их знания о проблемной области могут существенно разниться и по форме и по смыслу. При одношаговом (одноактном) подходе к процессу представления знаний чрезвычайно трудно выявить логические противоречия в знаниях пользователей и их неполноту. Человек способен работать с такими (противоречивыми и неполными) знаниями, а компьютер – нет.

В-третьих, одноактная и прямая технология представления знаний в компьютере практически не оставляет конечному пользователю места в этом процессе. Предоставляя ему право лишь на формирование начального замысла и приемку результатов. На практике это выливается в бесконечные доработки и переделки алгоритмов и программ, что существенно удорожает сроки реализации проекта и отдаляет сроки его сдачи заказчику.

Вместо такой технологии предлагается поэтапная итеративная технология представления знаний в процессе системного проектирования ИБЗ (рис. 3.1), которая свободна от указанных недостатков. Смысл этой технологии заключается в том, что представление знаний осуществляется не одноактно, и не методом спонтанных итераций «конечный пользователь-программист», а путем последовательной реализации следующих этапов:

Этап I. Представление знаний на основе ролевых фреймов (переход от естественного языка к фреймовым описаниям).

Этап II. Представление знаний с помощью концептуальной семантической сети (переход от фреймовых описаний к описаниям в виде концептуальной семантической сети).

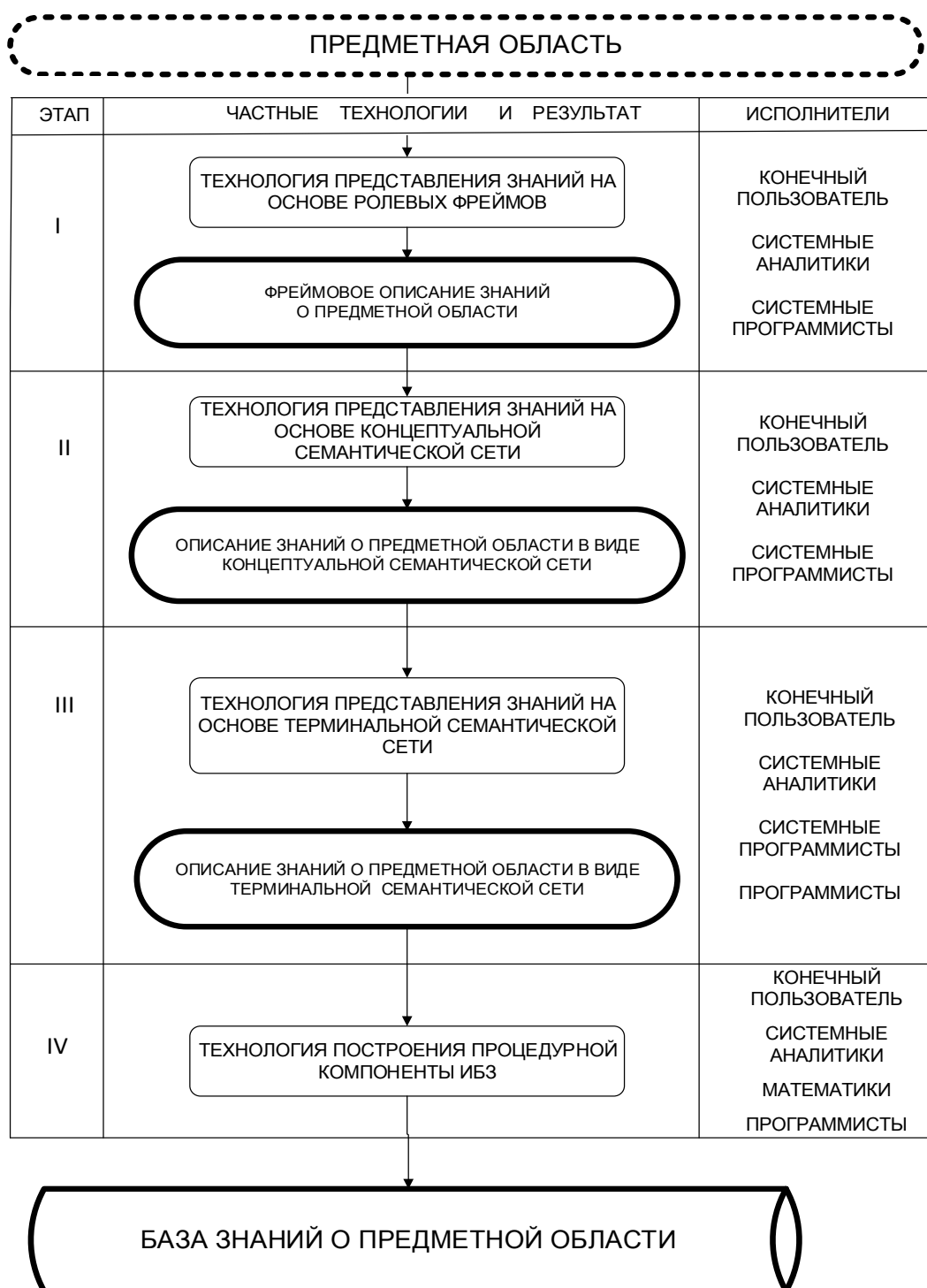


Рис. 3.1. Общая технология представления знаний в процессе системного проектирования ИБЗ

Этап III. Представление знаний с помощью терминальной семантической сети (переход от описаний в виде концептуальной семантической сети к описаниям в виде терминальной семантической сети – расширение концептуальной семантической сети).

Этап IV. Построение процедурной компоненты ИБЗ, отражающей динамику предметной области, оптимизация порядка работы прикладных модулей и формирование полной структуры ИБЗ.

Преимущества такого подхода по сравнению с одноактной и спонтанно-итеративной процедурой «естественный язык → компьютерная база знаний» определяются следующими положениями.

Во-первых, реализация такой технологии при проектировании ИБЗ дает возможность заменить интуитивные эвристические соображения строго формальными методами формирования единиц знаний.

Во-вторых, существенно снижаются требования к программным языковым средствам (языкам программирования высокого уровня), используемым для компьютерной реализации базы знаний.

В-третьих, такая технология за счет последовательно-итеративной организации выполнения операций позволяет более полно использовать возможности и знания конечного пользователя, отводя ему не только роль контролера, но и непосредственного исполнителя проекта по созданию ИБЗ.

В-четвертых, в такую технологию органически вписываются вопросы оптимизации работы пакетов прикладных программ, что актуально для крупномасштабных ИБЗ, ориентированных на логическую и сетевую обработку информации.

В-пятых, разукрупнение процедуры представления данных позволяет ввести специализацию проектировщиков ИБЗ, при которой исполнители различных категорий решают свойственные им задачи, оставляя руководителю проекта роль координатора.

В-шестых, открывается возможность поддержки проектных решений путем использования четырех частных технологий, а именно: технологии представления знаний на основе ролевых фреймов; технологии представления знаний на основе концептуальной семантической сети; технология представления знаний на основе терминальной семантической сети; технология построения процедурной компоненты базы знаний.

Рассмотрим содержание указанных частных технологий.

3.1. ПРЕДСТАВЛЕНИЕ ЗНАНИЙ РОЛЕВЫМИ ФРЕЙМАМИ

В основу такого представления знаний положена идея, выдвину-
тая американским ученым Марвином Минским, показавшим, что
фреймовое представление адекватно воспроизводит мыслительный
процесс человека при отображении в его психике знаний о любой про-
блемной области. Суть технологии, реализующей эту идею, иллюстри-
руется схемой рис. 3.2 и заключается в переходе от естественно-
языкового описания предметной области к ее фреймовому описанию.

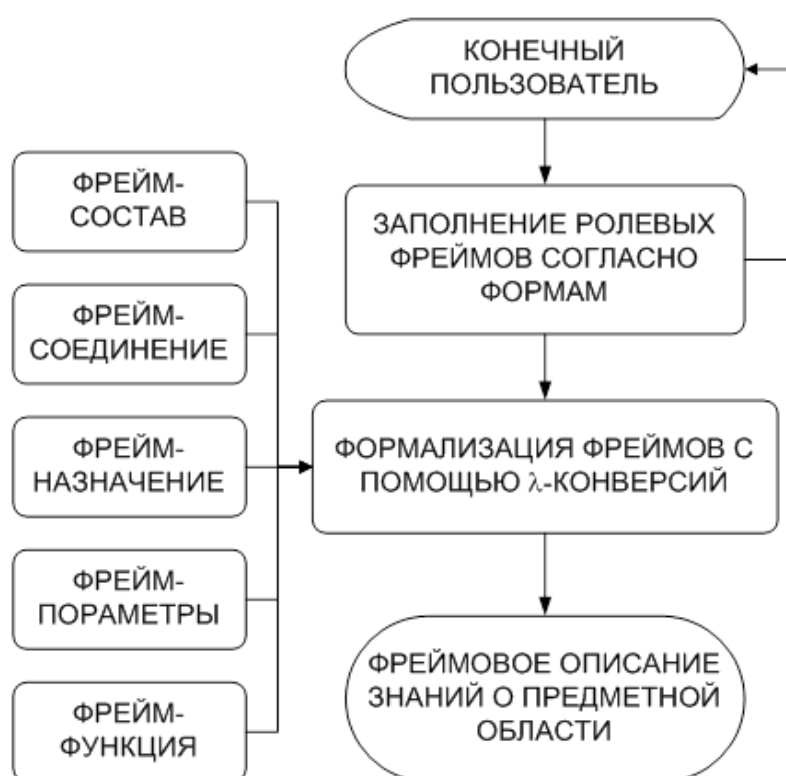


Рис. 3.2. Технология представления знаний с помощью ролевых фреймов

Для компьютерной реализации такого перехода можно использо-
вать различные алгоритмические и программные средства: рекурсив-
ные функции, KRL и FRL-языки, λ -конверсии, язык универсального
семантического кода (УСК) и другие. Однако среди перечисленных
средств несомненными преимуществами обладают λ -конверсии. Во-
первых, они носят наиболее универсальный характер в смысле воз-
можностей описания предметной области. Во-вторых, λ -конверсии
имеют достаточно простую и в тоже время эффективную систему фор-
мальных эквивалентных преобразований, имеющую всего четыре пра-
вила, главное из которых – подстановка. В-третьих, λ -конверсии до-
пускают прямую интерпретацию с помощью компьютерных языков
высокого уровня (во всяком случае для этого не требуется построение

специальных трансляторов, как это необходимо, например, при использовании KRL и FRL-языков). В-четвертых, фреймовые описания на основе λ -конверсий позволяют перейти к представлению предметных знаний в виде терминальных семантических сетей, что открывает новые, более широкие возможности формализованного представления знаний (поскольку мощность формальных эквивалентных преобразований в семантических сетях много больше, чем мощность аналогичных преобразований λ -конверсий).

Рассмотрим представление ролевых фреймов с помощью расширенной модификации λ -конверсий. Расширения достигаются за счет введения дополнительных постулатов:

$$\alpha: (\lambda\phi. (\lambda x. \phi)) \psi \text{ imc } \lambda x. \psi;$$

$$\alpha^*: \lambda x. D \text{ imc } \lambda B. [B/x] D,$$

где: λ – оператор функциональной абстракции; imc – отношение непосредственной конвертируемости; ϕ, ψ – предикатные или функциональные символы одного сорта; B, D – правильно построенные формулы языка; x – переменная.

Использование постулата α позволяет распространить понятие слота не только на переменные, но и на предикатные и функциональные символы. Это предоставляет возможность декларировать в рамках одного λ -выражения необходимое количество фреймов при совпадении их с точностью до аргументных мест и сортов у замещаемых предикатных или функциональных символов. Постулат α^* допускает вложение λ -выражений друг в друга без ограничений глубины вложения, что позволяет адекватно описывать многослойные структуры.

При конструировании λ -выражений будем исходить из того, что в ИБЗ должны быть представлены знания о следующих сущностях: иерархических объектах различной природы и процессах в них протекающих; законах функционирования элементов (подсистем и систем) объектов на всех стратах их представления; взаимосвязях и взаимодействиях элементов (подсистем и систем) объектов на всех стратах их представления; особенностях управления объектами и положений эксплуатационной документации, необходимой операторам для управления ИКС; средствах поддержания и ведения диалога с оператором на языке, ограниченном профессиональной лексикой. Тогда минимально необходимый набор типовых фреймов должен включать: фрейм-состав; фрейм-соединение; фрейм-назначение; фрейм-параметр и фрейм-функция.

Фрейм-состав (F_C) предназначен для представления знаний о предметной структуре (составе) исследуемых объектов на различных уровнях детализации. Им отражается ситуация: «объект x содержит объекты y_1, y_2, \dots, y_n , которые имеются в объекте x в количествах $c_1,$

c_2, \dots, c_n , соответственно». Такой фрейм описывается вложенным λ -выражением вида:

$$F_C = (...(\lambda x: D_x, y_1: D_1, \dots, y_n: D_n. \text{CONTAIN} (\langle ch, x \rangle, \langle v, y_1 \rangle, \dots, \langle v, y_n \rangle)); \lambda: y_1: D_1, c_1: D_1^*. \text{QUANTITY} (\langle ch, y_1 \rangle, \langle v, c_1 \rangle)) \dots \lambda y_n: D_n, c_n: D_n^*. \text{QUANTITY} (\langle ch, y_n \rangle, \langle v, c_n \rangle)) \text{ conv} \quad (3.1)$$

$\lambda x: D_x. \text{CONTAIN} (\langle ch, x \rangle, \langle v, \lambda y_1: D_1, c_1: D_1^*. \text{QUANTITY} (\langle ch, y_1 \rangle, \langle v, c_1 \rangle)) \dots \langle v, \lambda y_n: D_n, c_n: D_n^*. \text{QUANTITY} (\langle ch, y_n \rangle, \langle v, c_n \rangle)))$, где запись вида $x: D_x$ (и аналогичные) означает, что переменная x имеет сорт D_x . Через ch и v обозначены падежные отношения («характеристика» и «значение характеристики, соответственно»).

Фрейм-соединение (F_S) предназначен для отображения различных типов соединений в системах. Он отображает ситуацию «субъект x соединяет объект y с объектом z » и описывается λ -выражением вида:

$$F_S = \lambda \{ x: D_x, y: D_y, z: D_z. \text{CONNECT} (\langle s, x \rangle, \langle o, y \rangle, \langle o, z \rangle) \}, \quad (3.2)$$

где s и o – падежные отношения, соответственно «кого соединяют» (то, что производит действие) и «с кем соединяют» (то, над чем совершается действие), D – имя или сорт объекта (субъекта).

Нетрудно заметить, что фреймы типа (3.2) легко вкладываются друг в друга, что позволяет использовать их для описания иерархических структур в системах различного назначения.

Фрейм-назначение (F_N) служит для представления знаний о предназначении объектов предметной области через их функции и описывается λ -выражением вида:

$$F_N = \lambda R: D_R, x: D_x, y: D_y, w: D_w, z: D_z. R(\langle s, x \rangle, \langle o, y \rangle, \langle u, w \rangle, \langle d, z \rangle), \quad (3.3)$$

где u и d – падежные отношения, соответственно «быть входом», «быть выходом».

Пример использования (3.3) проиллюстрируем записью следующего выражения: объект (K) преобразует сообщение (C) из формы (IC) в форму (PC). Этому выражению соответствует следующая нормальная форма λ -выражения:

$$\text{TRANSFER} (\langle s, KC \rangle, \langle o, C \rangle, \langle u, IC \rangle, \langle d, PC \rangle), \quad (3.4)$$

которая получается при подстановке предикатного символа и индивидуальных констант соответствующих сортов в (3.3), то есть $(((((\lambda R: D_R, x: D_x, y: D_y, w: D_w, z: D_z. R(\langle s, x \rangle, \langle o, y \rangle, \langle u, w \rangle, \langle d, z \rangle) \text{ TRANSFER}) KC) C) IC) PC) \text{ conv TRANSFER} (\langle s, KC \rangle, \langle o, C \rangle, \langle u, IC \rangle, \langle d, PC \rangle))$.

Фрейм-параметры (F_P) предназначен для представления знаний о значениях параметров какого-либо объекта в дискретные моменты времени. λ -выражение для этого вида фреймов аналогично (3.2) с добавлением падежного отношения «быть моментом времени».

Фрейм-функция (F_F) описывает порядок расчета параметров p_i некоего объекта при заданной функции $p_i(t) = f [(a, a_2, \dots, a_N), t]$, где a_j

– аргументы, к которым применяется функция, t – текущее время. Обобщенное λ -выражение для фреймов этого типа имеет вид:

$$F_F = \lambda\{p: D_p, t: D_t, f: D_f, a_1: D_1, \dots, a_N: D_N. \text{ CALCULATE } (\langle res, p \rangle, \langle \tau, t \rangle) = (\langle vf, f \rangle) (\langle arg_1 a_1 \rangle, \dots, \langle arg_N a_N \rangle)\}, \quad (3.5)$$

где res – результат применения функции, arg – аргумент, vf – падежное отношение «вид функции», τ – падежное отношение «быть моментом времени», а запись $x: D_x$ означает, как и ранее, что переменная x имеет имя или сорт D_x .

При наличии стандартных вычислительных процедур использование выражения (3.5) позволяет задать алгоритм вычисления сложной составной функции произвольного вида.

Представленные выше фреймовые описания обладают двумя очевидными недостатками. Во-первых, в них присутствует дублирование однотипной информации, что приводит при большем объеме знаний к нерациональному расходу памяти. Поэтому в практических моделях, использующих фреймовые описания, применяется принцип наследования свойств. Суть его состоит в упорядочении рекурсии (2.11) с помощью специального дерева зависимостей, в котором узлы соответствуют фреймам, а путь от корней к вершине указывает на порядок рекурсии. При этом вся информация, записанная во фреймах, лежащих на пути от корневой вершины до данной, автоматически переносится и в данную вершину. Во-вторых, фреймовые описания (в какой бы форме они не отображались) не обладают развитым аппаратом формальных эквивалентных преобразований.

Устранение этого недостатка возможно путем перехода от фреймовых описаний к представлению знаний семантическим сетями. Вместе с тем в относительно простых ситуациях фреймовые описания позволяют построить логические конструкции, обеспечивающие вывод новых знаний. В настоящее время для этого широко используется метод продукций с использованием, как правило, связок строго имплекативного вида: $\alpha \rightarrow \beta$. Представляется, что более уместно воспользоваться понятием дедуктивной секвенции, выступающей расширением продукционных выводов.

Определение 3.1. Дедуктивной секвенцией будем называть выражение:

$$\begin{array}{c} G_1 \Leftarrow \\ \dots \Leftarrow \\ G_n \Leftarrow \end{array} \left| \begin{array}{c} \\ F \\ \end{array} \right| \begin{array}{c} \rightarrow \Delta_1 \\ \rightarrow \dots, (n, m \geq 0), \\ \rightarrow \Delta_m \end{array} \quad (3.6)$$

где G_i и Δ_j – antecedentes и succedentes секвенции соответственно, представляющие собой конечные последовательности правильно построенных формул.

Выражение (3.6) понимается следующим образом:

- $n = 0, m = 0$, то есть $0 \Leftarrow F \rightarrow 0$ как «имеется противоречие»;
- $n = 0, m \neq 0$, то есть $0 \Leftarrow F \rightarrow (\Delta_1 \wedge \Delta_2 \wedge \dots \wedge \Delta_m)$ как «имеет место $\Delta_1 \wedge \Delta_2 \wedge \dots \wedge \Delta_m$ »;
- $n \neq 0, m = 0$, то есть $(G_1 \wedge G_2 \wedge \dots \wedge G_n) \Leftarrow F \rightarrow 0$ как « $G_1 \wedge G_2 \wedge \dots \wedge G_n$ приводят к противоречию»;
- $n \neq 0, m \neq 0$, то есть $(G_1 \wedge G_2 \wedge \dots \wedge G_n) \Leftarrow F \rightarrow (\Delta_1 \wedge \Delta_2 \wedge \dots \wedge \Delta_m)$ как «если $G_1 \wedge G_2 \wedge \dots \wedge G_n$, то имеет место $\Delta_1 \wedge \Delta_2 \wedge \dots \wedge \Delta_m$ ».

Из приведенных записей видно, что замена стандартных импликативных продукций дедуктивными секвенциями вида (3.6) позволяет строить секвенциальные деревья для описания тех ситуативных знаний о предметной области, которые в явном виде не содержались в исходном описании.

Пример. Пусть в некоторой системе имеет место отклонение параметра «р» от нормы. Необходимо выяснить по фрейму-составу F_C имеется ли резервный элемент. Если он имеется (определяется по фрейм-параметры F_P), то необходимо его включить и произвести соответствующие изменения во фрейм-параметры. В виде древовидного исчисления секвенций указанное выражение записывается в виде:

$$((\langle p \not\subset P \rangle \Leftarrow F \rightarrow F_S) \Leftarrow F \rightarrow F_S). \quad (3.7)$$

3.2. ПРЕДСТАВЛЕНИЕ ЗНАНИЙ КОНЦЕПТУАЛЬНЫМИ СЕМАНТИЧЕСКИМИ СЕТЯМИ

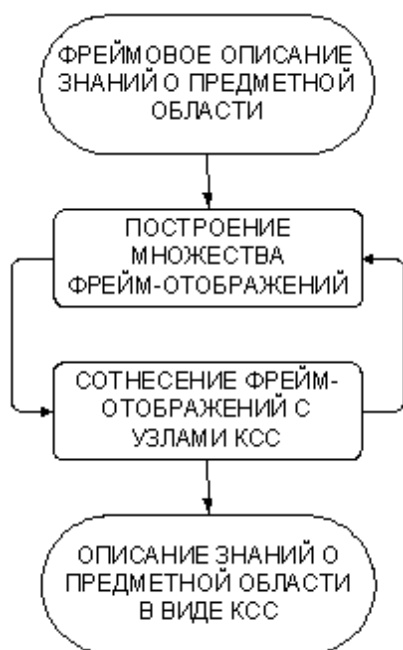


Рис. 3.3. Технология представления знаний с помощью КСС

Во фреймовых описаниях отсутствуют прямые указания на отношения между понятиями-компонентами (это содержится за кадром, подразумевается). Для того чтобы эти отношения стали явными предлагается использовать концептуальные семантические сети. С помощью этой сети детализируется (уточняется) знания, содержащиеся во фреймовых описаниях, за счет введения отношений (r_i), например, «быть частью – r_1 », «быть подчиненным» и т.д. Суть технологии представления знаний концептуальными семантическими сетями иллюстрируется схемой рис. 3.3 и заключается в реализации перехода от фреймовых описаний предметной

области к ее описаниям на языке концептуальных семантических сетей. Введем ряд определений.

Определение 3.2. Концептуальной семантической сетью (далее – КСС) будем называть граф, узлы которого принадлежат множеству $X = \{x\}$, а дуги (то есть ориентированные бинарные связи) – множеству $R = \{r\}$.

Элементы множества X соответствуют обобщенным семантическим категориям – описаниям абстрактных понятий. Элементы множества R относятся к отношениям типа «быть элементом», «содержать», «иметь имя», «быть функцией», «быть агентом», «быть акцией» и другим, часть которых поясняется далее.

На рис. 3.4 в качестве иллюстрации представлен фрагмент КСС применительно к предметной области «управление персоналом», обозначенной цифрой 1. Остальные цифры означают: 2 – персонал; 3 – функция; 4 – информация; 5 – руководитель; 6 – отдел продаж; 7 – отдел кадров; 8 – генеральный директор; 9 – зам. генерального директора по кадрам; 10 – данные отдела маркетинга; 11 – данные отдела кадров.

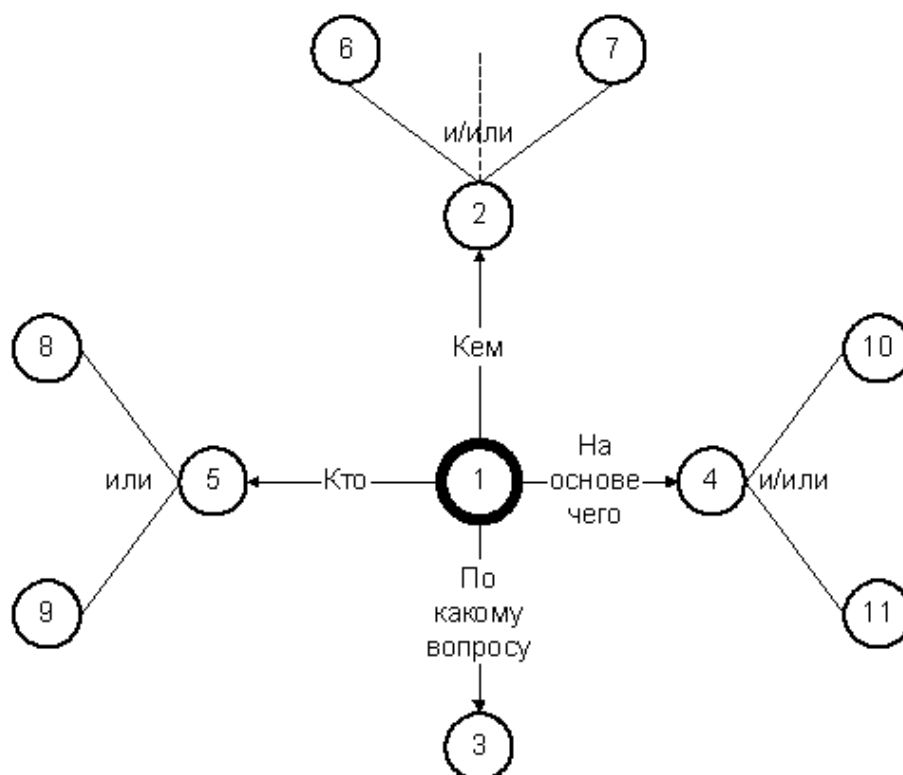


Рис. 3.4. Фрагмент КСС предметной области «управление персоналом»

Определение 3.3. Окрестностью первого порядка (или 1-окрестностью) относительно $x_i \in X$ будем называть множество пар $\{r^{i,1}, x^{i,1}\} = \xi_{x_i}^{(1)}$, таких, что $\{r^{i,1}\} = R^{i,1}$ представляет собой множество исхо-

дящих из x_i отношений (связей), а $\{x^{i,1}\} = X^{i,1}$ – множество узлов концептуальной семантической сети, присоединенных к x_i связями $\{r^{i,1}\}$.

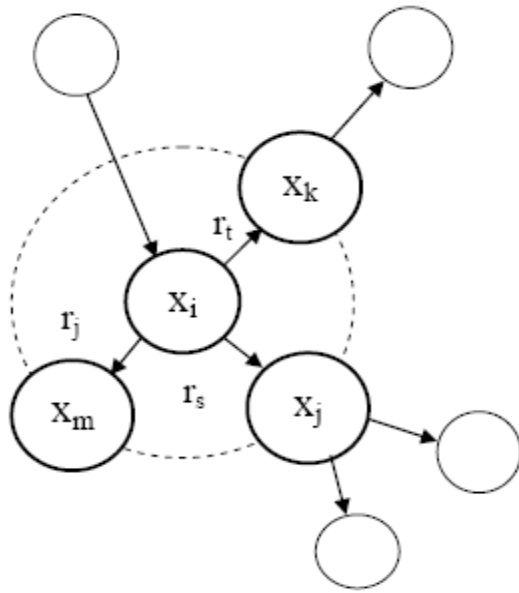


Рис. 3.5. К понятию окрестности КСС

Таким образом, все узлы окрестности $\xi_{x_i}^{(1)}$ на графе КСС отделены от x_i путем единичной длины (рис. 3.5). Узлы k -го уровня, отделенные от x_i путями длины k обозначим $x^{i,k}$, а отношения k -го уровня, присоединяющие $\{x^{i,k}\}$ к узлам $\{x^{i,k-1}\}$, обозначим $\{r^{i,k}\}$. Соответствующие множества получают обозначения $X^{i,k} = \{x^{i,k}\}$ и $R^{i,k} = \{r^{i,k}\}$. Тогда окрестность k -го порядка относительно x_i (или k -окрестностью) будет объединение множеств узлов и отношений, входящих в $k-1$ окрестность относительно x_i со всеми I окрестностями узлов множества $X^{i,k-1}$:

$$\xi_{x_i}^{(k)} = \xi_{x_i}^{(k-1)} \cup \left(\bigcup_{n=1}^{k-1} \xi_{x^{i,n}}^{(1)} \right), k \geq 2. \quad (3.8)$$

Определение 3.4. Фреймом-отображением f_i назовем информационную структуру, соответствующую I -окрестности узла x_i в КСС.

Соответствие между x_i и f_i определим зависимостью $f_i = G(x_i)$, где функция G задает отображение множества узлов X и отношений R , образующих КСС, на множество фреймов F , образующих предметную часть базы знаний $G: (X, R) \rightarrow F$.

Введем в рассмотрение предикат $P_{r_k}(x_i, x_j)$, принимающий значение «истина», если в рассматриваемой КСС к узлу x_i посредством отношения r_k присоединен узел x_j . Формально во введенных обозначениях соответствие между фрейм-отображениями и узлами КСС выражается соотношением

$$\begin{aligned} \forall_{f_i} (f_i \in F) \exists x_i (x_i \in X) : \{ (f_i = G(x_i) \wedge (\forall x_j (x_j \in X^{i,1})) \} \quad (3.9) \\ \exists_{f_i} (f_i \in F^i, F^i \subset F) : [f_j = G(x_j) \wedge P_{r_k}(x_i, x_j)] \}. \end{aligned}$$

Это соотношение означает, что всякий фрейм f_i , соотносимый с некоторым узлом x_i , связывается с подмножеством «подчиненных»

фреймов F^i , элементы которого $\{f_j\} = F^i$ соответствуют узлам $\{x_j\} = X^{j,1}$, I -окрестности узла x_j .

3.3. ПРЕДСТАВЛЕНИЯ ЗНАНИЙ ТЕРМИНАЛЬНЫМИ СЕМАНТИЧЕСКИМИ СЕТЯМИ

Суть этой технологии иллюстрируется схемой рис. 3.6 и заключается в реализации перехода от описания предметной области в виде КСС к так называемой терминальной семантической сети (далее – ТСС).

В процессе решения конкретных задач пользователи оперируют понятиями, отражающими реальные физические объекты, факты, события и другие единицы знаний. Когда ИБЗ сообщается (или выводится ею в процесс функционирования) новый факт или описание нового объекта, его можно представить конъюнкцией предикатов

$$P_{r_1}(\bar{x}_1, \bar{x}_2) \wedge P_{r_2}(\bar{x}_2, \bar{x}_3) \wedge \dots \wedge P_{r_k}(\bar{x}_k, \bar{x}_{k+1}), \quad (3.10)$$

где $\{r_i\} \subset R$, $i = 1, 2, \dots, k$ – связи (отношения) в КСС, а элементы $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{k+1}\} \subset \bar{X}$ – терминальные величины, соответствующие либо наименованию определенного объекта, либо числу, либо тексту, либо стандартной комбинации терминальных величин (множество, вектор, матрица, структурная запись из полей, заполненных терминальными величинами).

Таким образом, в процессе представления новой единицы знаний происходит активизация некоторых фрагментов

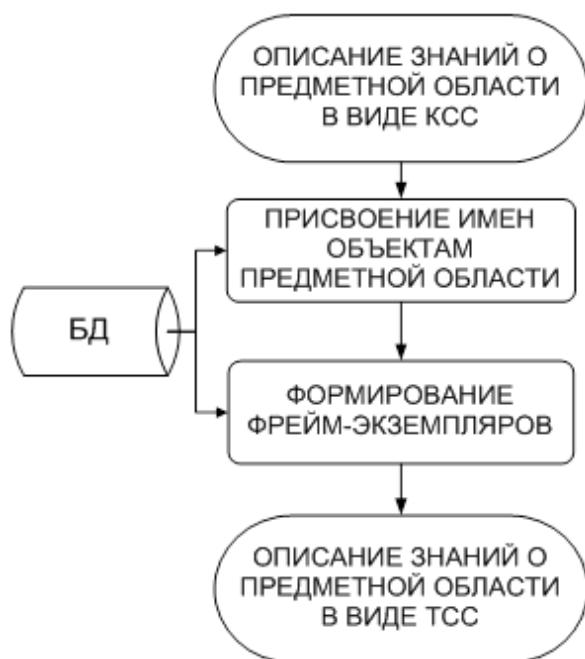


Рис. 3.6. Технология представления знаний с помощью ТСС

КСС, выражающаяся в том, что вместо абстрактных символов $\{x_i\}$ подставляются терминальные величины $\{\bar{x}_i\}$. Очевидно, что при обработке очередной единицы знаний, активизируемый участок КСС является связным компонентом графа (X, R) . В этом случае удобно из исходной КСС выделять активизируемые фрагменты и формировать из них так называемую терминальную семантическую сеть (ТСС), установив со-

ответствие между узлами $\{x_i\}$ и конкретизирующими их значениями $\{\bar{x}_i\}$.

Такие соответствия можно задать специальными связями типа «иметь имя» и «иметь значением», которые присоединяют любой символ \bar{x}_i в ТСС к соответствующему узлу x_i в КСС. Следовательно, можно ввести предикат $\bar{P}(x_i, \bar{x}_i)$, истинность которого указывает на наличие указанного типа связей между \bar{x}_i и x_i ($\bar{P}(x_i, \bar{x}_i) = \text{«истина»}$, если между \bar{x}_i и x_i существует связь типа «иметь имя» и «иметь значением»). Учитывая введенный формализм, можно считать ТСС некоторым продолжением КСС и рассматривать ее как единую семантическую сеть, содержащую как абстрактные понятия, так и терминальные величины. Однако с практической точки зрения удобнее рассматривать отдельно КСС и ТСС, имея в виду, что при решении задач информационного поиска множество терминальных величин, описывающих конкретную предметную область, представляет собой некоторую базу данных в ее традиционном понимании.

Как уже отмечалось, всякое входное сообщение ИБЗ содержит некоторое понятие – тему высказывания. Такому понятию соответствуют определенный узел x_i в КСС и фрейм f_i в базе знаний. Некоторые из фигурирующих в высказывании величин $\{\bar{x}\}$ указывают на x_i либо непосредственно (с помощью связей «иметь имя» и «иметь значением»), либо косвенно, ссылаясь на узлы из окрестности $\xi_{x_i}^{(1)}$. Будем говорить, что такие величины образуют экземпляр фрейма f_i , обозначаемый \bar{f}_i .

Определение 3.5. Экземпляром фрейма f_i назовем информационную структуру \bar{f}_i , которая образуется из терминальных величин $\{\bar{x}\}$ и связей $\{r\}$, соответствующих (может быть и неполной) I-окрестности узла $x_i = G^{-1}(x)$, где G^{-1} – функция, обратная G . Соответствие между f_i и \bar{f}_i обозначается зависимостью $\bar{f}_i = W(f_i)$.

Каждое высказывание о некотором объекте, факте, событии или процессе можно представить в базе знаний совокупностью взаимосвязанных экземпляров $\{\bar{f}_i\} \subset \bar{F}$, соответствующих подмножеству $\{f_i\} \subset F$. В конечном счете, вся ТСС может быть представлена в базе знаний множеством экземпляров F , описывающих некоторый фрагмент предметной области. При этом, ТСС интеллектуальной базы знаний состоит из совокупности взаимосвязанных фреймов-отображений, образующих пользовательский уровень знаний, и совокупности взаимосвязанных фреймов-экземпляров, образующих ее прагматический уровень.

Таким образом, логика перехода от концептуальных к терминальным семантическим сетям очень проста и заключается в назначении объектам, образующим КСС, конкретных имен за счет введения отношения «быть именем» (r_2) и понятий-имен. Проще: ТСС – это та же КСС, дополненная отношением «быть именем» и конкретными именами всех объектов КСС. Иначе, ТСС – это персонализированная КСС.

3.4. ТЕХНОЛОГИЯ ПОСТРОЕНИЯ ПРОЦЕДУРНОЙ КОМПОНЕНТЫ БАЗЫ ЗНАНИЙ

Эта технология иллюстрируется схемой рис. 3.7 и применяется на завершающем этапе проектирования ИБЗ.

Представление предметной области в виде КСС и ТСС позволяет описывать статические отношения между объектами и терминальными единицами предметной области. Однако в практических приложениях этого недостаточно, поскольку зачастую требуется описать динамику процесса, в частности условия формирования новых фрагментов ТСС. Это становится возможным, если ввести правила привязки к некоторым элементам КСС.

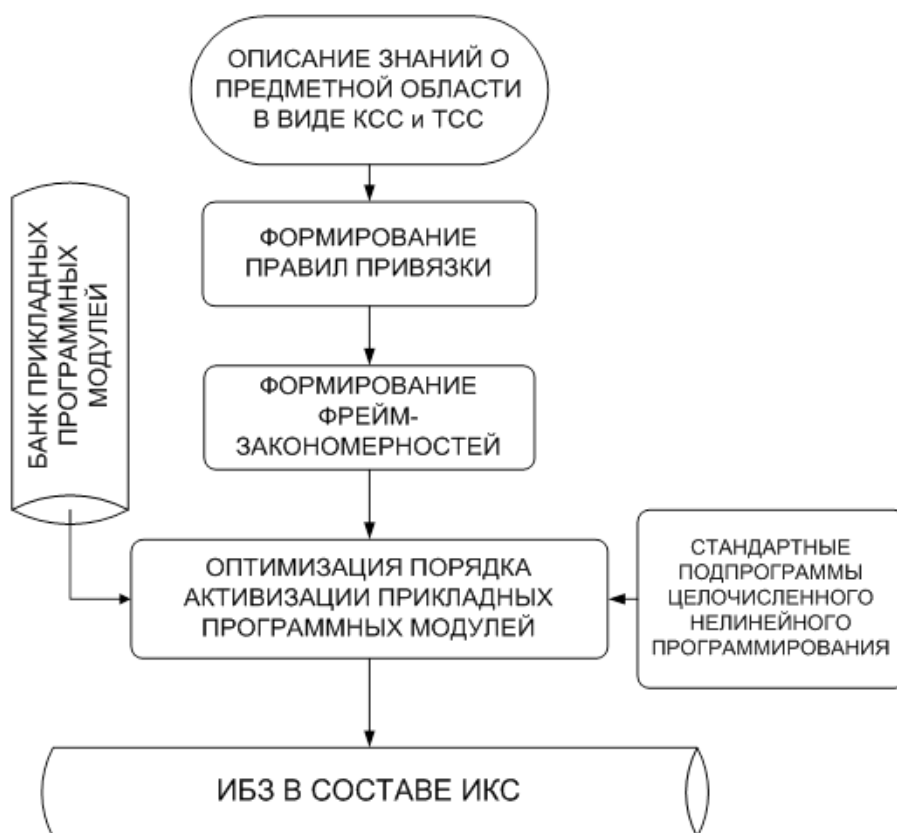


Рис. 3.7. Технология построения процедурной компоненты ИБЗ

Определение 3.6. Правилем привязки σ назовем тройку $\langle c, s, d \rangle$, в которой «с» – условие применимость данного правила, «s» – следствие, содержащее список операций, подлежащие выполнению в момент применения правила, «d» – задержанное действие, содержащее список операций, которые должны быть выполнены по окончании обработки всех правил.

Условие «с» формируется в терминах состояния ТСС. Оно предлагает выполнение некоторых характеристик терминальных величин или некоторых конъюнкций предикатов вида $\bar{P}(x_i, \bar{x}_i)$, позволяющих установить существование связей между объектами в КСС и терминальными величинами в ТСС.

Следствие «s» задает последовательность операций, таких, как формирование или модификация фрагментов ТСС, инициирование новых правил и т.п.

Задержанное действие «d» отражает последовательность обращения к прикладным программным модулям (ППМ), составляющих пользовательский уровень базы знаний. В частном случае «d» определяет порядок вложения секвенций (3.6).

Все множество правил $\{\sigma\} \subset \Sigma$ можно разбить на некоторые подмножества Σ_i , связываемые с дугами r_i в КСС. Тогда при обработке фрагмента сети, включающего дугу r_i , происходит инициирование всех связанных с ней правил Σ_i . В результате этого порождаются два процесса. Первый, обусловленный обработкой пар $\{(c, s)\}$, приводит к дополнительной модификации ТСС. Второй обусловлен накоплением задержанных действий $\{d\}$. В результате формируется некоторая траектория активации ППМ, соответствующая входному запросу.

Множество правил Σ_i отражается в процедурной части ИБЗ информационными структурами, которые назовем фреймами-закономерностями $\tilde{f}_i \subset \tilde{F}$. Каждый такой фрейм объединяет группу правил Σ_i , связанных с дугой r_i , и проецирует пары $\{r_i, \Sigma_i\}$ на множество следствий:

$$\tilde{F}: (R, \Sigma) \xrightarrow{O_i} \mu S, \quad (3.11)$$

где μ – квантор нечеткости, принимающий такие значения как «почти всегда», «иногда», «в исключительных случаях»; O_i – сигнал, свидетельствующий о реализации данного правила.

Таким образом, четверка $\{X, R, \bar{X}, \Sigma\}$ в совокупности с (3.11), которой в предметной части ИБЗ сопоставляется множество фреймов-отображений и фреймов-экземпляров, а в процедурной части – множество фреймов-закономерностей и пакетов прикладных программных модулей, определяет все необходимые компоненты знаний для построения, модификации и анализа модели предметной области ИБЗ.

Как уже отмечалось, в процессе функционирования базы знаний происходит инициализация некоторых фрагментов КСС и ТСС, а также некоторой совокупности правил $\{r_i, \Sigma_i\}$, связанных с дугами указанных сетей. В правилах $\Sigma_i = \{c_i, s_i, d_i\}$ нас будут интересовать задержанные действия d_i , связанные с работой ППМ, являющихся неотъемлемой составной частью ИБЗ. Увязка условий c_i и следствий s_i с d_i осуществляется тем, что их обработка (c_i и s_i) производится параллельно с инициализацией фрагментов КСС и ТСС. В момент завершения инициализации образуется некоторый перечень ППМ, выполнение которых и завершает решение прикладной задачи (пользователь получает мотивированный ответ на поставленный вопрос). Если количество ППМ мало (< 10) и они слабо связаны по входам и выходам, то принципиальных трудностей не возникает. Однако для реальных ИБЗ характерно наличие большого числа ППМ, которые жестко связаны по входам и выходам. В связи с чем возникают две задачи: анализа работоспособности комплекса ППМ и оптимальной организации их выполнения. Подход к решению первой задачи основан на использовании КСПЯ, рассмотренного в главе 2. При этом плекс-элементами выступают ППМ, а КСП-грамматика содержит два дополнительных правила: в структуре ППМ не должно быть плекс-элементов с пустыми входами-выходами и матрица конкатенаций не должна иметь циклов.

Для решения второй задачи введем обозначения: $M = \{m_j\}$ – совокупность ППМ, определяемых совокупностью задержанных действий $\{d_i\}$; $\{m_j^{\text{in}}\}, \{m_j^{\text{out}}\}$ – множества входных (in) и выходных (out) полюсов модулей m_j . Оптимизацию выполнения ППМ целесообразно осуществлять в три этапа. На первом этапе все m_j упорядочиваются по следующему алгоритму.

Шаг 1. Определяются элементы $m_j^1 \in M$, для которых истинно

$$E_{m_{jk}^1} (m_j^1 \in M) ((m_j^1)^{\text{out}} \cap m_{jk}^{\text{in}} \neq \emptyset). \quad (3.12)$$

Эти элементы образуют множество модулей первого уровня $M^1 = \{m_j^1\}$.

Шаг 2. Определяются элементы $\{m_j^v\} = M^v, m_j^v \in M$, для которых

$$(\forall_{m_{ij} \in \bigcup_{s=1}^{v-1} M^s}) ((m_j^v)^{\text{in}} \cap m_{jk}^{\text{out}} \neq \emptyset). \quad (3.13)$$

Эти элементы образуют множество модулей v -го уровня.

Процесс продолжается до тех пор, пока очередное множество M^v не окажется пустым. В результате реализации указанных шагов получается некоторый ориентированный граф, отображающий связи ППМ по входам и выходам, и определяющий частичную упорядоченность

совокупности ППМ. Выполнение этой совокупности модулей во времени зависит от конфигурации комплекса технических средств, поддерживающего базу знаний. В случае, когда технические средства обеспечивают независимое выполнение модулей, они могут активизироваться в произвольном порядке, и проблем не возникает. В том же случае, когда технические средства не обеспечивают независимое выполнение модулей, для определения оптимального порядка их активизации требуется привлечение специальных методов.

Сущность второго этапа заключается в построении такого расписания активизации модулей, при котором суммарное время из выполнения будет минимальным. Вполне очевидно, что здесь возможны два режима оптимизации: а) когда для каждого модуля задан единственный исполнитель; б) когда таких исполнителей несколько (то есть один и тот же модуль может быть выполнен на различных технических средствах ИБЗ). Соответственно этому возможны две постановки задачи.

Случай единственного исполнителя. Пусть: m_{ik} – i -й модуль, выполняемый k -м исполнителем; $t(m_{ik})$ – продолжительность выполнения i -го модуля k -м исполнителем; t_{ik} – неизвестные (искомые) моменты начала выполнения i -го модуля; $Y_{i,j,k}$, переменная, принимающая два значения, 1 – если m_{ik} выполняется исполнителем k раньше, чем m_{jk} , и 0 – в противном случае. Тогда искомое расписание будет решением следующей задачи. Определить

$$\min_{t_{ik}} \max[t_{ik} + t(m_{ik})] \quad (3.14)$$

при ограничениях: 1) $t_{ik} \geq 0$; 2) $t_{ik} \geq \max[t_{ik} + t(m_{ik})]$; 3) если $m_{ik} \in M^S$, $m_{jv} \in M^P$, то $S > P$; 4) $[M + t(m_{jk})]Y_{i,j,k} + (t_{ik} - t_{jk}) \geq t(m_{jk})$; $[M + t(m_{jk})](1 - Y_{i,j,k}) + (t_{jk} - t_{ik}) \geq t(m_{ik})$.

Ограничение 2) описывает упорядоченность выполнения ППМ, определенную на первом этапе, а ограничения 3) - 4) задают требование одноканальности исполнителей. Задача относится к классу задач целочисленного нелинейного программирования и решается методом «ветвей и границ».

Случай нескольких исполнителей. Пусть: $N = \{1, 2, \dots, k, \dots, n\}$ – множество исполнителей, обеспечивающих выполнение ППМ; $t(m_i)_j$ – время выполнения модуля m_i исполнителем с номером k (если этот исполнитель в принципе не может выполнять модуль m_i , то $t(m_i)_j = \infty$). Требуется составить такое расписание исполнения модулей, при котором продолжительность выполнения всей совокупности ППМ будет минимальным. Введем следующие переменные: x_{ik} , равную 1, если m_i выполняется исполнителем с номером k , и $Y_{i,j,k}$, имеющую тот же смысл, что и в предыдущем случае. Тогда задача составления оптимального расписания выполнения совокупности ППМ в случае нескольких исполнителей запишется так:

$$\max_i [t_{ik} + t(m_i)_k], \quad (3.15)$$

при ограничениях: 1) $\sum_{m_i} x_{ik} \geq 1$; 2) $\sum_k x_{ik} = 1$; 3) $t_{ik} \geq 0$; 4) $t_{ik} \geq \max_{jv} [t_{jv} + t(m_j)_v]$, если $m_{ik} \in M^S$, $m_{jv} \in M^P$, $S > P$; 5) $[M + t(m_{jk})] Y_{i,j,k} + (t_{ik} - t_{jk}) \geq t(m_j)_k$; 6) $[M + t(m_j)_k] (1 - Y_{i,j,k}) + (t_{jk} - t_{ik}) \geq t(m_i)_k$.

Здесь ограничения 1) – 2) устанавливают правила использования исполнителей (не должно быть лишних исполнителей, то есть модулей, не задействованных для решения данной задачи пользователя, и на исполнение каждой вычислительной операции не должно назначаться более одного модуля). Смысл остальных ограничений аналогичен предыдущему случаю. Сформулированная задача, также, как и (3.15) решается методом «ветвей и границ».

4. НЕЙРОСЕТЕВЫЕ МЕТОДЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

4.1. ИДЕИ НЕЙРОИНФОРМАТИКИ

Нейросетевыми называются методы представления знаний, основанные на использовании искусственных нейронных сетей. Искусственные нейронные сети, они же коннекционистские (от англ. *connection* – связь), представляют собой устройства, использующие огромное число взаимосвязанных элементарных условных рефлексов, названных по имени канадского физиолога Д. Хебба – «синапсами Хебба».

В настоящее время нейросетевые методы представления знаний применяются для решения задач обработки изображений, управления роботами и непрерывными производствами, для понимания и синтеза речи, для диагностики заболеваний людей и обнаружения технических неполадок в приборах, для имитации пищевых технологических процессов. Та часть работ, которая связана с разработкой устройств переработки информации на основе принципов работы нейронных сетей, относится к области нейроинформатики.

Суть всех подходов нейроинформатики заключается в разработке методов синтеза нейронных сетей, имитирующих процессы функционирования различных объектов, и позволяющих решать те или иные задачи, возлагаемые на ИБЗ. Нейрон при этом выглядит как очень простое устройство: нечто вроде усилителя с большим числом входов и одним выходом. Различие между подходами и методами – в деталях представлений о работе нейрона, и, конечно, в представлениях о работе связей. В отличие от цифровых микропроцессорных систем, представляющих собой сложные комбинации процессоров и запоминающих блоков, в искусственных нейросетях память и операции сосредоточены в связях между процессорами-нейронами. Тем самым основная нагрузка на выполнение операций представления и анализа знаний ложится не на отдельные процессоры, а на всю архитектуру системы, структура которой определяется межнейронными связями.

Таким образом, ядром нейроинформатики выступает паллиативная идея, что нейроны можно моделировать довольно простыми автоматами, а вся сложность интеллекта, гибкость его функционирования и другие важнейшие качества определяются связями между нейронами. С этой идеей, выражением сути которой служит фраза: «связи – все, свойства элементов – ничто», тесно связаны следующие аксиомы:

1) несмотря на то, что элементы, из которых строя искусственные нейронные сети, однородны и просты, с их помощью можно имитировать процессы любой сложности;

2) из простых и ненадежных элементов можно построить вполне надежную систему, когда при разрушении случайно выбранной части система сохраняет свои полезные свойства;

3) предполагается, что нейросеть достаточно богата по своим возможностям и достаточно избыточна, чтобы компенсировать бедность выбора элементов, их ненадежность, возможные разрушения части связей.

Названные аксиомы, определяя принципы нейросетевых методов представления знаний, не указывают на то, как же их строить и научить решать реальные задачи. На первый взгляд кажется, что искусственные нейросети не допускают прямого программирования, то есть формирования связей по явным правилам. Это, однако, не совсем так. Существует большой класс задач (нейронные системы ассоциативной памяти, статистической обработки, фильтрации и др.), для которых связи формируются по явным формулам. Но еще большее (по объему существующих приложений) число задач требует неявного процесса. По аналогии с обучением животных или человека этот процесс называется обучением нейросетевой модели.

Методы обучения искусственных нейросетей в составе ИБЗ достаточно просты и имеют несложную практическую реализацию. Обучение обычно строится так: существует задачник – набор примеров с заданными ответами. Эти примеры предъявляются ИБЗ. Нейроны получают по входным связям сигналы – «условия примера», преобразуют их, несколько раз обмениваются преобразованными сигналами и, наконец, выдают ответ – также набор сигналов. Отклонение от правильного ответа штрафуются. Обучение состоит в минимизации штрафа как (неявной) функции связей.

Для решения некоторых типов задач уже существуют типовые конфигурации нейросетей. В общем же случае подбор оптимальной архитектуры сети является трудоемкой слабо формализуемой задачей, при решении которой обычно руководствуются следующими эвристическими правилами, заимствованными из теории проектирования сложных систем:

- возможности сети возрастают с увеличением числа, образующих ее нейронов, то есть чем больше нейронов, тем адекватнее представляются и анализируются знания о предметной области;
- плотность связей между нейронами оказывает положительное влияние на характер процесса обучения, то есть чем теснее связаны нейроны в сети, тем быстрее и полнее идет процесс ее обучения;
- сложность алгоритмов функционирования сети способствует усилению ее мощи, то есть чем разветвленнее сеть, тем выше ее возможности по представлению и анализу знания о предметной области;

– введение обратных связей повышает адаптивные способности нейросетей, но одновременно может привести к снижению ее динамической устойчивости, то есть возникают ситуации, когда уже обученная нейронная сеть начинает вести себя непредсказуемым образом.

Итак, нейросетевые методы представления знаний базируются на трех источниках – это имеющиеся на сегодняшний день представления о строении и функционировании человеческого мозга, современное психолого-педагогическое понимание процессов обучения и общие положения проектирования сложных систем.

4.2. БАЗОВЫЕ ЭЛЕМЕНТЫ И АРХИТЕКТУРА НЕЙРОСЕТЕЙ

Для создания нейросетей используется специальная «схемотехника», в которой базовые элементы – сумматоры, синапсы, нейроны и т.п. объединяются в сети, предназначенные для решения задач. Фактически эта «схемотехника» представляет собой особый язык для представления нейронных сетей и их обсуждения. При программной и аппаратной реализации, описания, выполненные на этом языке, переводятся на языки программирования высокого уровня, пригодные для программно-компьютерного исполнения.

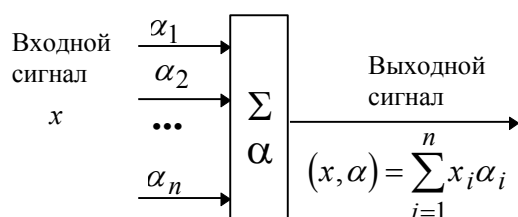


Рис. 4.1. Адаптивный сумматор

Наиболее важный элемент любой нейросети – это адаптивный сумматор (рис. 4.1), который вычисляет скалярное произведение вектора входного сигнала x на вектор параметров α . Адаптивным он называется из-за наличия вектора настраиваемых параметров α .

Для многих задач представления и анализа знаний полезно иметь линейную неоднородную функцию выходных сигналов. Ее вычисление также можно представить с помощью неоднородного адаптивного сумматора, имеющего $n + 1$ вход и получающего на 0-й вход постоянный единичный сигнал (рис. 4.2).

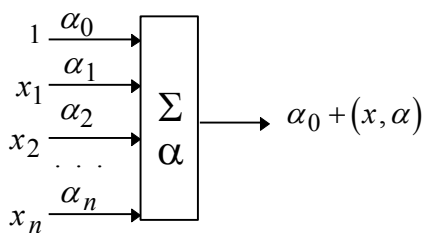


Рис. 4.2. Неоднородный адаптивный сумматор

Следующий базовый элемент нейросети называется нелинейным преобразователем (рис. 4.3).

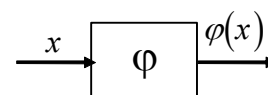


Рис. 4.3. Нелинейный преобразователь сигнала

Он получает скалярный входной сигнал x и переводит его в $\varphi(x)$, где φ – функция активации нейрона.

Точка ветвления (рис. 4.4) служит для рассылки одного сигнала по нескольким адресам. Она получает скалярный входной сигнал x и передает его всем своим выходам.

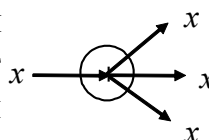


Рис. 4.4. Точка ветвления

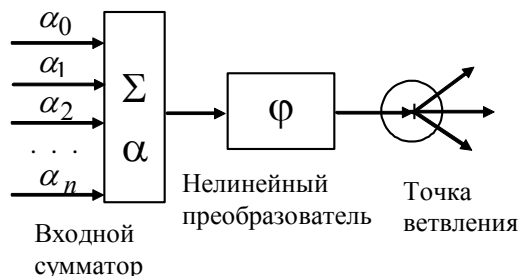


Рис. 4.5. Стандартный искусственный нейрон

Стандартный искусственный нейрон состоит из входного сумматора, нелинейного преобразователя и точки ветвления на выходе (рис. 4.5). Линейная связь (синапс) отдельно от сумматоров не встречается, однако для некоторых рассуждений бывает

удобно выделить этот элемент (рис. 4.6). Он умножает входной сигнал x на «вес синапса» α .

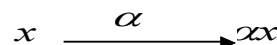


Рис. 4.6. Синапс

Иногда бывает полезно «присоединить» связи не к входному сумматору, а к точке ветвления. В результате получаем элемент, двойственный адаптивному сумматору и называемый «выходная звезда». Его выходные связи производят умножение сигнала на свои веса.

Перейдем теперь к вопросу составления нейросетей из базовых элементов. Вообще говоря, эти элементы можно соединять произвольным образом, лишь бы входы получали какие-нибудь сигналы. Но такой произвол порождает трудности обучения сети, поэтому на практике используют несколько стандартных архитектур, из которых строят нейросети.

Можно выделить две базовые архитектуры нейросетей – слоистые и полносвязные. В слоистых сетях нейроны расположены в нескольких слоях (рис. 4.7).

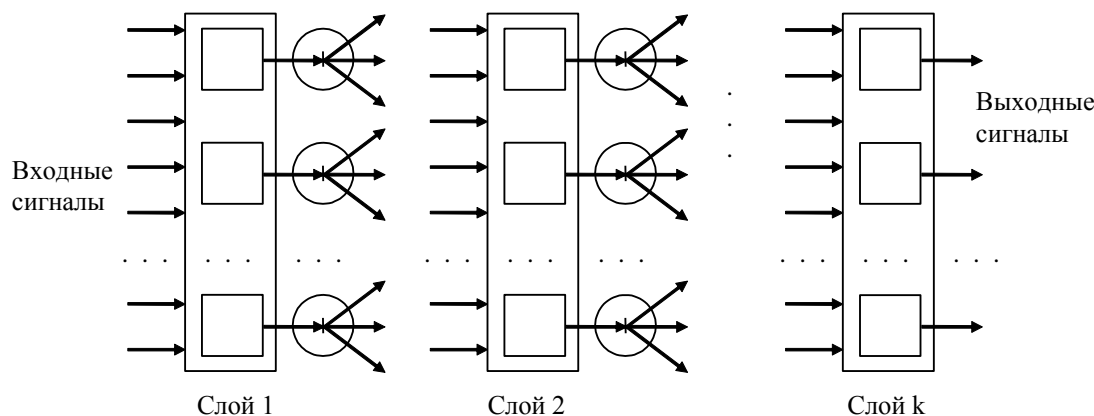


Рис. 4.7. Слоистая нейросеть

Нейроны первого слоя получают входные сигналы, преобразуют их и через точки ветвления передают нейронам второго слоя. Далее срабатывает второй слой и так до k-го слоя, который выдает выходные сигналы для интерпретатора и пользователя. Если не оговорено противное, то каждый выходной сигнал i-го слоя подается на вход всех нейронов (i+1)-го. Число нейронов в каждом слое может быть любым и не связывается с количеством нейронов в других слоях. Стандартный способ подачи входных сигналов: все нейроны первого слоя получают каждый входной сигнал. Особое распространение получили трехслойные сети, в которых каждый слой имеет свое наименование: первый – входной, второй – скрытый, третий – выходной.

В полносвязных сетях каждый нейрон передает свой выходной сигнал остальным нейронам, включая самого себя (обратная связь). Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети. Все входные сигналы подаются всем нейронам.

Элементы слоистых и полносвязных сетей могут выбираться по-разному, однако существует стандартное правило: на входе должен находиться нейрон с адаптивным неоднородным линейным сумматором на входе. Для полносвязной сети входной сумматор нейрона фактически распадается на два: первый вычисляет линейную функцию от входных сигналов сети, второй – линейную функцию от выходных сигналов других нейронов, полученных на предыдущем шаге.

Функция активации нейронов (характеристическая функция) φ , преобразующая выходной сигнал сумматора, может быть одной и той же для всех нейронов сети. В этом случае нейросеть называют однородной (гомогенной). Если же φ зависит еще от одного или нескольких параметров, значения которых меняются от нейрона к нейрону, то модель называют неоднородной (гетерогенной).

Функция активации нейрона может иметь различный вид и оказывает существенное влияние на характеристики нейросетевой модели. Одной из наиболее распространенной функцией активации является логистическая функция или сигмоид $\varphi(t) = 1 / (1 + e^{-\beta t})$, где β - параметр, подбираемый пользователем и влияющий на пологость функции.

* * *

Мы рассмотрели основные идеи нейросетевых методов представления, которые можно с успехом использовать при системном проектировании ИБЗ. Полагаем, что изложенного вполне достаточно для того, чтобы системный аналитик (не специалист в области нейроинформатики) получил представление об этом сравнительно новом методе создания в памяти компьютера моделей сложных систем.

Можно выделить три неоспоримых преимущества нейросетевого подхода к представлению знаний в процессе системного проектированию ИБЗ. Во-первых, нейронные сети обладают способностью быстрой обработки информации, что дает возможность расширить набор аналитических операций, выполняемых в реальном режиме времени. Во-вторых, нейронные сети способны к обучению, а уже будучи обученными, они могут эффективно обобщать полученную информацию и демонстрировать хорошие результаты на данных, не использовавшихся в процессе обучения. Это качество позволяет использовать нейросети для имитации функционирования объектов, для которых не удастся установить их состав, структуру и принципы работы, но частично известны реакции «вход-выход». В-третьих, однородность, проявляющаяся в составе и конструкции элементов нейросети, обеспечивает возможность построения компонентов ИБЗ на однотипных аппаратных или программных средствах, что существенно ускоряет и удешевляет процесс их проектирования.

5. ТИПОВЫЕ МЕТОДЫ ПОИСКА РЕШЕНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ БАЗАХ ЗНАНИЙ

Традиционный подход к формализации задач оценки и поиска решений состоит в определении существенных характеристик проблемной ситуации, которые совместно с существующими между ними отношениями могут быть описаны количественно. В терминах таких характеристик и отношений формируются ограничения и критерии выбора решений. При этом типовая задача оценки и поиска решений формально записывается в следующем виде:

$$\max (\min) F (x_1, x_2, \dots, x_N) \quad (5.1)$$

при ограничениях

$$G_k (x_1, x_2, \dots, x_N) = 0; k = 1, 2, \dots, K; \quad (5.2)$$

$$x_i \in X; i = 1, 2, \dots, N, \quad (5.3)$$

где переменные (x_1, x_2, \dots, x_N) – варьируемые характеристики задачи (компоненты решения); функционал (5.1) – критерий выбора решения (выбор \max или \min зависит от условия задачи и смысла критерия); соотношения (5.2) – уравнения связи между характеристиками (алгебраические, дифференциальные и др.); соотношения (5.3) – возможные значения варьируемых характеристик.

В традиционных информационно-расчетных системах для решения таких задач используется широкий арсенал методов математического программирования. В ИБЗ использование этих методов не всегда приемлемо. На то есть причины. Во-первых, в ИБЗ переменные (x_1, x_2, \dots, x_N) как правило не количественные, а лингвистические, то есть их значениями выступают не числа, а слова и предложения естественного или искусственного языка. Во-вторых, связи между переменными выражаются не в виде математических уравнений, а часто задаются с помощью лингвистических, логических или словесных выражений. В-третьих, критерии выбора формулируются не в виде точного математического функционала, а описываются качественными формулировками, например в виде текстовых указаний по предпочтительности, недопустимости или желательности того или иного варианта решения.

Поэтому в ИБЗ, наряду с традиционными количественными, реализуются специфические методы поиска решений, ориентированные на качественное (понятийное) описание компонентов решений, связей между ними и критериев выбора.

5.1. ПОИСК РЕШЕНИЙ НА СЕМАНТИЧЕСКИХ СЕТЯХ

Сущность подхода к поиску решений, реализуемого этим методом, состоит в следующем. Для рассматриваемой проблемной области выделяется группа качественных характеристик, существенных с точки

зрения принятия решений, и перечисляются их возможные лингвистические значения. Используя эти характеристики, формируют те или иные высказывания, описывающие проблемную ситуацию. Среди таких высказываний могут оказаться как допустимые, так и недопустимые. К недопустимым относятся высказывания либо не имеющие смысла в данной проблемной области, либо очевидно нецелесообразные для конкретной ситуации. Все остальные высказывания считаются допустимыми и образуют в совокупности базу предметных знаний.

База предметных знаний формализуется с помощью семантической сети, вершинами которой являются имена характеристик, а дугами – их возможные значения. Формулировка задачи поиска решения на такой сети осуществляется путем фиксации значений характеристик, описывающих условия, в которых принимается решение, и указания характеристик, значения которых необходимо определить. Искомые значения характеристик определяются в два этапа. Вначале выявляются допустимые решения, то есть такие решения, характеристики которых удовлетворяют заданным условиям. Затем (если решение неоднозначное) с помощью некоторой системы предпочтений отыскивается рациональное решение из числа допустимых. Система предпочтений образует базу процедурных знаний, которая позволяет с использованием одной и той же семантической сети решать самые различные задачи поиска решений в данной проблемной области, в том числе и взаимосвязанные. Иными словами, реализуется классическая двухэтапная схема принятия системного решения, когда вначале анализируется обстановка и указывается, что можно и чего нельзя делать, а затем (если это возможно) определяется, что лучше делать.

Поясним сказанное на примере описания ситуации из военной области. «Зеленые» осуществляют высадку воздушного десанта для захвата и уничтожения тылового объекта «синих». «Синие», занимая круговую оборону объекта, создали подвижную оперативную и резервную группы, которые могут применяться как для уничтожения десанта, так и для сковывания его действий до подхода сил старшего начальника путем атаки с ходу или путем занятия обороны. Боевые действия могут вестись на маршруте выдвижения десанта после сбора и приведения его подразделений в боевую готовность, а также в районе высадки десанта до и после приведения его подразделений в боевую готовность. «Синие» получили сигнал оповещения о высадке воздушного десанта «зеленых» и приказ на начало боевых действий. Войска «синих», находящиеся в данном районе, приведены в полную боевую готовность. Личный состав занял оборонительные позиции и посты по боевому расписанию. Подвижная и резервная группы выдвинуты в угрожаемые районы. Командование «синих» уточняет оперативную обстановку и должно принять решение на боевые действия.

Введем характеристики, описывающие за «синих» рассматриваемую ситуацию, и их возможные (для данного примера) значения: источники действий (X_1): подвижная оперативная группа – a_1 ; резервная группа – a_2 ; цели действий (X_2): сковывание сил десанта – b_1 ; уничтожение десанта – b_2 ; способы действий (X_3): оборона – c_1 ; атака с ходу – c_2 ; место совершения действий (X_4): на маршруте выдвижения десанта – d_1 ; в районе высадки десанта – d_2 ; время совершения действий (X_5): до приведения десанта в боевую готовность – e_1 ; после приведения десанта в боевую готовность – e_2 .

Таким образом, для описания рассматриваемой ситуации вводятся пять лингвистических переменных X_1, X_2, X_3, X_4, X_5 , каждая из которых может принимать по два значения. Естественно, что для полного описания реальной ситуации этих данных недостаточно, однако, учитывая иллюстративный характер примера, мы ограничимся таким минимальным объемом информации. Лингвистические уравнения связи между введенными переменными, соответствующие соотношениям (5.2) и определяющие допустимые комбинации значений переменных, будем формировать, задавая невозможность или нецелесообразность сочетаний их значений. При этом все другие комбинации будем считать допустимыми.

Предположим, что в нашем примере невозможными (нецелесообразными) являются следующие сочетания значений: $X_4 = d_1, X_5 = e_1$ (боевые действия на маршруте выдвижения десанта до момента сбора и приведения его подразделений в боевую готовность); $X_2 = b_2, X_3 = c_1$ (уничтожение десанта путем ведения обороны); $X_1 = a_1, X_2 = b_2$ (использование резервной группы для сковывания сил десанта). На практике формирование таких условий осуществляется исходя из результатов оценки (уточнения) складывающейся оперативной обстановки (за себя и за противника) и на основе указаний, поступивших от старшего начальника. Кроме того, используются нормативные данные по боевым возможностям сил и тактико-техническим характеристикам применяемого вооружения.

Допустимые сочетания переменных X_1, X_2, X_3, X_4, X_5 представлены в виде семантической сети на рис. 5.1.

Каждой цепочке D_x ($x = 1, 2, \dots, 11$) этой сети можно поставить в соответствие некоторое осмысленное в рассматриваемой ситуации высказывание, связанное с физически реализуемым действием. Например, цепочке $D_0 = (a_1, b_1, c_1, d_1, e_2)$, отмеченной на рисунке жирной линией, соответствует следующее высказывание: подвижная оперативная группа (a_1) с целью сковывания сил десанта (b_1) занимает оборону (c_1) на маршруте выдвижения десанта (d_1) после приведения подразделений десанта в боевую готовность (e_2). Здесь понятия «с целью», «занимает» являются вспомогательными и представляют собой фактически имена

лингвистических переменных, выраженные в соответствии с грамматикой русского языка. Их использование придает фразе соответствующую смысловую окраску.

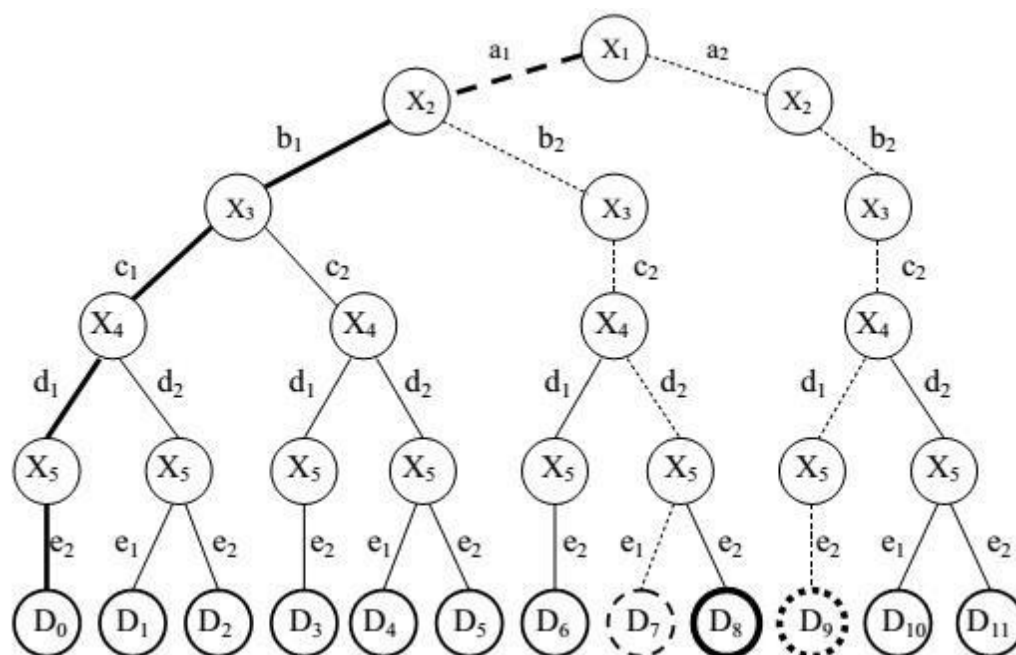


Рис. 5.1. Семантическая модель поиска решений (пример)

Фиксируя определенные значения некоторых переменных X_1, X_2, X_3, X_4, X_5 , то есть задавая «вход» ситуации принятия решения, можно определить ее «выходы» путем вычисления значений некоторых или всех остальных переменных с помощью введенных выше отношений недопустимости сочетаний значений переменных. При этом процедура поиска решения формально сводится к фиксации определенных звеньев семантической сети и нахождению всех допустимых цепочек, проходящих через эти звенья. Далее введем какой-либо критерий предпочтительности и с его использованием из всех допустимых цепочек выберем одну, соответствующую рациональному решению.

Сформулируем теперь одну из возможных задач поиска решений в рассматриваемой ситуации.

Задача 1. Определить действия «синих» (D^1_x), которые они должны осуществить для уничтожения воздушного десанта «зеленых», если известно, что «зеленые» уже завершили высадку десанта, их подразделения собрались в районе сосредоточения и готовы к боевым действиям.

В формализованном виде такая задача формулируется так: необходимо определить такие цепочки $D^1_x = (X_1, X_2, X_3, X_4, X_5)$ семантической сети, для которых $X_2 = b_2$ и $X_5 = e_2$. Здесь переменные X_2, X_5 – «входы», а переменные X_1, X_4, X_4 – «выходы» ситуации поиска решений. Как видно из рисунка, существует четыре таких цепочки, то есть

четыре допустимых решения: $D^1_6 = (a_1, b_2, c_2, d_1, e_2)$; $D^1_8 = (a_1, b_2, c_2, d_2, e_2)$; $D^1_9 = (a_2, b_2, c_2, d_1, e_2)$; $D^1_{11} = (a_2, b_2, c_2, d_2, e_2)$. Для выявления из найденных допустимых решений рационального варианта необходимо сформулировать соответствующий критерий выбора. Введем следующий критерий: лучшим (наиболее рациональным) из числа допустимых считается такое решение D_x , которое ближе всех находится к некоторому, заранее сформулированному, эталонному решению E . За меру близости между решениями D_x и E примем величину $\rho(D_x, E)$, равную числу несовпадающих значений одноименных компонентов X_i (возможно, взвешенных некоторыми весами $\alpha_i \geq 0$, $\sum \alpha_i = 1$). Нетрудно проверить, что такая мера удовлетворяет всем аксиомам метрики и имеет простой физический смысл: чем меньше несовпадений значений одноименных компонентов решений, тем меньше отличаются эти решения. Формально можно записать:

$$D_{\text{opt}} = \text{Arg min}_{x \in X} [\rho(D_x, E)], \quad (5.4)$$

где X – номера допустимых по условию задачи цепочек семантической сети (в нашем примере $X = 6, 8, 9, 11$).

Эталонные решения E формулируются исходя из: а) имеющегося опыта решения подобных задач на учениях, на тренировках и в ходе предшествующих боевых действий; б) в соответствии с указаниями, содержащимися в решении старшего начальника; в) на основе использования общих положений различных регламентирующих документов (уставов, наставлений, руководств и т.п.). Возможны иные пути определения эталонных решений и использование других критериев выбора рациональных решений. Для рассматриваемого метода это не принципиально. Важен принцип: в бою не бывает тупиковых ситуаций, если ситуация назрела, то решение должно быть принято. В том случае, когда не представляется возможным указать эталонные решения, на базе той же семантической сети можно построить другую схему поиска решений, позволяющую применительно к конкретной ситуации последовательно ответить на вопросы типа: что категорически запрещено делать, чего не рекомендуется делать, чего лучше не делать и что надо обязательно делать. В этом случае вместо эталонных решений вводятся запрещающие, предупреждающие, рекомендуемые и предписывающие правила, и производится соответствующая «раскраска» дуг сети.

Предположим теперь, что в качестве эталонного из каких-либо соображений выбрано следующее утверждение: «наибольший успех в разгроме воздушного десанта противника достигается при атаке с ходу силами подвижных оперативных групп в период выброски, сбора и приведения его подразделений в боевую готовность». Для рассматриваемой задачи I это означает, что эталонное решение $E^1 = D^7 = (a_1, b_2, c_2, d_2, e_1)$. На рисунке цепочка, соответствующая этому эталонному решению, показана штрихованной линией. Расстояние каждого из допусти-

мых решений $D^1_6, D^1_8, D^1_9, D^1_{11}$ до рационального будет $\rho(D^1_6, E^1) = 2$, $\rho(D^1_8, E^1) = 1$, $\rho(D^1_9, E^1) = 3$ и $\rho(D^1_{11}, E^1) = 2$. Следовательно, решение D^1_8 меньше всех отличается от эталона, и в соответствии с введенным выше критерием, является более предпочтительным. Суть его состоит в том, что рациональным действием «синих» при разгроме воздушного десанта «зеленых» после сбора и приведения его подразделений в боевую готовность является атака с ходу силами подвижной оперативной группы в районе десантирования.

Отметим некоторые особенности поиска решений на семантических сетях, когда рассматриваются не отдельные задачи, а совокупности взаимосвязанных задач. Для простоты рассмотрим решение двух взаимосвязанных задач, полагая, что аналогичные рассуждения справедливы при решении k ($k > 2$) взаимосвязанных задач. Пусть одной из таких задач является задача 1, а другой – задача 2, полностью совпадающая с ней по условию и отличающаяся лишь эталонным решением. Так как для описания этих задач используется одна и та же семантическая сеть, то условия второй задачи могут быть формализованы аналогично тому, как это делалось в первой задаче, но с заменой переменных X на Y . А именно, необходимо определить все такие цепочки $D^2_y = (Y_1, Y_2, Y_3, Y_4, Y_5)$ семантической сети, для которых $Y_2 = b_2$, $Y_5 = e_2$. В качестве эталона для задачи 2 выберем решение $E_2 = (a_2, b_2, c_2, d_1, e_2)$. В этом случае решение задачи очевидно. Согласно введенному критерию, рациональным считается решение D^2_9 , поскольку оно полностью совпадает с эталонным – $\rho(D^2_9, E_2) = 0$.

Предположим теперь, что боевая обстановка складывается так, что у «синих» для разгрома десанта «зеленых» имеется возможность привлечь силы какой-либо одной группы – подвижной оперативной или резервной. Это означает, что решения задач 1 и 2 связаны таким образом, что: $X_1 = Y_1$, то есть источники действий при реализации этих задач должны быть одними и теми же. Заметим, что полученные решения задач 1 и 2 не удовлетворяют условию $X_1 = Y_1$, так как значение первой компоненты решения $D^1_8 = a_1$, а решения $D^2_9 = b_2$, то есть при решении задачи 1 рациональным было признано применение сил подвижной оперативной группы, а задачи 2 – сил резервной группы. Следовательно, если учитывать условие $X_1 = Y_1$, то полученные ранее решения не являются совместно рациональными, и требуется поиск некоего компромисса, то есть необходимо произвести координацию этих решений.

Координацию решений задач 1 и 2 можно осуществить путем регулирования связи $X_1 = Y_1$. При этом под рациональным регулированием связи будем понимать выбор такой из ее допустимых реализаций, которая обеспечивает минимум (по реализациям) максимума (по задачам) отклонений рационального в каждой задаче решения от своего

эталона, или формально:

$$W = \min_{X_1=Y_1} \max[\min_{x \in X} \rho(D^1_x, E^1), \min_{y \in Y} \rho(D^2_y, E^2)]. \quad (5.5)$$

Как видно из приведенной записи, такой критерий позволяет найти компромиссное решение, которое в нашем случае интерпретируется следующим образом. Поскольку рациональные решения каждой задачи в отдельности достигаются при прямо противоположных стратегиях ($X_1 \neq Y_1$), то максимизация наименьших отклонений решения каждой задачи от своего эталона обеспечивает наибольшее приближение каждой отдельной задачи к своему рациональному решению. Минимизация же по стратегиям наибольших отклонений позволяет выбрать одну компромиссную стратегию, лучшую для двух задач в целом. В рассматриваемом примере возможны две стратегии: $X_1 = Y_1 = a_1$ и $X_1 = Y_1 = a_2$. Для первой стратегии существует два допустимых решения D_6 и D_8 , одно из которых D_8 – рациональное для задачи 1 и оба – нерациональные для задачи 2, причем $\rho_1(D_6, E^1) = 2$, $\rho_1(D_8, E^1) = 1$, $\rho_2(D_6, E^2) = 1$, $\rho_2(D_8, E^2) = 2$, $\max(\rho_1, \rho_2) = 2$. Для второй стратегии тоже существуют два допустимых решения D_9 и D_{11} , где D_9 – решение, рациональное для задачи 2 и оба – нерациональные для задачи 1, причем $\rho_1(D_9, E^1) = 3$, $\rho_1(D_{11}, E^1) = 2$, $\rho_2(D_9, E^2) = 2$, $\rho_2(D_{11}, E^2) = 1$, $\max(\rho_1, \rho_2) = 3$. Таким образом, в соответствии с введенным минимаксным критерием рациональным при условии $X_1 = Y_1$ следует признать первый вариант реализации связи ($X_1 = Y_1 = a_1$) с окончательным комплексным решением $D_8 = (a_1, b_2, c_2, d_2, e_2)$.

Заметим, что если без учета связи $X_1 = Y_1$ отклонения рациональных решений от эталонов в задачах 1 и 2 составляли соответственно $\rho_1 = 1$, $\rho_2 = 0$, то в результате регулирования связи эти отклонения будут соответственно $\rho_1 = 1$, $\rho_2 = 2$. Следовательно, в данном случае координирование реализовано путем увеличения показателя ρ_2 в задаче 2, для которого он при раздельном решении был наименьшим. Именно в этом состоит сущность введенного критерия координации: проявление осторожности в оценке ситуаций и в сбалансированном учете интересов участников коалиции (в нашем случае задач).

На практике (помимо рассмотренного минимаксного критерия) используются другие критерии координации. В частности, возможно применение так называемого усредняющего критерия, по которому в качестве рациональной реализации связи выбирается значение, обеспечивающее минимизацию среднего (по задачам) отклонения от эталонов. В отличие от минимаксного критерия, использование усредняющего критерия связано с известным риском: плохой текущий выбор может быть оценен как хороший в среднем.

Метод поиска решений на семантических сетях, реализованный в ИБЗ, позволяет решать достаточно широкий класс слабо формализуе-

мых задач. Его реализация связана с разработкой естественного проблемно ориентированного языка, а также с созданием декларативных и предметных баз знаний в рассматриваемой проблемной области. Другой важной особенностью этого метода является присутствие пользователя как на этапе обучения (пополнение модели знаний, уточнение компонентов решений, ввод эталонных решений и т.п.), так и на этапе решения конкретных задач (передача управления очередному блоку, изменение эталонных решений, уточнение постановок задач и т.п.).

5.2. СИТУАЦИОННЫЙ ПОИСК РЕШЕНИЙ

В своей основе ситуационный поиск решений представляет собой удачную попытку реализовать на компьютере рефлексные механизмы мышления человека. Идея метода иллюстрируется схемой, приведенной на рис. 5.2. Пусть в каждый текущий момент времени мы можем

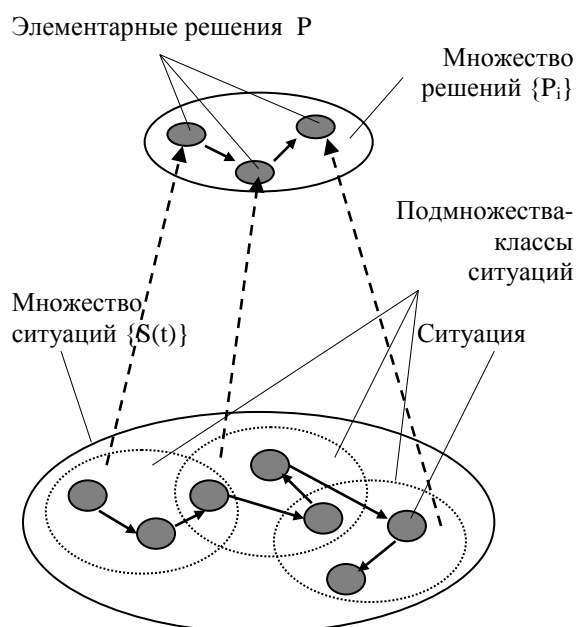


Рис. 5.2. Идея ситуационного поиска решений

получить неформальное описание складывающейся ситуации, отражающей: состояние объекта управления $\{S_1\}$; состояние управляющего объекта $\{S_2\}$ и текущие цели управления $\{C\}$. Обозначим это описание через $S(t)$. Предполагается, что информации, содержащаяся в $S(t)$, вместе с той информацией, которая уже имеется в памяти управляющего объекта, достаточно для принятия решения по управлению, которое формируется из заданного множества элементарных решений $\{p_i\}$. Решение при этом рассматривается как некоторая цепочка, состоя-

щая из элементарных решений, или дерево, путь по которому определяется множеством возникающих при управлении ситуаций на входе управляющего объекта. Тогда для каждого момента времени t можно рассмотреть элементарную задачу определения того решения $P \in \{p_i\}$, которое необходимо принять в момент времени t при наличии ситуации $S(t)$. Это означает, что в процессе обучения необходимо на множестве $\{S(t)\}$ выделить такие подмножества-классы, каждому из которых соответствовало бы некоторое элементарное решение $P \in \{p_i\}$.

Отметим две особенности такой постановки задачи. Во-первых, поскольку мы имеем дело с неформальным описанием ситуаций, то не

представляется возможным четко выделить подмножества-классы в $\{S(t)\}$, соответствующие элементарным решениям. В реальных задачах границы между подмножествами-классами размыты и, фактически, можно говорить не о разбиении множества $\{S(t)\}$, а о его покрытии. Во-вторых, учитывая, что в практических задачах мощность множества $\{S(t)\}$ намного превышает мощность множества $\{p_i\}$, обобщение текущих ситуаций в подмножества-классы следует проводить постепенно, а полезность каждого промежуточного обобщения должна проверяться путем сравнения с экспертными решениями по этому вопросу.

В реализации ситуационного поиска решения можно выделить следующие основные стадии: описание ситуаций, обобщение ситуаций и выбор решения.

Описание ситуаций. Для описания ситуаций используется модификация языка семантических сетей, названная языком ситуационного управления [Поспелов, 1975]. Его лексику образуют: базовые понятия $\{v\}$; базовые отношения $\{r\}$; имена $\{i\}$; элементарные решения $\{p\}$ и оценки $\{O\}$. Из элементов этих множеств по определенным правилам грамматики (подобным правилам, рассмотренным в разделе 8.3) конструируются тексты, дающие описание входных ситуаций $S(t)$. Множество базовых понятий состоит из физических понятий и понятий-классов. Физические понятия задаются набором признаков, значения которых определяют эти понятия (например, совокупностью показаний некоторых приборов, установленных на объекте управления). Будем обозначать этот набор как $\{\pi_1, \pi_2, \dots, \pi_n\}$. Понятия-классы есть просто совокупности физических понятий, однородные с некоторой точки зрения, например, система, комплекс, средство, элемент и т.п. Отнесение того или иного физического понятия к понятию-классу является для модели внешней функцией, и правила этого отнесения в модели не формируются. Имена служат для персонификации элементов, входящих в некоторое понятие-класс. Отношения r устанавливают связи между элементами множеств $\{\pi\}$, $\{v\}$, $\{i\}$, $\{p\}$. Таким образом, описание ситуации с использованием такого языка представляет собой некоторую структуру, составленную из элементов множеств $\{\pi\}$, $\{v\}$, $\{i\}$, $\{p\}$, связанных отношениями $\{r\}$. Возможности этого языка продемонстрируем на упрощенном примере описания следующей ситуации. Десятый мотострелковый батальон (мсб) «зеленых» занимает оборону населенного пункта «О». Пятой механизированной бригаде (мбр) «синих», усиленной вертолетами огневой поддержки, поставлена задача захватить населенный пункт «О». Первый мотострелковый (мсб) и второй танковый (тб) батальоны пятой мбр «синих» находятся на марше и приближаются к зонам действий минометных и противотанковых средств «зеленых». Вертолеты огневой поддержки «синих» вошли в зону действия зенитных ракет «зеленых».

Введем обозначения. Понятия-классы: v_1 – мсб, v_2 – «синие», v_3 – населенный пункт, v_4 – мбр, v_5 – «зеленые», v_6 – вертолеты огневой поддержки, v_7 – тб, v_8 – марш (движение), v_9 – зона действия противотанковых средств, v_{10} – зона действия зенитных ракет, v_{11} – зона действия минометных средств, v_{12} – мпб. Имена: i_1 – «10», i_2 – «О», i_3 – «5», i_4 – «1», i_5 – «2». Отношения: r_1 – «иметь имя», r_2 – «находиться на (в)», r_3 – «быть средством усиления», r_4 – «приближаться к», r_5 – «принадлежать к классу», r_6 – «одновременно», r_7 – «оборонять», r_8 – «захватывать». Тогда с учетом введенных обозначений рассматриваемая ситуация может быть описана выражением: $S(t) = ((v_1 r_5 v_5)(v_1 r_1 i_1)(v_1 r_7 v_3 r_1 i_2)) \wedge ((v_4 r_5 v_2)(v_4 r_1 i_3)(v_4 r_8 v_3 r_1 i_2) (v_6 r_3 v_4)) \wedge (((v_{12} r_5 v_2)(v_{12} r_1 i_4)(v_{12} r_4 v_{10})(v_{12} r_2 v_8)) r_6 ((v_7 r_5 v_2)(v_7 r_1 i_5) (v_7 r_4 v_9)(v_7 r_2 v_8))) r_6 (v_6 r_2 v_{11})$.

Использование подобных структур исключает неоднозначность понимания описываемых ситуаций (характерное для естественного языка). Они «понятны» компьютеру, то есть могут легко записываться, храниться и обрабатываться, образуя базу предметных знаний. Кроме того, такие тексты допускают формальные преобразования своих конструкций, в соответствии с грамматикой данного языка, что позволяет выводить новые понятия и обобщать ситуации (разделять множество ситуаций на подмножества-классы).

Обобщение ситуаций. При наличии обозримого множества ситуаций $\{S(t)\}$ можно было бы поставить каждой из них соответствующее решение. Однако в практически интересных случаях мощность множества $\{S(t)\}$ настолько велика, что нет никакой надежды на сопоставление «ситуация – решение». Одним из способов уменьшения количества ситуаций является их обобщение. Именно поэтому проблема обобщения ситуаций является центральной в ситуационном поиске решений. В настоящее время практическое применение нашли три способа обобщения ситуаций. Самым простым из них является обобщение по именам: к одному классу ситуаций относятся все ситуации, описание которых отличается лишь именами, стоящими на определенных местах, и которые требуют при своем возникновении однотипных решений. Вторым способом обобщения ситуаций является формирование функций принадлежности к классу ситуаций на основе значений признаков $\{\pi\}$, которые рассматриваются как лингвистические переменные с соответствующими функциями принадлежности. С использованием операций над функциями принадлежности строятся нечеткие алгоритмы распознавания типов ситуаций. Третьим способом является обобщение ситуаций по структуре отношений, которыми они описываются. Суть этого способа сводится к выделению в формализованных описаниях ситуаций, некоторых типовых структур (фреймов), которыми и определяются обобщенные ситуации. Пример такого способа формирования обобщенных ситуаций (понятий) рассматривался выше при описании

языка семантических сетей. Следует отметить, что при ситуационном поиске решений именно этот способ обобщения оказался наиболее продуктивным.

В общем случае обобщение ситуаций можно представить в виде многослойной структуры, между слоями которой имеются связи, указывающие переход от более мелких описаний к более крупным. Эти связи устанавливаются согласно изложенным способам обобщения и взвешиваются оценками O_j , значения которых вводятся пользователем при обучении ИБЗ.

Выбор решения осуществляется с помощью специальных корреляционных правил (или правил вывода), имеющих в простейшем случае следующий вид: $S^1(t) \Rightarrow p_1$. Смысл этого правила состоит в том, что при наличии ситуации $S^1(t)$ необходимо принять решение p_1 . В частности, для рассмотренного выше примера можно ввести корреляционное правило: $(v_6 \ r_2 \ v_{11}) \Rightarrow p_1$, которое означает, что при наличии в описании исходной ситуации структуры, стоящей в левой части правила или ее эквивалента (с точностью до формальных преобразований, допускаемых грамматикой данного языка), необходимо принять решение p_1 . В том случае если p_1 означает открыть огонь зенитными ракетами, то на естественном языке введенное корреляционное правило читается следующим образом: если вертолеты огневой поддержки «синих» вошли в зону действия зенитных ракет «зеленых», то зенитному взводу необходимо дать команду на открытие огня. Корреляционные правила вводятся в модель на этапе обучения и в совокупности образуют базу ее процедурных знаний. Помимо предписывающих, возможно использование рекомендующих (если $S^2(t)$, то лучше p_1 , но можно p_2 или p_3) и запрещающих корреляционных правил (если $S^3(t)$, то нельзя p_1, p_2, \dots, p_N), а также их комбинаций.

5.3. МНОГОКРИТЕРИАЛЬНЫЙ ПОИСК РЕШЕНИЙ

Пусть имеется некоторый объект, функционирование которого зависит от вектора управляемых параметров $\bar{X} = (x_1, x_2, \dots, x_m)$, а его эффективность оценивается вектором численных показателей $\bar{Q} = (q_1, q_2, \dots, q_s)$, причем $q_i = q_i(\bar{X}, \bar{\omega})$, $i = \overline{1, S}$ ($\bar{\omega}$ – вектор неуправляемых параметров). Тогда в математическом виде задачу многокритериального поиска решений можно выписать следующим образом:

$$\begin{aligned} \bar{Q} = (q_1, q_2, \dots, q_s) &\xrightarrow{\bar{x} \in D} \text{opt} \\ D: q_i &= q_i(\bar{X}, \bar{\omega}), i = \overline{1, S}, \\ f_p(\bar{X}, \bar{\omega}) &\leq 0, p = 1, 2, 3, \dots \end{aligned} \quad (5.6)$$

Символом D обозначена область, определяющая область поиска

допустимых решений \bar{X} . Обычно она задается ограничениями типа равенства, неравенства, дискретности или функциональной связи. Выражение $\bar{Q} = (q_1, q_2, \dots, q_s) \xrightarrow{\bar{x} \in D} \text{opt}$, где $\text{opt} = \max$ или \min есть критерий оптимизации. Выбор \max или \min зависит от решаемой задачи и смысла показателей $\bar{Q} = (q_1, q_2, \dots, q_s)$. Неравенства $f_p(\bar{X}, \bar{\omega}) \leq 0$ определяют функциональную связанность управляемых параметров, и указывают на то, что значения величин (x_1, x_2, \dots, x_m) нельзя выбирать произвольным образом.

Рассмотрим, как решаются подобные задачи в прикладных модулях ИБЗ. Для наглядности допустим, что в задаче используются два показателя $q_1(x)$ и $q_2(x)$, которые необходимо минимизировать. Область допустимых решений зададим отрезком действительных чисел $x \in [a, b]$. Характер изменения значений $q_1(x)$ и $q_2(x)$ в нормированном масштабе приведен на рис. 5.3, который показывает, что минимумы по каждому

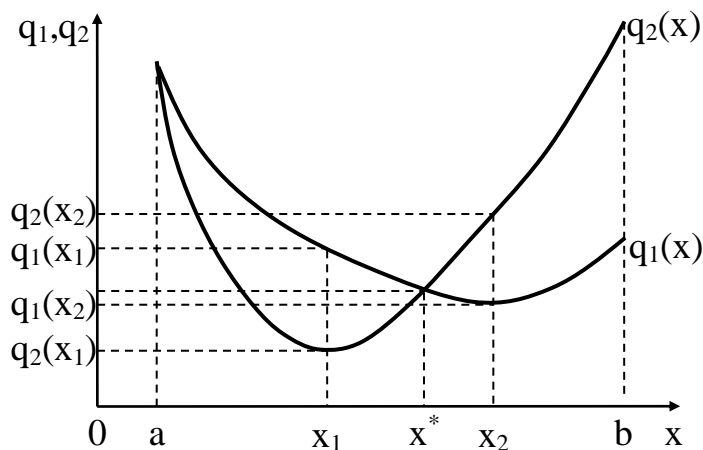


Рис. 5.3. Вид $q_1(x)$ и $q_2(x)$, $x \in [a, b]$

из показателей достигаются в точках x_1 и x_2 соответственно для $q_1(x)$ и $q_2(x)$. Для $x \in [a, x_1]$ и $x \in [x_2, b]$ показатели $q_1(x)$ и $q_2(x)$ ведут себя согласованно, одновременно уменьшаясь либо увеличиваясь.

Рассматривая область допустимых решений $x \in [x_1, x_2]$ видим, что уменьшение значений показателя $q_1(x)$ ведет к уве-

личению значений показателя $q_2(x)$, то есть показатели как бы конфликтуют. Этот «конфликт» отображается в пространстве показателей $\{q\}$ в определенную область (рис. 5.4), которая называется множеством Парето. Решения $\bar{X} \in D$, которые определяют множество Парето, называются не худшими. Множество не худших решений обозначим через M_0 . В общем случае для векторной модели оптимизации вводится правило, позволяющее оценивать решения, — безусловный критерий предпочтения (БКП). Решение x_2 безусловно лучше решения x_1 ($x_2 \succ x_1$) в смысле векторного показателя \bar{Q} , если $q_i(\bar{X}_2) \leq q_i(\bar{X}_1)$ для всех i , и хотя бы одно неравенство строгое. Если все $q_i(\bar{X}_2) = q_i(\bar{X}_1)$, то решение \bar{X}_2 эквивалентно решению \bar{X}_1 ($\bar{X}_2 \sim \bar{X}_1$).

Из всего множества D допустимых решений БКП выделяет подмножество M_0 , не худших решений, определяющих множество Парето.

Другими словами, оператор БКП реализует принцип оптимизации по Парето. Действительно, для $\forall x \in [a, x_1]$, $x_1 > x$, так как $q_2(x_1) < q_2(x)$ и $q_1(x_1) < q_1(x)$. То же самое – $x_2 > x \forall x \in [x_2, b]$. В области же $x^* \in [x_1, x_2]$ любые x конфликтуют, например x_1 и $x^* \in [x_1, x_2]$, поскольку $q_1(x_1) > q_1(x^*)$, а $q_2(x_1) < q_2(x^*)$.

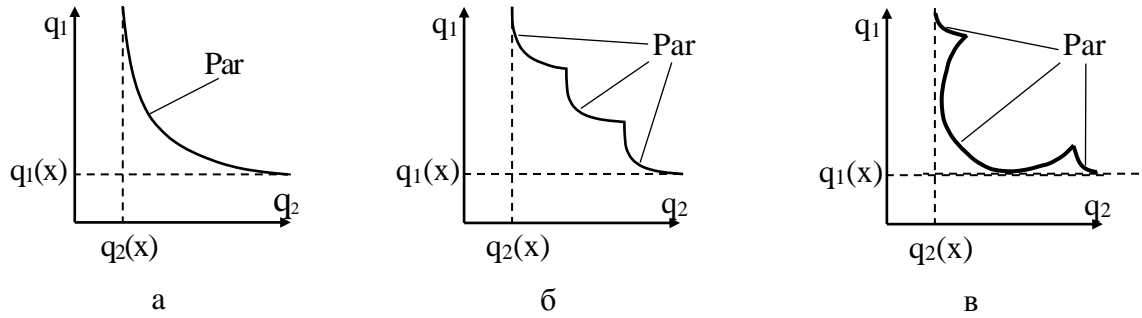


Рис. 5.4. Примеры множества Парето (Par):

а – непрерывное; б – несвязное; в – с изолированными точками

Следовательно, итогом многокритериального поиска решений формально можно считать нахождение множества M_0 – не худших решений. В частных случаях для этих целей используются следующие формулы:

– для выпуклых областей D и выпуклых $q_i(x)$, $i = \overline{1, S}$

$$\text{Par} = \{ \bar{q} (x^*): \sum_{i=1}^S \alpha_i q_i(\bar{x}^*) = \min_{x \in D} \sum_{i=1}^S \alpha_i q_i(\bar{x}) \};$$

– для не выпуклых областей D

$$\text{Par} = \{ \bar{q} (x^*): \max_i \alpha_i q_i(\bar{x}^*) = \min_{x \in D} \max_i \alpha_i q_i(\bar{x}) : q_i(\bar{x}) \geq 0 \}, \text{ при}$$

$$\text{всех } \bar{\alpha} \in A = \{ (\alpha_1, \alpha_2, \dots, \alpha_S): \sum_{i=1}^S \alpha_i = 1, \alpha_i > 0 \}.$$

Результат в виде множества M_0 допускает множество решений, и в этом смысле является неоднозначным. Естественнo предположить, что окончательные решения следует искать среди элементов множества M_0 . Поэтому следующий шаг решения задачи (5.6) заключается в сужении множества M_0 . В каждом случае такое сужение следует проводить с учетом целей, поставленных пользователем ИБЗ. Наиболее простым способом получения одного конкретного решения из множества Парето является сведение задачи векторной оптимизации к задаче скалярной оптимизации путем выделения одного критерия (главного) и переводом остальных в разряд ограничений. Другой, не менее распространенный способ, заключается в построении глобального критерия в виде мультипликативной, аддитивной, иерархической или какой-либо иной

свертки частных показателей.

Таким образом, многокритериальный поиск решений осуществляется в два этапа — это выделение области компромиссов (поиск решений оптимальных по Парето), а затем ее сужение на основе некоторой схемы компромиссов в частном случае до единственного решения, оптимального с точки зрения лица, принимающего решение.

5.4. ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ

Общая идея построения генетических алгоритмов примерна та же, что и нейросетевых моделей — паллиативный подход. Отличие заключается в том, что эти алгоритмы имитируют функционирование генных структур в процессе эволюции живых организмов. Генетические алгоритмы — это процедуры поиска, основанные на механизмах естественного отбора и наследования. В них используется эволюционный принцип выживания наиболее приспособленных особей. Они отличаются от традиционных методов оптимизации несколькими базовыми элементами. В частности, генетические алгоритмы: обрабатывают не значения параметров самой задачи, а их закодированную форму; осуществляют поиск решения исходя не из единственной точки, а из их некоторой популяции; используют только целевую функцию, а не ее производные либо иную дополнительную информацию; применяют вероятностные, а не детерминированные правила выбора.

Стандартный генетический алгоритм, был впервые подробно описан и исследован в работах де Джонга. Он проанализировал простую схему кодирования генов битовыми строками фиксированной длины и соответствующих генетических операторов, выполнил значительное количество численных экспериментов, сравнивая результаты с оценками, предсказываемыми биологами-генетиками. Как показали последующие исследования, данный метод оказался эффективным не только при исследовании биолого-генетических проблем, но и при поиске решений в ИБЗ.

В своем простейшем варианте генетический алгоритм строиться следующим образом.

Задается функция оптимальности $f(x)$, определяющая эффективность каждого найденного решения (найденной комбинации признаков). Формируемое решение кодируется как вектор $(x_1, x_2, \dots, x_\lambda)$, который называется «хромосома» и соответствует битовой маске, то есть двоичному представлению набора исходных переменных. В хромосоме выделяются части вектора — «гены», изменяющие свои значения в определенных позициях — «аллелях».

В соответствии с заранее определенными условиями инициализируется исходная «популяция» $P^0(x_1^0, x_2^0, \dots, x_\lambda^0)$ потенциальных решений – совокупность решений на конкретной итерации, состоящая из некоторого количества хромосом λ , число которых задаётся изначально и в процессе перебора обычно не изменяется.

Каждая хромосома $x_i, i = 1, \dots, \lambda$ в популяции декодируется в форму, необходимую для последующей оценки, и ей присваивается значение эффективности в соответствии с вычисленной функцией оптимальности. Кроме того, каждой хромосоме присваивается вероятность воспроизведения $Q(x_i), i = 1, \dots, \lambda$, которая зависит от эффективности данной хромосомы. Существуют различные критерии отбора, самый популярный из них – $\max f(x_i) / \sum_{j=1}^{\lambda} f(x_j)$. В соответствии с этим критери-

ем создается новая популяция хромосом. Хромосомы производят потомков, используя операции рекомбинации: кроссинговер (хромосомы скрещиваются, обмениваясь частями строк (рис. 5.5.)) и мутация (вероятностное изменение аллелей (рис. 5.6)).

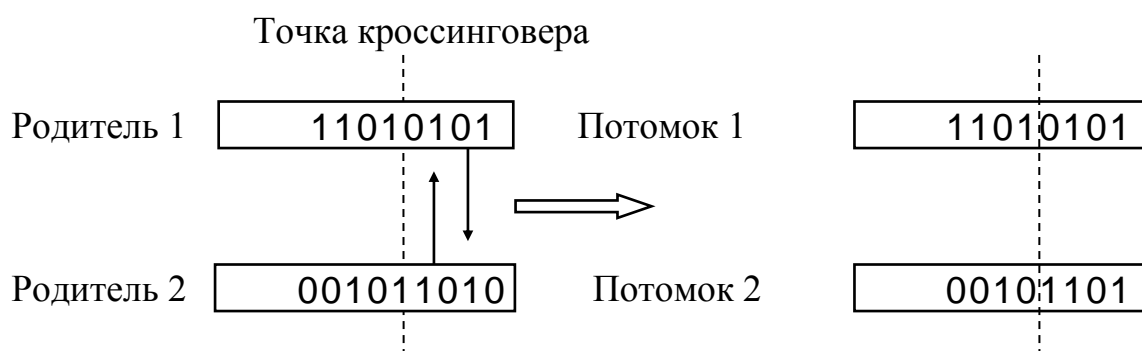


Рис. 5.5. Одноточечная схема кроссинговера



Рис. 5.6. Мутация

Процесс прекращается, если получено удовлетворительное решение, либо исчерпано отведенное на эволюцию время. Если процесс не окончен, то вновь повторяются процессы оценки и воспроизведения новой популяции.

Операция воспроизведения служит для создания следующей популяции на основе предыдущей при помощи операторов кроссинговера и мутации, которые имеют случайный характер. Каждой хромосоме промежуточной популяции X_i^t в случае необходимости подбирается партнёр и созданная хромосома помещается в результирующую популяцию. Оператор кроссинговера производит скрещивание хромосом и обмен генетическим материалом между родителями для получения потомков. Этот оператор служит для формирования новых областей популяционного пространства и улучшения существующих популяций (эволюционное приспособление). Простейший односточный кроссинговер производит обмен частями, на которые хромосома разбивается точкой кроссинговера, выбираемой случайно. Двухточечный кроссинговер обменивает кусок строки, попавшей между двумя точками. Предельным случаем является равномерный кроссинговер, в результате которого все биты хромосом обмениваются с некоторой вероятностью. Оператор мутации применяется к каждому биту хромосомы с небольшой вероятностью ($p_i \approx 0,001$), в результате чего бит (аллель) изменяет значение на противоположное. Мутация нужна для расширения пространства поиска и предотвращения невозвратимой потери бит в аллелях.

В генетических алгоритмах присутствует также оператор воспроизведения, называемый «инверсия», который заключается в реверсировании аллелей между двумя случайными позициями, однако для большинства задач он не имеет практического смысла и поэтому малоэффективен. Обобщенная блок-схема генетического алгоритма при его реализации в задачах поиска решений в ИБЗ представлена на рис. 5.6.

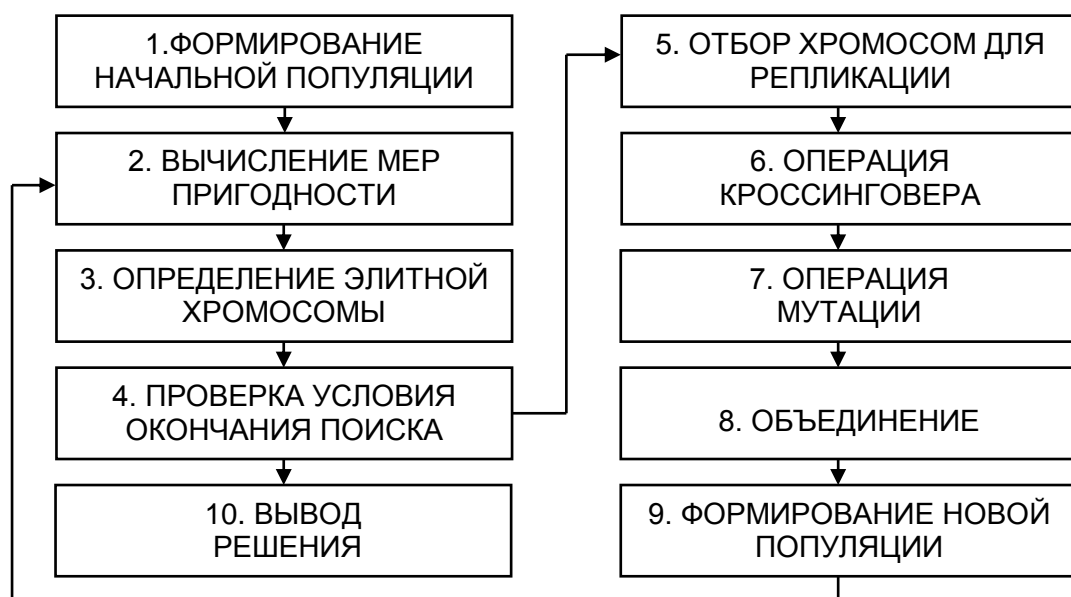


Рис. 5.6. Блок-схема генетического алгоритма

Решения, найденные с помощью генетического алгоритма, нельзя признать оптимальными в обычном – математическом смысле, но это не мешает применять этот метод для поиска решений в ИБЗ различного функционального назначения.

6. МЕТОДЫ ОЦЕНКИ КАЧЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ

6.1. КРИТЕРИИ КАЧЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ

Социальные критерии, характеризуют социальные последствия использования ИБЗ в составе ИКС. Для оценки таких последствий пока не найдено универсальных шкал и методов их измерения (даже понятийных), поэтому для каждого класса ИБЗ используются свои специфические социальные критерии, отражающие особенности и характерные черты проблемной области и окружающей среды данного проекта.

Функциональные критерии характеризуют ИБЗ с точки зрения качества выполнения ею своей основной и дополнительных функций и отражают такие свойства ИБЗ как: функциональная полнота; структурная полнота; отсутствие структурного дублирования; отсутствие управленческого дублирования; информационная безызбыточность; информационная достаточность; ресурсная обеспеченность; ресурсная согласованность; структурная связность.

Эргономические критерии характеризуют степень удобства общения пользователей с техническими и программными средствами ИБЗ, необходимый уровень обученности обслуживающего персонала и уровень специальной подготовки конечных пользователей и операторов данной ИБЗ.

Прагматические критерии характеризуют так называемую действенность ИБЗ, то есть степень удовлетворения своего предназначения, и выражаются такими сравнительными категориями как «полное», «частичное», «условное».

Технические критерии характеризуют уровень технического совершенства ИБЗ и ее компонентов (подсистем, узлов, блоков, технических средств) и оцениваются с помощью лингвистических шкал типа «высокий – низкий» или таких категорий как «мировой уровень», «отечественный уровень».

Эксплуатационные критерии характеризуют ИБЗ и ее компоненты с точки зрения удобства проведения различных организационно-технических мероприятий (настроек, профилактик, регламентных работ и т.п.). Их оценка основывается на качественных шкалах типа: «удобно – неудобно», «доступно – недоступно» и других подобного характера.

Технологические критерии характеризуют уровень технологий, использованных при проектировании ИБЗ и ее компонентов, а так же уровень технологичности разработки спроектированной ИБЗ. Они оце-

ниваются с помощью лингвистических шкал типа «высокий – низкий» или таких категорий как «мировой уровень», «отечественный уровень».

Экономические критерии выражаются в стоимостных показателях проекта по созданию ИБЗ и ее составных частей. Однако не всегда представляется возможным оценить эффект внедрения ИБЗ по критерию «выгоды – затраты», а зачастую такая оценка и нецелесообразна. Это утверждение справедливо, прежде всего, для ИБЗ, входящих в состав крупномасштабных ИКС типа ГАС РФ «Правосудие». Для ИБЗ такого класса достаточно на качественном уровне определить ожидаемые эффекты их создания и разработать технико-экономическое обоснование проекта, подкрепив его, где это возможно и необходимо экономическими расчетами.

Критерии надежности и безопасности характеризуют ИБЗ с точки зрения надежность комплекса технических средств, своевременности представления информации пользователям, ее полноты и достоверности, сохранения конфиденциальности, защищенности от несанкционированного доступа и опасных программно-технических воздействий.

Перечень критериев качества применительно к проекту ИБЗ в составе ГАС РФ «Правосудие» приведен в табл. 6.1.

Таблица 6.1. Критерии качества ИБЗ (на примере ГАС РФ «Правосудие»)

Типы критериев	Содержание критериев
Социальные	Рост доверия к результатам судебных решений; повышение оперативности работы судов; открытость судопроизводства для средств массовой информации и граждан
Функциональные	Функциональная полнота; структурная полнота; отсутствие структурного дублирования; отсутствие управленческого дублирования; информационная безызбыточность; информационная достаточность; ресурсная обеспеченность; ресурсная согласованность; структурная связность
Эргономические	Пользовательский интерфейс; комфортные условия работы
Прагматические	Степень удовлетворения ИБЗ своего предназначения
Технические	Уровень технического совершенства ИБЗ и ее компонентов (подсистем, комплексов)
Технологические	Уровень технологий, использованных при проектировании ИБЗ и ее компонентов, а так же уровень технологичности разработки спроектированной системы

Экономические	Нормированная стоимость данного проекта по отношению к самому дорогому варианту
Эксплуатационные	Удобство проведения различных организационно-технических мероприятий (настроек, профилактик, регламентных работ и т.п.)
Надежность и безопасность	Надежность и своевременность представления информации пользователям; полнота выходной информации; достоверность информации; сохранение конфиденциальности информации; защищенность от несанкционированного доступа; защищенность от опасных программно-технических воздействий

6.2. ИНТЕГРАЛЬНАЯ ОЦЕНКА КАЧЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ

Суть задачи заключается в том, чтобы, зная оценки качества проекта по локальным критериям (указанным в табл. 6.1), получить некую обобщающую или интегральную оценку качества проектируемой ИБЗ целиком.

Для решения задачи введем следующие обозначения:

P_0 – интегральная оценка качества проектируемой ИБЗ;

P_1 – социальная эффективность ИБЗ, включающая следующие составляющие: рост доверия к результатам судебных решений (P_{11}); повышение оперативности работы судов (P_{12}); открытость судопроизводства для средств массовой информации и граждан (P_{13});

P_2 – функциональная эффективность ИБЗ, включающая следующие составляющие: функциональная полнота (P_{21}); структурная полнота (P_{22}); отсутствие структурного дублирования (P_{23}); отсутствие управленческого дублирования (P_{24}); информационная безызбыточность (P_{25}); информационная достаточность (P_{26}); ресурсная обеспеченность (P_{27}); ресурсная согласованность (P_{28}); структурная связность (P_{29});

P_3 – эргономическая эффективность ИБЗ, включающая следующие составляющие: качество пользовательского интерфейса (P_{31}); комфортность работы пользователя (P_{32});

P_4 – прагматическая эффективность ИБЗ, характеризующая степень соответствия ИБЗ своему предназначению;

P_5 – техническая эффективность ИБЗ, характеризующая уровень технических решений, принятых при ее создании;

P_6 – технологическая эффективность ИБЗ, характеризующая уровень технологий, которые предполагается использовать при проектиро-

вании ИБЗ и ее компонентов, а также уровень технологичности разработки системы;

P_7 – экономическая эффективность ИБЗ, нормированная стоимость данного проекта по отношению к самому дорогому варианту;

P_8 – эксплуатационная эффективность проекта ИБЗ, включающая следующие составляющие: удобство настроек (P_{81}); профилактик (P_{82}); регламентных работ (P_{83});

P_9 – надежность и безопасность ИБЗ, включающая, следующие составляющие: надежность предоставления информации (P_{91}); своевременность предоставления информации (P_{92}); полнота выходной информации (P_{93}); достоверность информации (P_{94}); сохранение конфиденциальности информации (P_{95}); защищенность от несанкционированного доступа (P_{96}).

Интегральная оценка на основе аддитивной свертки. Пусть имеются экспертные оценки указанных выше критериев и их составляющих. Пусть каждая из указанных оценок (и их составляющих) дана по шкале $[0,1]$ (1-высшая, 0-низшая оценка). Тогда аддитивная интегральная оценка качества проекта может быть получена по формуле:

$$P_0 \approx 0,1 \sum_{i=1}^9 \mu_i P_i, (\sum_{i=1}^9 \mu_i = 1), \quad (6.1)$$

где $\mu_i (i=1, \dots, 9)$ – коэффициенты относительной значимости локальных критериев качества проектов;

$$\begin{aligned} P_1 &\approx 0,33 \sum_{i=1}^3 \mu_{1i} P_{1i}; (\sum_{i=1}^3 \mu_{1i} = 1); P_2 \approx 0,11 \sum_{i=1}^9 \mu_{2i} P_{2i}; (\sum_{i=1}^9 \mu_{2i} = 1); \\ P_3 &= 0,5 \sum_{i=1}^2 \mu_{3i} P_{3i}; (\sum_{i=1}^2 \mu_{3i} = 1); P_8 = 0,25 \sum_{i=1}^4 \mu_{8i} P_{8i}; (\sum_{i=1}^4 \mu_{8i} = 1); \\ P_9 &= 0,17 \sum_{i=1}^6 \mu_{9i} P_{9i}; (\sum_{i=1}^6 \mu_{9i} = 1), \end{aligned} \quad (6.2)$$

где $\mu_{1i}, \mu_{2i}, \mu_{3i}, \mu_{8i}, \mu_{9i}$ – коэффициенты относительной значимости составляющих локальных критериев.

Достоинства метода – простота и наглядность, а недостатки связаны с тем, что высокая интегральная оценка некачественного проекта, может получиться за счет высокого значения только одному какому-то критерию при низких значениях остальных.

Интегральная оценка на основе мультипликативной свертки. Пусть, как и ранее, интегральная оценка проекта ИБЗ (P_0) определяется критериями $P_1 - P_9$, оценка каждого из которых (и их составляющих) дается по шкале $[0,1]$ (1-высшая, 0-низшая оценка). Тогда интегральная

оценка качества проекта методом мультипликативной свертки может быть получена с использованием следующей формулы:

$$P_0 = \prod_{i=1}^9 \mu_i P_i, (\sum_{i=1}^9 \mu_i = 1), \quad (6.3)$$

где $\mu_i (i=1, \dots, 9)$ – как и ранее, коэффициенты относительной значимости локальных критериев;

$$\begin{aligned} P_1 &= \prod_{i=1}^3 \mu_{1i} P_{1i}; (\sum_{i=1}^3 \mu_{1i} = 1); P_2 = \prod_{i=1}^9 \mu_{2i} P_{2i}; (\sum_{i=1}^9 \mu_{2i} = 1); \\ P_3 &= \prod_{i=1}^2 \mu_{3i} P_{3i}; (\sum_{i=1}^2 \mu_{3i} = 1); P_8 = \prod_{i=1}^4 \mu_{8i} P_{8i}; (\sum_{i=1}^4 \mu_{8i} = 1); \\ P_9 &= \prod_{i=1}^6 \mu_{9i} P_{9i}; (\sum_{i=1}^6 \mu_{9i} = 1), \end{aligned} \quad (6.4)$$

где все компоненты имеют тот же смысл, что и в предыдущем случае.

Особенность интегральной оценки на основе (6.3) и (6.4) заключается в том, что она обращается в нуль, если среди хотя бы одна из величин P_i или P_{ji} равна нулю. В нашем случае это означает, что рассматриваемые проекты по созданию ИБЗ, у которых одна из оценок P_i близка к нулю, будут отвергнуты, как несоответствующие требованиям. Иными словами, такая оценка по сравнению с (6.1) весьма чувствительна к минимально допустимым значениям своих составляющих. Для компенсации этого недостатка можно использовать комбинированный вариант, например, когда P_0 рассчитывается по формуле (6.1), а P_{ji} – по формулам (6.4), или наоборот, когда P_0 рассчитывается по формуле (6.3), а P_{ji} – по (6.2).

Интегральная оценка на основе метрического критерия. При использовании этого подхода интегральная оценка проекта получается из соотношения вида:

$$P_0 = \sqrt{\sum_{i=1}^9 \mu_i (1 - P_i)^2}, (\sum_{i=1}^{14} \mu_i = 1), \quad (6.5)$$

где коэффициенты μ_i имеют тот же смысл, что и в предыдущих случаях, а P_i рассчитываются по формулам:

$$P_1 = \sqrt{\sum_{i=1}^3 \mu_{1i} (1 - P_{1i})^2}; P_2 = \sqrt{\sum_{i=1}^9 \mu_{2i} (1 - P_{2i})^2}; \quad (6.6)$$

$$P_3 = \sqrt{\sum_{i=1}^2 \mu_{3i} (1 - P_{3i})^2}; P_8 = \sqrt{\sum_{i=1}^4 \mu_{8i} (1 - P_{8i})^2};$$

$$P_9 = \sqrt{\sum_{i=1}^6 \mu_{9i} (1 - P_{9i})^2},$$

где все компоненты имеют тот же смысл, что и в предыдущем случае.

Этот подход можно интерпретировать так: интегральная оценка есть расстояние в 9-ти мерном фазовом пространстве от точки, характеризующей некий идеальный проект (то есть проект, у которого все локальные критерии равны единице), до точки, соответствующей оцениваемому проекту. Такой подход обладает тем недостатком, что может дать односторонне завышенную оценку за счет значительного превышения значения одного из критериев над всеми остальными. Этот недостаток в определенной степени можно преодолеть, если для расчета величины P_0 воспользоваться формулой (6.5), для расчета ее компонентов – формулами (6.4).

Другой подход повышения адекватности интегральной оценки проектов заключается в использовании показателя «трудности» достижения целей проекта (T), где под целями проекта понимается достижение максимально возможных значений величин $P_1 - P_9$. В этом случае интегральная оценка дается соотношением:

$$T = 1 - \prod_{i=1}^9 (1 - T_i), \quad (6.7)$$

где $T_i = \frac{\varepsilon_i (1 - P_i)}{P_i (1 - \varepsilon_i)}$ – «трудность» достижения цели проекта по i -му критерию; $P_i \geq \varepsilon_i$ – пороговое значение i -го критерия.

Поскольку все критерии нормированы к единице, то есть максимальное значение каждого из них равно 1, то данный метод допускает простейшую геометрическую интерпретацию: объем 9-ти мерного единичного параллелепипеда, характеризующего идеальный проект с максимально возможными значениями критериев, из которого вычтен объем параллелепипеда, характеризующего оцениваемый проект. Таким образом, оставшийся объем 9-ти мерной фигуры характеризует степень достижения оцениваемого проекта его целей. При таком подходе обеспечивается более равномерный учет каждого критерия в формировании интегральной оценки и исключается неоправданное увеличение интегрального итога за счет одного критерия, имеющего большую величину по сравнению со всеми остальными критериями.

Интегральная оценка на основе матрицы потерь. Особенность рассмотренных выше подходов заключается в том, что, выбирая вари-

ант проекта наилучший по одному из критериев, мы неизбежно проигрываем по другим критериям. Естественно возникает вопрос о том, насколько существенен этот проигрыш для каждого из вариантов выбора. Ответ на этот вопрос дает подход на основе использования так называемой матрицы потерь.

Рассмотрим n проектов, каждый из которых характеризуется девятью уже известными нам критериями. Обозначим через X_{ij} оценку i -го проекта по j -му критерию, и сведем оценки всех проектов в матрицу

$$D = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{19} \\ X_{21} & X_{22} & \dots & X_{29} \\ \dots & \dots & \dots & \dots \\ X_{n1} & X_{n2} & \dots & X_{n9} \end{bmatrix}.$$

Проведем дополнительную нормировку критериев:

$$\bar{X}_{ij} = \frac{X_{ij} - X_{ij}^{\min}}{X_{ij}^{\max} - X_{ij}^{\min}} - \text{для критериев ориентированных на максимум, то}$$

$$\text{есть чем больше значение критерия, тем выше качество проекта по} \\ \text{этому критерию; } \bar{X}_{ij} = 1 - \frac{X_{ij} - X_{ij}^{\min}}{X_{ij}^{\max} - X_{ij}^{\min}} - \text{для критериев ориентиро-}$$

ванных на минимум, то есть чем меньше значение критерия, тем выше качество проекта по этому критерию, где X_{ij}^{\max} , X_{ij}^{\min} – минимально возможное и максимально возможные значения критерия; \bar{X}_{ij} – нормированное значение критерия X_{ij} .

Данная нормировка позволяет привести все критерии к одному типу, то есть к критериям, ориентированным на максимум. Это дает возможность построить вектор Y^* идеального варианта проекта, критерии которого находятся по формуле $Y_j = \max_i X_{ij}$. Учитывая, что согласно условиям нормировки самое лучшее значение, принимаемое любым критерием равно 1, получаем, что все компоненты вектора Y^* идеального варианта, будут равны 1, то есть $Y^* = \{1, 1, \dots, 1\}$.

Далее, строим вспомогательную матрицу $A = \|\alpha_{ij}\|$ по следующему правилу: произвольный элемент матрицы α_{ij} есть значение i -го критерия, если выбирается проект, лучший по j -му критерию. На основе этой матрицы и вектора Y^* идеального варианта проекта строим

итоговую матрицу $P = \|p_{ij}\| = Y^* - A = \|y_j^* - \alpha_{ij}\| = \|1 - \alpha_{ij}\|$, элементы которой характеризует потери при выборе конкретного проекта.

Интегральная оценка методом иерархической дихотомии.

Идея данного подхода заключается в том, что критерии качества проекта ИБЗ ($P_1 - P_{49}$) упорядочиваются в иерархическую структуру, на каждом уровне которой дается агрегированная оценка критериев предыдущего уровня. При этом возможны следующие варианты оценок.

Гарантированная оценка, когда качественным свойством оцениваемого проекта является равномерное улучшение по всем локальным критериям. В этом случае на каждом i -ом уровне дихотомическая оценка имеет вид:

$$P_{ij} = \min_j [(\mu_{ij}P_{ij}, \mu_{i+1,j+1}P_{i+1,j+1})], \quad (6.8)$$

где μ_{ij} – коэффициенты, отражающие относительную значимость локальных критериев, образующих данный уровень.

При этом интегральная оценка дается по формуле:

$$P_0 = \min_j \left\{ \min_j \left\{ \min_j \left\{ \min[(\mu_{ij}, P_{ij}), (\mu_{i+1,j+1}, P_{i+1,j+1})] \right\} \right\} \right\}, \quad (6.9)$$

Положительным свойством оценки (6.9) является простота выделения «узких мест» проекта, т.е. тех его критериев, которые являются «критическими» и на улучшение которых следует обратить первоочередное внимание. Оценка (6.9) имеет и другую важную интерпретацию: она является гарантированной оценкой степени достижения целей проекта (например, $P_0 = 0,2$ означает, что близость проекта к цели составляет не менее чем 20% по каждому локальному критерию).

Оптимистическая оценка, когда качественным свойством оцениваемого проекта является улучшение хотя бы по одному из локальных критериев. В этом случае уровневая дихотомическая оценка принимает вид

$$P_{ij} = \max_j [(\mu_{ij}P_{ij}, \mu_{i+1,j+1}P_{i+1,j+1})], \quad (6.10)$$

а интегральная оценка может быть получена по формуле:

$$P_0 = \max_j \left\{ \max_j \left\{ \max_j \left\{ \max[(\mu_{ij}, P_{ij}), (\mu_{i+1,j+1}, P_{i+1,j+1})] \right\} \right\} \right\}. \quad (6.11)$$

Возможен комбинированный вариант, когда для одних уровней используется гарантированная оценка, а для других оптимистическая. Такой вариант целесообразен в том случае, когда недостатки по одним критериям проекта могут компенсироваться достижениями по другим.

Интегральная оценка на основе нечетких множеств. Рассмотренные выше подходы к интегральной оценке качества проектов по созданию ИБЗ имеют тот существенный недостаток, что получаемые с их помощью результаты в значительной мере зависят от принятой базы сравнения. Причем оценки могут быть получены только при условии, если критерии качества выражены в количественных категориях. На практике реализация этих требований вызывает определенные затруднения: какая-то часть критериев будет носить качественный характер, по каким-то критериям нельзя с уверенностью задать допустимые значения. В целях устранения указанных недостатков для получения интегральных оценок качества проектов по созданию ИБЗ можно использовать подходы, основанные на положениях теории нечетких множеств.

Наиболее наглядными, являющимися прямым обобщением построения интегральных оценок для традиционных задач, являются модели аддитивной и мультипликативной свертки. Пусть некоторый проект ИБЗ оценивается теми же показателями, что и в предыдущих случаях, но заданных не числами, а нормированными $[0,1]$ функциями принадлежности $\eta_i(P_i)$. Так, например, $\eta_4(P_4)$ – функция принадлежности прагматического качества ИБЗ, характеризующая степень соответствия ИБЗ своему предназначению. Распространяя действие (6.1), (6.2) и (6.3), (6.4) на функции принадлежности и, определив для них операции умножения и сложения, получаем аддитивную или мультипликативную свертку в категориях нечетких множеств:

$$\eta_0(P_0) \approx 0,07 \left[\begin{aligned} & (0,33 \sum_{i=1}^3 \mu_{1i} \eta_{1i} P_{1i}) + (0,11 \sum_{i=1}^9 \mu_{2i} \eta_{2i} P_{2i}) + \\ & + (0,5 \sum_{i=1}^2 \mu_{3i} \eta_{3i} P_{3i}) + (0,25 \sum_{i=1}^4 \mu_{8i} \eta_{8i} P_{8i}) \\ & + (0,17 \sum_{i=1}^6 \mu_{9i} \eta_{9i} P_{9i}) \end{aligned} \right]; \quad (6.12)$$

$$\eta_0(P_0) = \left[\begin{aligned} & \left(\prod_{i=1}^3 \mu_{1i} \eta_{1i} P_{1i} \right) \times \left(\prod_{i=1}^9 \mu_{2i} \eta_{2i} P_{2i} \right) \times \\ & \left(\prod_{i=1}^2 \mu_{3i} \eta_{3i} P_{3i} \right) \times \left(\prod_{i=1}^4 \mu_{8i} \eta_{8i} P_{8i} \right) \times \\ & \times \left(\prod_{i=1}^6 \mu_{9i} \eta_{9i} P_{9i} \right) \end{aligned} \right]; \quad (6.13)$$

Другой метод свертки локальных критериев качества проектов, определенных на нечетких множествах, предполагает построение для

каждой оценки результирующего множества, являющегося пресечением исходных. То есть интегральная оценка $\eta_0(P_0)$ по всей совокупности локальных критериев проекта $\eta_i(P_i)$, $i = 1, 2, \dots, n$, задается в виде нечеткого множества

$$P_0 = \{\eta_{P_1}(P_1) / P_1; \eta_{P_2}(P_2) / P_2; \dots \eta_{P_n}(P_n) / P_n\}.$$

При этом интегральная оценка получается как пересечение множеств P_i : $P_0 = P_1 \cap P_2 \cap \dots \cap P_n$, а функция принадлежности интегральной оценки проекта определяется по формуле: $\eta_0(P_0) = \min(\eta_{P_1}(P_1); \dots \eta_{P_i}(P_i); \dots \eta_{P_n}(P_n))$. Лучшим будет считаться тот проект, который имеет наибольшую функцию принадлежности интегральной оценки.

В том случае, если задана важность каждого локального критерия в форме весовых коэффициентов, определяемых набором чисел $\mu_1, \mu_2, \dots, \mu_n$, то нечеткая функция принадлежности интегральной оценки проекта находится по формуле

$$\eta_0(P_0) = \min(\mu_1 \eta_{P_1}(P_1); \dots \mu_i \eta_{P_i}(P_i); \dots \mu_n \eta_{P_n}(P_n)).$$

Следующая модель нечеткой свертки основана на использовании расстояния Хемминга, определяющего «дистанцию» между оцениваемым проектом и некоторым эталонным проектом. Для случая двух критериев P_1 и P_2 , заданных своими функциями принадлежности $\eta_{P_1}(P_1)$ и $\eta_{P_2}(P_2)$, расстояние Хемминга $D(P_1, P_2)$ определяется по формуле:

$$D(P_1, P_2) = (\eta_{P_1}(x_1) - \eta_{P_2}(x_1)) + (\eta_{P_1}(x_2) - \eta_{P_2}(x_2)) + \dots + (\eta_{P_1}(x_n) - \eta_{P_2}(x_n)),$$

где x_i – последовательные дискретные значения функций принадлежности $\eta_{P_1}(P_1)$ и $\eta_{P_2}(P_2)$ на осях P_1 и P_2 .

Определим, для примера, расстояние Хемминга для двух нечетких множеств, заданных дискретными функциями принадлежности: $\eta_{P_1}(P_1) = \{10/0,2; 15/0,4; 60/0,9; 65/0,7\}$; $\eta_{P_2}(P_2) = \{10/0,4; 15/0,6; 60/0,4; 65/0,3; 30/0,6\}$. Тогда расстояние Хемминга определится следующим образом:

$$D(P_1, P_2) = |0,2 - 0,4| + |0,4 - 0,6| + |0,9 - 0,4| + |0,7 - 0,3| + |0 - 0,6| = 1,9.$$

Ценность такого подхода заключается в том, что он позволяет осуществлять сравнение между двумя нечеткими множествами, определяя степень их близости. Это дает ключ к интегральной оценке проектов при отсутствии формализованных критериев оптимальности. С этой целью необходимо задать нечеткое множество, характеризующее идеальную (эталонную) оценку, а затем определить расстояние Хемминга каждого из оцениваемых проектов до этого эталона. Наилучший

выбор будет соответствовать наименьшему расстоянию. При этом следует отметить, что построение эталона будет в данном случае гораздо проще, чем в случае представления проекта конкретными числовыми критериями: достаточно принять все критерии на уровне «очень высокие», а значимость их оставить уже заданную. Это и позволит построить нечеткое множество – эталон, расстояние до которого будет определяться при интегральной оценке.

6.3. ОЦЕНКА КАЧЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ ПО ЛОКАЛЬНЫМ КРИТЕРИЯМ

Для решения этой задачи предлагается использовать экспертную методику, существо которой заключается в следующем.

Шаг 1. Получение экспертных оценок. Пусть с целью оценки качества некоторого проекта по созданию ИБЗ сформирована экспертная группа в составе n специалистов. Эксперты, ознакомившись с сутью проекта, дали каждый свои оценки его качества по локальным критериям $P_1 - P_9$. При этом экспертам было предложено давать свои оценки по градациям: «отлично», «хорошо», «удовлетворительно», «плохо», «очень плохо», используя шкалу $[0,1]$ – (1-высшая оценка, 0-низшая оценка). Полученные оценки для каждого локального критерия сводим в табл. 6.2.

Таблица 6.2. Экспертные оценки качества ИБЗ по локальному критерию P_k

Градации качества (i):	Номер эксперта (j)						
	1	2	3	4	5	6	7
«отлично»							
«хорошо»							
«удовлетворительно»							
«плохо»							
«очень плохо»							

Шаг 2. Сближение мнений экспертов и корректировка результатов экспертизы. Оценкам каждого эксперта присваиваем порядковые номера (R_{ij}) в соответствии с убыванием их значений, то есть наиболее высокая оценка получает первый номер, а наименьшая – наибольший. В том случае, если все оценки j -го эксперта по градации i различны, то ранги оценок совпадают с числами натурального ряда от 1 до N . Если же для некоторых градаций оценки эксперта совпадают, то им присваиваются ранги, равные среднему арифметическому соответствующих чисел натурального ряда.

Вычисляем сумму рангов S_i по i -й градации:

$$S_i = \sum_{j=1}^n R_{ij}; i = 1, 2, \dots, m, \quad (6.15)$$

где R_{ij} – ранг оценки, данный j -м экспертом по i -й градации.

Рассчитываем среднее значение суммы рангов:

$$\bar{S} = \frac{1}{m} \sum_{i=1}^m S_i. \quad (6.16)$$

Определяем коэффициент конкордации:

$$W = 12 \sum_{i=1}^m (S_i - \bar{S})^2 / n^2 (m^3 - m). \quad (6.17)$$

Если величина W меньше 0,5, то результаты экспертизы признаем неудовлетворительными.

Определяем, кто из экспертов высказал мнение отличное от большинства. Для этого рассчитываем коэффициенты парной ранговой корреляции ρ_{ij} , характеризующие попарный уровень согласованности мнений экспертов:

$$\rho_{jk} = \frac{6 \sum_{i=1}^m (R_{ij} - R_{jk})^2}{m(m^2 - 1)}, \quad (6.18)$$

где R_{ij}, R_{ik} – ранги попарных оценок i -го и j -го экспертов ($i = \overline{1, n}; j = \overline{1, n}; j \neq k$).

Результаты расчетов сводим в табл. 6.3.

Таблица 6.3. Коэффициенты парной ранговой корреляции

	1 экс- перт	2 экс- перт	3 экс- перт	4 экс- перт	5 экс- перт	6 экс- перт	7 экс- перт
1 эксперт	0						
2 эксперт		0					
3 эксперт			0				
4 эксперт				0			
5 эксперт					0		
6 эксперт						0	
7 эксперт							0

Коэффициенты парной ранговой корреляции ρ_{jk} должны находиться в пределах от -1 до +1. Если $\rho_{jk} = 1$, то мнения j -го и k -го экс-

пертов полностью совпадают; при $\rho_{jk} = -1$ – противоположны. Отрицательные значения коэффициентов парной корреляции свидетельствуют о слишком больших разногласиях между экспертами, что может означать или неудачно подобранный состав группы экспертов или нечетко поставленную им задачу. В этом случае проводим реформирование группы экспертов, повторно знакомим экспертов с содержанием проекта, уточняем задачи экспертизы и повторяем шаги 1 и 2. Такой циклический процесс осуществляем до тех пор пока не удастся получить согласованное мнение экспертов.

Шаг 3. *Вычисление экспертной оценки проекта по локальным критериям.* Используя согласованные данные табл. 6.2, для каждого локального критерия вычисляем средние оценки качества проекта по i -ой градации (\bar{C}_i^k):

$$\bar{C}_i^k = \frac{1}{m} \sum_{j=1}^n C_{ij}^k, \quad (6.19)$$

где n – число экспертов; m – число градаций, принятых при оценке критериев (в нашем случае $m = 5$); k – номер локального критерия ($k = 1, \dots, 9$).

Затем задаемся весом каждой градации v_i ($i = \overline{1,5}$), например, 0,8; 0,6; 0,4; 0,2; 0,1 для градаций «отлично», «хорошо», «удовлетворительно», «плохо», «очень плохо», соответственно, и по формуле:

$$P_k = \frac{1}{n} \left(\sum_{i=1}^m v_i \bar{C}_i^k \right) \quad (6.20)$$

вычисляем экспертные оценки качества проекта по каждому из локальных критериев. Результаты сводим в табл. 6.4.

Таблица 6.4. Результаты экспертной оценки качества проекта ИБЗ по локальным критериям

Наименование критерия	Обозначение	Оценка
социальная эффективность	P_1	
функциональная эффективность	P_2	
эргономическая эффективность	P_3	
прагматическая эффективность	P_4	
техническая эффективность	P_5	
технологическая эффективность	P_6	
экономическая эффективность	P_7	
эксплуатационная эффективность	P_8	
надежность и безопасность	P_9	

7. ОСНОВЫ УПРАВЛЕНИЯ ПРОЕКТАМИ ПО СОЗДАНИЮ ИНТЕЛЛЕКТУАЛЬНЫХ БАЗ ЗНАНИЙ

7.1. ГЛАВНЫЕ ПОНЯТИЯ

Управление проектами по созданию ИБЗ представляется структурной композицией четырех компонентов: субъекта управления; объекта управления; процесса управления проектом и окружения проекта.

Субъектами управления выступают активные участники проекта, взаимодействующие при выработке и принятии проектных решений в процессе его осуществления: заказчик (инвестор), генеральный подрядчик, команда управления проектом (главный конструктор, его заместители, ответственные исполнители составных частей проекта).

Объектом управления выступает сама ИБЗ на проектной фазе ее жизненного цикла (от формирования замысла до приемки проекта включительно).

Процесс управления проектом – это целенаправленное воздействие субъектов управления на объект управления посредством принятия и реализации решений в интересах достижения целей проекта с учетом существующих ограничений социального, технического, экономического, организационного, технологического и другого плана.

Окружение проекта – это среда, порождающая внешние и внутренние факторы, которые способствуют или мешают достижению целей проекта. К таким факторам, прежде всего, относятся финансовые, материально-технические, организационно-кадровые и правовые аспекты обеспечения проектов по созданию ИБЗ.

7.2. ОСОБЕННОСТИ УПРАВЛЕНИЯ ПРОЕКТАМИ

Основные трудности управления проектами по созданию ИБЗ связаны с тем, что этому процессу свойственны следующие особенности:

- субъективность, обусловленная влиянием человеческого фактора при задании целей и принятии проектных решений;
- противоречивость, обусловленная наличием множества разнородных и зачастую несовпадающих целей проекта, а также существованием противоречий между заказчиком, пользователем, разработчиками, конкурентами, злоумышленниками и другими лицами, так или иначе связанными с проектированием и последующим использованием ИБЗ;
- неопределенность, связанная с изменчивостью и непредсказуемостью окружения проекта;
- слабая формализуемость, выражающаяся в том, что многие сущностные стороны объекта управления выяснены не настолько хорошо и полно, что их можно выразить в числах или символах, прихо-

дится оперировать не только количественными, но и словесными (качественными) описаниями их аспектов, показателей и критериев функционирования.

7.3. КОНЦЕПЦИЯ УПРАВЛЕНИЯ ПРОЕКТАМИ

В силу указанных особенностей проблема управления проектами по созданию ИБЗ требует применения особой концепции, которая (помимо системной ориентации) должна предоставлять возможность парировать трудности, возникающие вследствие отмеченных особенностей. В наилучшей степени этому отвечает гомеостатическая концепция управления, идея которой поясняется схемой, представленной на рис. 7.1, где символами $Q_1 - Q_5$ обозначены следующие операции: Q_1 – концептуализация проблемы (определение целей проекта и общего облика проектируемого объекта, разработка технического задания); Q_2 – установление критериев принятия проектных решений; Q_3 – идеализация (упрощение облика проектируемого объекта до уровня, позволяющего оценить качество проекта по установленным критериям); Q_4 – декомпозиция проекта на составные части, формирование и выполнение частных проектов; Q_5 – композиция результатов частных проектов, оформление результатов проектирования, представление проекта заказчику и его защита.

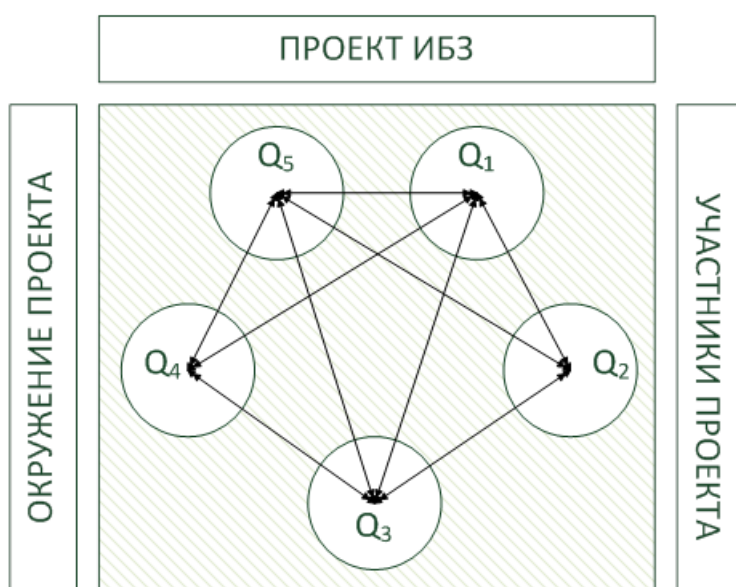


Рис. 7.1. Идея гомеостатической концепции управления проектами по созданию ИБЗ

Реализация этих операций выглядит следующим образом. Пусть проведено предпроектное обследование и сформулированы основные

требования к проекту. Необходимо разработать и выбрать вариант, удовлетворяющий требованиям заказчика, то есть в первом приближении иметь альтернативные варианты построения ИБЗ, а для сопоставления альтернативных вариантов и хотя бы самого поверхностного выбора из них предпочтительного нужно установить соответствующие критерии. Таким образом, оказывается, что, приступив к выполнению операции Q_1 , нельзя получить никаких достаточно обоснованных результатов, не начав одновременно с ней выполнять Q_2 и т.д. Но, получив для некоторой операции определенное решение, необходимо проверить, как такое решение может сказаться на выполнении предыдущих операций. Изложенное поясняет смысл соединения на схеме рис. 1.8 всех операций Q_1 - Q_5 между собой разнонаправленными стрелками – по крайней мере, теоретически все они должны выполняться параллельно и во взаимосвязи, а это не что иное, как системный гомеостаз, в ходе которого реализуется взаимозависимость операций проектирования ИБЗ и на этой основе находится оптимальное

7.4. ТЕХНОЛОГИЯ УПРАВЛЕНИЯ ПРОЕКТАМИ

На практике гомеостатическая концепция управления проектом по созданию ИБЗ. реализуется в виде технологии, которая в схематичном виде представлена на рис. 7.2.

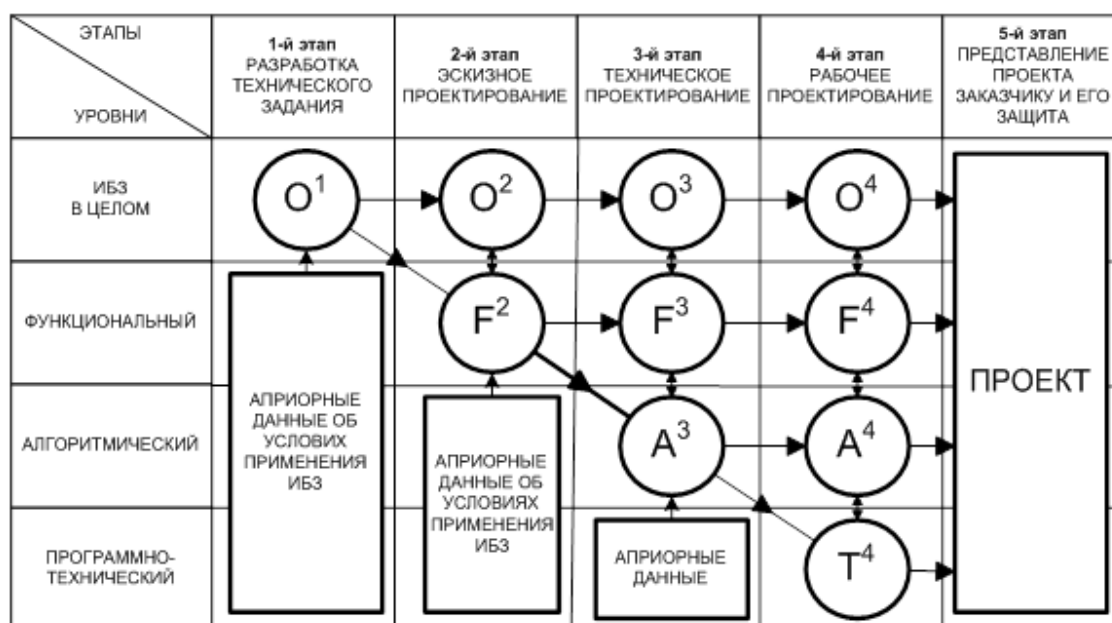


Рис. 7.2. Технология управления проектом по созданию ИБЗ

Согласно этой технологии, процесс управления проектом развития ИКС реализуется в виде поэтапного итерационного циклического

поиска рационального проектного решения в заданных условиях и при заданных ограничениях. По времени он разделяется на стандартные этапы, каждый из которых соответствует определенной точке зрения на объект проектирования и управления, причем с переходом от этапа к этапу происходит повышение уровня детализации объекта проектирования. На каждом из выделенных этапов процесс управления проектом реализуется в виде совокупности типовых операций общесистемного (О), функционального (F), алгоритмического (А) и программно-технического проектирования (Т).

По сути, эта технология задает генеральную линию самоорганизующегося многовариантного процесса управления проектом по созданию ИБЗ и, будучи наложенной, на конкретную организационную структуру участников проекта, служит основой для разработки содержательного постоянно уточняемого сетевого план-графика выполнения НИОКР. Принципиальное преимущество данной технологии перед линейной заключается в том, что она позволяет минимизировать количество проектных циклов, упорядочить работу исполнителей проекта и тем самым повысить качество проведения проектных работ и сократить сроки их проведения.

7.5. ВИДЫ НЕОПРЕДЕЛЕННОСТЕЙ ПРИ УПРАВЛЕНИИ ПРОЕКТАМИ И СПОСОБЫ ИХ УЧЕТА

Под неопределенностью при решении задач управления проектами понимается отсутствие, неполнота, недостаточность информации о характеристиках внешней и внутренней среды, влияющих на качество проектных решений. Практически управление любым проектом по созданию ИБЗ сопряжено двумя видами неопределенностей: фундаментальной и утилитарной.

Фундаментальная или квантовая неопределенность проявляется в том, что максимальная точность определения (измерения) количественно описываемых характеристик управляемого объекта зависит от присущей данному объекту области неопределенности, внутри которой повышение точности определения одной характеристики влечет за собой снижение точности определения другой характеристики. Природа этой неопределенности состоит в том, что любая измеряемая величина, являющаяся количественным выражением свойства управляемого объекта, влияет через внутрисистемный инвариант на другие его свойства. В результате наблюдается нечто, подобное закону сохранения: точность измерения зависимых характеристик объекта управления – величина постоянная. В результате, чем точнее измеряется одна характеристика, тем большая погрешность возникает при измерении другой взаимосвязанной характеристики.

Фундаментальную неопределенность нельзя ликвидировать, так сказать «силовым» способом, но ее можно учесть в процессе управления. Так, например, при расчете стоимости проекта по созданию ИБЗ, следует исходить из того, что вследствие неопределенности рынка, погрешность такого расчета составит не менее 15-20% в сторону повышения. Попытки уменьшить эту погрешность инструментальными или какими-то другими средствами приведут лишь к тому, что пропорционально возрастут погрешности определения других параметров проекта, в частности, такого как срок его и выполнения.

Под *утилитарной или практической неопределенностью* понимают отсутствие, неполноту, недостаточность у лица, принимающего решение, данных о конкретных характеристиках проекта и его окружения. Эта неопределенность ситуативна по своей природе: она может присутствовать на одних этапах проектирования, и отсутствовать на других; одним процессам управления она присуща, другим — несвойственна.

В зависимости от характера информированности лица, принимающего решение, различают следующие виды утилитарной неопределенности.

Вероятностная неопределенность — заключается в знании только распределения вероятности возможных значений некоторой характеристики объекта проектирования, но не ее конкретного значения.

Интервальная неопределенность — заключается в знании не конкретного значения характеристики объекта проектирования, а множества ее возможных значений в некотором ограниченном интервале.

Нечеткая неопределенность — заключается в знании только степени принадлежности характеристики объекта проектирования к какой-либо величине или диапазону величин.

Другим признаком выделения видов утилитарной неопределенности служит характер ее влияния на проектные решения. С этой точки зрения выделяют следующие виды неопределенности.

Объективная неопределенность — неполная информированность о характеристиках, влияющих на принятие проектных решений, которую проектировщик ИБЗ может учесть, но от которой он не может избавиться.

Субъективная неопределенность — неполная информированность о характеристиках, влияющих на принятие проектных решений, возникающая вследствие специфичности восприятия данным проектировщиком объекта управления.

Третьим признаком выделения видов утилитарной неопределенности служит способ представления параметров, влияющих на принятие проектных решений.

Параметрическая неопределенность — возникает тогда, когда неопределенность касается технических, эксплуатационных, эргономических, информационных, стоимостных и других параметров ИБЗ.

Рефлексивная неопределенность — возникает вследствие неполной информированности проектировщиков ИБЗ о замыслах, намерениях и действиях тех сил, которые будут стремиться нарушить ее функционирование в процессе будущей эксплуатации.

Структурная неопределенность — имеет место тогда, когда неопределенность касается связей, взаимодействий и отношений как внутри коллектива проектировщиков ИБЗ, так и с их окружением.

Когда проектировщики ИБЗ сталкиваются с неопределенностями такого рода, то они обычно оперируют следующими способами их парирования:

- сознательно (или бессознательно) игнорируют существование неопределенностей, то есть принимают проектные решения так, как будто неопределенностей не существует, предполагая при этом, что их учет будет осуществлен операторами уже в ходе эксплуатации ИБЗ;

- экспертным путем выбирают наиболее существенные, с их точки зрения, виды неопределенностей и используют соответствующие теории и математические структуры, ориентированные на учет выбранных видов неопределенностей;

- организуют лабораторные эксперименты, испытания и опытную эксплуатацию макета ИБЗ (создавая при этом условия, максимально приближенные к реальным), по результатам которых осуществляют ее доработку, модернизацию, а при необходимости и коренную реконструкцию.

Если исключить первый способ как заведомо неконструктивный, то следует признать, что остальные два способа имеют право на существование. Причем, с утилитарной точки зрения наиболее эффективным будет третий способ, позволяющий в реальных условиях (или приближенным к ним) выяснить спектр неопределенностей и практически оценить их влияние на качество функционирования ИБЗ. Вместе с тем, он обладает двумя очевидными недостатками. Во-первых, требует для своей реализации экономических и материальных затрат, и во-вторых, увеличивает сроки проектирования ИБЗ, превращая его в перманентную доработку системы, что тоже влечет за собой дополнительные затраты.

Учитывая сказанное, при управлении проектами по созданию ИБЗ рекомендуется комплексный подход к учету неопределенностей, при котором не исключаются лабораторные эксперименты и испытания, но одновременно используются теоретические методы учета неопределенностей, а именно:

а) применение экспериментальных распределений вероятностей или, в крайнем случае, их аппроксимаций при учете вероятностно-параметрических неопределенностей;

б) типизация внешних условий применения ИБЗ с последующим применением экспертных процедур ранжирования их важности для парирования неопределенностей внешнего характера;

в) использование аппарата теории нечетких множеств и лингвистического моделирования для учета неопределенностей нечеткого и структурного свойства;

г) применение специальных процедур целевой декомпозиции процесса проектирования ИКС для минимизации неопределенностей субъективного характера;

д) использование аппарата теории игр и теории конфликта для парирования неопределенностей рефлексивного свойства.

7.6. ПРИЕМКА ПРОЕКТА

Одним из наиболее ответственных этапов управления проектами по созданию ИБЗ является представление результатов проектных работ заказчику и их защита. Этот вопрос узаконен соответствующими ГОС-Тами, однако на содержательном уровне в нем остается много «подводных камней». Ограничимся следующим. Проводя работы на этом этапе, исполнители проекта должны (помимо соблюдения всех формальных процедур) исходить из того, что в процессе приемки проекта заказчик вправе и должен получить обоснованные ответы на следующие типовые вопросы:

Исходя из каких соображений выполнение данного проекта было поручено данному исполнителю? Проводился ли конкурс при заключении договора на выполнение проекта?

В чем заключалась суть проекта? Каковы цели проекта, и какие задачи решались для их достижения?

Разрабатывался ли план проектных работ? Если да, то на какие этапы он разделялся, как соблюдались сроки выполнения этапов и каковы были объемы поэтапного финансирования?

Какие сторонние коллективы привлекались для производства проектных работ? Заключались ли субдоговора? Если да, то каковы были объемы финансирования субподрядчиков?

Какие альтернативные варианты технических решений были рассмотрены в процессе выполнения проекта, и на основе каких соображений осуществлялось их ранжирование? Какие альтернативы были исключены из рассмотрения как заведомо непригодные?

Какие исходные предположения составляли базис проектных работ? Как эти предположения выражены и учтены в ходе проектирования?

8. Какие критерии использовались при проведении проектных работ, какие методы (методики) использовались для оценки проектируемого объекта по этим критериям?

Результаты каких фундаментальных исследований использовались при проведении проектных работ?

Чем подтверждается комплексный характер данного проекта?

Какие типы неопределенностей учитывались при проведении проектных работ, какие методы применялись для парирования неопределенностей?

Учитывались ли при проведении проектных работ возможности конкурентов, противников и злоумышленников, какие методы применялись для учета этих возможностей?

Каким образом предполагается обеспечивать информационную безопасность объекта проектирования?

Проводились ли в процессе проектных работ натурные, лабораторные и компьютерные эксперименты? Если да, то в чем заключаются их результаты?

Сколько заявок на предполагаемые изобретения было подано в процессе выполнения проектных работ?

Что служит основанием считать данные проектные работы завершенными?

Какие решения должен принять заказчик по результатам выполненного проекта? Когда и на каком уровне должны быть приняты эти решения? Какие последствия могут повлечь за собой те или иные решения? Каковы будут затраты на реализацию предлагаемых решений? Какова ожидаемая прибыль от реализации проекта, и в чем она может быть выражена?

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Балан В.П., Душкин А.В., Новосельцев В.И., Сумин В.И. Управление конфликтами: учебное пособие для вузов / Под ред. проф. В.И. Новосельцева. – М.: Горячая линия – Телеком, 2015. – 144 с.
2. Барский А.Б. Нейронные сети: распознавание, управление, принятие решений. – М.: Финансы и статистика, 2004. – 176 с.
3. Братко И. Алгоритмы искусственного интеллекта на языке PROLOG. – М.: Вильямс, 2004. – 640 с.
4. Борисов В.В., Круглов В.В., Федулов А.С. Нечеткие модели и сети. – М.: Горячая линия – Телеком, 2015. – 284 с.
5. Борисов В.В., Федулов А.С., Зернов М.М. Основы нечеткого логического вывода. – М.: Горячая линия – Телеком, 2014. – 122 с.
6. Борисов В.В., Федулов А.С., Зернов М.М. Основы нечеткой арифметики. – М.: Горячая линия – Телеком, 2014. – 84 с.
7. Борисов В.В., Федулов А.С., Зернов М.М. Основы теории нечетких отношений. – М.: Горячая линия – Телеком, 2014. – 88 с.
8. Борисов В.В., Федулов А.С., Зернов М.М. Основы теории нечетких множеств. – М.: Горячая линия – Телеком, 2014. – 86 с.
9. Бурков В.Н., Новиков Д.А. Как управлять проектами. – М: СИНТЕГ-ГЕО, 1997. – 126 с.
10. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2000. – 384 с.
11. Галушкин А.И. Нейронные сети: основы теории. – М.: Горячая линия – Телеком, 2014. – 496 с.
12. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М: Физматлит, 2006. – 320 с.
13. Девятков В.В. Системы искусственного интеллекта. – М.: МГТУ им. Н.Э. Баумана, 2001. – 352 с.
14. Дёмин Б.Е. Методологические основы и модели обоснования проектов крупномасштабных информационно-коммуникационных систем. – Воронеж: Научная книга, 2006. – 332 с.
15. Джонс М.Т. Программирование искусственного интеллекта в приложениях. – М.: ДМК Пресс, 2004. – 312 с.
16. Душкин А.В., Новосельцев В.И., Сумин В.И. Моделирование систем управления и информационно-технического обеспечения: учебное пособие для вузов / Под ред. проф. В.И. Новосельцева. – М.: Горячая линия – Телеком, 2015. – 192 с.
17. Душкин А.В., Новосельцев В.И., Сумин В.И. Математические модели и информационные процессы управления сложным объектом: монография. – Воронеж: Научная книга, 2014. – 125 с.

18. Информационная безопасность государственных организаций и коммерческих фирм. Справочное пособие / Под ред. Л.Д. Реймана. – М.: НТЦ «ФИОРД-ИНФО», 2002. – 250 с.
19. Круглов В.В, Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия – Телеком, 2011. – 382 с.
20. Ларичев О.И. Теория и методы принятия решений, а также хроника событий в волшебных странах. – М.: Логос, 2002. – 392 с.
21. Липаев В.В. Обеспечение качества программных средств. – М.: СИНТЕГ, 2001. – 230 с.
22. Логико-лингвистические модели в военных системных исследованиях / В.И. Новосельцев и др. – М.: Воениздат, 1988. – 232 с.
23. Люгер Д.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. – М.: Вильямс, 2003. – 864 с.
24. Математические основы управления проектами / Под ред. В.Н. Буркова. – М.: Высшая школа, 2005. – 423 с.
25. Минский М. Фреймы для представления знаний. – М.: Мир, 1979. – 134 с.
26. Моделирование систем: учебник / Душкин А.В., Новосельцев В.И., Сумин И., Кравченко А.С. и др. – Воронеж: Научная книга, 2014. – 236 с.
27. Нечаев Ю.И. Искусственный интеллект: концепции и приложения. – Санкт-Петербург: ГМТУ, 2001. – 234 с.
28. Нечеткие множества в задачах управления и искусственного интеллекта / Под ред. Д.А. Поспелова. – М.: Наука, 1986. – 320 с.
29. Ногин В.Д. Принятие решений в многокритериальной среде: количественный подход. – М.: Физматлит, 2002. – 144 с.
30. Обработка нечеткой информации в системах принятия решений / А.Н. Борисов, А.В. Алексеев, Е.В. Меркурьев и др. – М.: Радио и связь, 1989. – 304 с.
31. Поспелов Д.А. Ситуационное управление: теория и практика – М.: Наука, 1986. – 288 с.
32. Рассел С., Норвиг П. Искусственный интеллект: современный подход. – М.: Вильямс, 2006. – 1408 с.
33. Рутковский Л. Методы и технологии искусственного интеллекта. – М.: Горячая линия – Телеком, 2010. – 520 с.
34. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – М.: Горячая линия – Телеком, 2013. – 384 с.
35. Смирнов А.В., Шереметов Л.Б. Многоагентная технология проектирования сложных систем // Автоматизация проектирования. – 1999. – №1. – С. 42.
36. Смолин Д.В. Введение в искусственный интеллект: конспект лекций. – М.: Физматлит, 2004. – 208 с.

37. Советов Б.Я. Представление знаний в информационных системах. – М: Академия, 2012. – 144 с.
38. Теоретические основы системного анализа / Под. ред. В.И. Новосельцева. – М.: Майор, 2006. – 592 с.
39. Тузов В.А. Компьютерная лингвистика. Опыт построения компьютерных словарей. – СПб.: СПбГУ, 2002. – 650 с.
40. Хайкин С. Нейронные сети: полный курс. – М.: Вильямс, 2006. – 1104 с.
41. Хансен Г., Хансен Д. Базы данных: разработка и управление / Пер с англ. – М.: БИНОМ, 2000. – 704 с.